

# **Gentoo Linux**

сборник статей

Версия 1.3 от 01.11.2007

**2007 г.**

## Оглавление

Gentoo Linux.....	1
Новое и измененное (changelog).....	18
Составители: .....	20
Цели данного сборника.....	20
Вступительное слово.....	21
Вводная информация о Linux.....	22
UNIX.....	22
История.....	22
Стандарты.....	28
Канонические команды UNIX.....	29
GNU/Linux.....	30
Linux (ядро).....	33
История.....	33
Версии.....	34
Поддержка.....	35
Стабильные версии.....	35
Архитектура.....	36
Совместимость.....	37
Лицензия.....	38
Символ.....	38
BSD.....	38
История BSD.....	38
GNU General Public License.....	40
Свобода.....	40
История.....	40
GPL v3.....	41
Схема GNU GPL.....	42
Linux не для идиотов: сборник рассказов и рецептов.....	43
Предисловие.....	43
Ядро и модули.....	43
Загрузка операционной системы.....	46
Организация памяти.....	47
System V shared memory.....	48
Файловая система.....	49
Права доступа.....	52
Журналируемые файловые системы.....	53
Отображенные в память файлы.....	53
Специальные файловые системы.....	54
Сетевые файловые системы.....	55
Создание файловых систем.....	55
Интерфейс sysctl.....	56
Статически и динамически собранные программы.....	56
Системная библиотека GNU libc.....	57
LD, Shared Library, SO и много страшных слов.....	58
Информация о процессах и файловая систем /proc.....	60
Создание процессов.....	60
Секреты /dev.....	61
Статическая организация каталога /dev.....	62
Что такое DevFS.....	63
Немного о UDEV.....	64

Блочные устройства.....	65
Распределение мажоров и миноров IDE дисков.....	67
Распределение мажоров и миноров для SCSI-дисков.....	68
Устройства SATA и переход с IDE на PATA.....	69
Logical Volume Manager.....	70
Software RAID.....	74
Device mapper.....	77
Host-RAID, или дешевых RAID-контроллеров не бывает.....	77
Сетевая подсистема.....	78
Маршрутизация IP и форвардинг.....	80
Фильтры пакетов.....	82
Стандартные таблицы и цепочки.....	82
Порядок применения стандартных таблиц и цепочек.....	83
Стандартные действия.....	84
Условия отбора.....	84
Примеры конфигураций iptables.....	85
Управление пользователями, NSS и PAM.....	87
X11 и все-все-все.....	89
Клиент-серверная архитектура X11.....	89
Графическая среда пользователя.....	90
Внутри X11. Упрощенная схема.....	90
Внутри X11. Растровые шрифты и их отображение.....	91
Внутри X11. FreeType и XFT.....	92
Внутри X11. Расширения.....	92
Внутри X11. Полезные утилиты.....	93
Внутри X11. Менеджер окон.....	93
Локализация.....	93
Файловые системы Linux.....	96
Создание файловых систем.....	96
Необходимое введение.....	96
Устройство файловых систем Unix-семейства.....	98
Файловые системы Linux.....	99
Ext2fs.....	99
О журналируемых файловых системах.....	99
ReiserFS.....	100
Ext3fs.....	101
XFS.....	102
Критерии выбора.....	102
Практические следствия.....	103
Библиография вопроса.....	105
Linux и файловые системы: еще раз о проблеме выбора.....	106
О Gentoo Linux.....	109
Что такое портежи?.....	109
Gentoo Linux.....	110
История Gentoo.....	110
Философия Gentoo.....	111
Советы пользователям Gentoo .....	113
Проект X.org.....	114
Основы Portage.....	114
Управление настройками Portage.....	115
Когда нужно использовать оверлеи?.....	115
Как часто нужно обновляться?.....	116
Деинсталляция пакетов.....	116
Удаление временных файлов.....	117

Gentoo Linux: установка и настройка.....	117
О Gentoo Linux.....	117
Получение дистрибутива.....	118
Установка - первый этап.....	123
Этап второй - подготовка к смене корня.....	125
Третий этап - внутри Gentoo.....	126
Этап четвертый: в недра ядра.....	128
Краткое руководство по установке Gentoo Linux x86.....	138
Установочный диск.....	138
Настройка сети.....	138
Подготовка дисков.....	138
Установка стадии.....	139
Установка базовой системы Gentoo.....	139
Настройка ядра.....	140
Настройка системы.....	140
Установка системных утилит.....	141
Настройка загрузчика.....	141
grub.....	141
# для пользователей genkernel.....	141
# для остальных пользователей (без initrd).....	141
# только если вам нужна двойная загрузка.....	141
lilo.....	142
# для пользователей, не использующих genkernel.....	142
# для пользователей genkernel.....	142
# Для двойной загрузки.....	142
Окончание установки.....	142
Описание альтернативных способов установки Gentoo Linux.....	143
1. Об этом документе.....	143
2. Загрузка с установочного диска при помощи Smart BootManager.....	143
3. Установка при помощи Knoppix.....	144
4. Установка со Stage 1 без доступа в сеть.....	145
5. Бездисковая установка с загрузкой через PXE.....	146
6. Установка Gentoo из существующей Linux-системы.....	150
Полезные советы по установке Gentoo/x86.....	153
1. Вступление.....	153
2. Расширенная установка.....	153
3. Упрощение установки.....	156
4. Решение ошибок/проблем.....	157
Восстановление сбойной установки.....	157
Настольная книга Gentoo Linux x86.....	159
A. Установка Gentoo.....	162
1. Об установке Gentoo Linux.....	162
1.a. Введение.....	162
2. Выбор подходящего источника для установки.....	164
2.a. Аппаратные требования.....	164
2.b. Установочные компакт-диски Gentoo.....	164
2.c. Скачивание, запись и загрузка с установочного диска Gentoo.....	166
3. Настройка сети.....	170
3.a. Автоматическое подключение к сети.....	170
3.b. Автоматизированная настройка сети.....	171
3.c. Ручная настройка сети.....	173
Использование DHCP.....	174
Подготовка беспроводного доступа.....	175
Освоение сетевой терминологии.....	176

Использование ifconfig и route.....	177
4. Подготовка дисков.....	178
4.a. Общие сведения о блочных устройствах.....	178
4.b. Разработка схемы разбиения диска.....	179
4.c. Использование fdisk для создания разделов.....	180
4.d. Создание файловых систем.....	184
Файловые системы (сравнение ! ).....	184
4.e. Монтирование.....	185
5. Установка установочных файлов Gentoo.....	186
5.a. Установка архива стадии.....	186
5.b. По умолчанию: использование файла стадии из интернета.....	187
5.c. Альтернатива: использование файла стадии с установочного диска.....	188
5.d. Установка дерева портежей.....	189
5.e. Настройка параметров компиляции.....	190
6. Установка базовой системы Gentoo.....	192
6.a. Изменение корневого каталога.....	192
6.b. Настройка Portage.....	194
Настройка переменной USE.....	196
7. Настройка ядра.....	197
7.a. Часовой пояс.....	197
7.b. Установка исходных кодов ядра.....	198
7.c. По умолчанию: ручная настройка.....	198
7.d. Альтернатива: использование genkernel.....	202
7.e. Модули ядра.....	203
8. Настройка параметров системы.....	204
8.a. Параметры файловых систем.....	204
Примечание (JohnBat26): пример моего /etc/fstab:.....	205
8.b. Параметры сети.....	205
8.c. Параметры системы.....	208
9. Установка нужных системных средств.....	209
9.a. Диспетчер устройств.....	209
9.b. Системные службы журналирования.....	209
9.c. Дополнительно: демон cron.....	210
9.d. Дополнительно: индексация файлов.....	210
9.e. Утилиты для обслуживания файловых систем.....	210
9.f. Сетевые средства.....	211
10. Настройка начального загрузчика.....	212
10.a. Выбор загрузчика.....	212
10.b. По умолчанию: использование GRUB.....	213
10.c. Альтернатива: использование LILO.....	217
10.d. Перезагрузка системы.....	219
11. Завершение установки Gentoo.....	219
11.a. Управление учетными записями.....	219
12. Чем заняться дальше?.....	220
12.a. Документация.....	220
12.b. Gentoo в интернете.....	221
B. Работа с Gentoo.....	221
1. Введение в Portage.....	221
1.a. Добро пожаловать в Portage.....	221
1.b. Дерево портежей.....	222
1.c. Обслуживание программного обеспечения.....	222
1.d. Когда Portage жалуется.....	226
2. USE-флаги.....	230
2.a. Что такое USE-флаги?.....	230

2.b. Использование USE-флагов.....	231
2.c. USE-флаги отдельных пакетов.....	233
3. Возможности Portage.....	234
3.a. Возможности Portage.....	234
3.b. Распределенная компиляция.....	235
3.c. Кэширование компиляции.....	236
3.d. Поддержка двоичных пакетов.....	237
4. Сценарии инициализации.....	238
4.a. Уровни запуска.....	238
4.b. Использование rc-update.....	241
4.c. Настройка служб.....	242
4.d. Написание сценариев инициализации.....	242
4.e. Изменение поведения уровней запуска.....	245
5. Переменные среды.....	247
5.a. Переменные среды.....	247
5.b. Глобальное определение переменных.....	248
5.c. Локальное определение переменных.....	250
C. Работа с Portage.....	251
1. Файлы и каталоги.....	251
1.a. Файлы Portage.....	251
1.b. Хранение файлов.....	252
1.c. Сборка программного обеспечения.....	253
1.d. Ведение журнала.....	254
2. Настройка с помощью переменных.....	254
2.a. Настройка Portage.....	254
2.b. Параметры сборки.....	254
2.c. Защита конфигурационных файлов.....	255
2.d. Параметры скачивания.....	255
2.e. Настройка Gentoo.....	256
2.f. Поведение Portage.....	257
3. Смешение ветвей программного обеспечения.....	257
3.a. Использование одной ветви.....	257
3.b. Одновременное использование стабильной и тестовой ветвей.....	258
3.c. Использование заблокированных пакетов.....	258
4. Дополнительные средства Portage.....	259
4.a. etc-update.....	259
4.b. dispatch-conf.....	261
4.c. quickpkg.....	261
5. Отступление от официального дерева.....	261
5.a. Использование собственного дерева Portage.....	261
5.b. Добавление неофициального сборочного файла ebuild.....	262
5.c. Программы, поддерживаемые не Portage.....	263
6. Использование ebuild.....	263
6.a. Emerge и Ebuild.....	263
6.b. Ручная установка программ.....	263
6.c. Дополнительные возможности Ebuild.....	265
6.d. Дополнительная информация.....	266
D. Настройка сети в Gentoo.....	266
1. Начальная настройка.....	266
1.a. Приступаем к настройке.....	266
2. Расширенная настройка.....	268
2.a. Расширенная настройка.....	268
2.b. Сетевые зависимости.....	270
2.c. Имена и значения переменных.....	270

3. Модульное построение сети.....	271
3.a. Сетевые модули.....	271
3.b. Обработчики интерфейса.....	272
3.c. DHCP.....	272
3.d. Модем ADSL.....	274
3.e. APIPA (автоматическая частная IP-адресация).....	274
3.f. Объединение интерфейсов.....	275
3.g. Образование моста (поддержка 802.1d).....	275
3.h. MAC-адрес.....	276
3.i. Образование туннеля.....	276
3.j. Виртуальные сети (поддержка 802.1q).....	276
4. Беспроводная сеть.....	277
4.a. Введение.....	277
4.b. Запросчик WPA.....	277
4.c. Утилиты Wireless tools.....	279
4.d. Раздельная настройка сети по ESSID.....	283
5. Дополнительные возможности.....	283
5.a. Стандартные функции-обработчики.....	283
5.b. Функции-обработчики wireless tools.....	285
6. Управление сетью.....	286
6.a. Управление сетью.....	286
6.b. ifplugd.....	286
Настольная книга по безопасности Gentoo.....	287
Введение.....	287
A. Безопасность системы.....	287
1. Соображения перед началом установки.....	287
1.a. Физическая безопасность.....	287
1.b. Планирование демонов/служб.....	287
1.c. Схемы создания разделов.....	288
1.d. Пользователь root.....	288
1.e. Правила безопасности.....	289
2. Закручивание гаек.....	290
2.a. USE-флаги.....	290
2.b. Защита GRUB паролем.....	290
2.c. Защита LILO паролем.....	291
2.d. Ограничение доступа к консоли.....	291
3. Журналирование.....	292
3.a. Введение.....	292
3.b. Журналирование: Syslogd.....	292
3.c. Metalog.....	294
3.d. Syslog-ng.....	294
3.e. Анализ журналов с помощью Logcheck.....	296
4. Монтирование разделов.....	296
4.a. Монтирование разделов.....	296
5. Ограничения пользователей/групп.....	297
5.a. /etc/security/limits.conf.....	297
5.b. /etc/limits.....	297
5.c. Квоты.....	298
5.d. /etc/login.defs.....	299
5.e. /etc/login.access.....	299
6. Права доступа к файлам.....	300
6.a. Файлы, доступные по чтению для всех.....	300
6.b. Файлы, доступные по записи для группы или для всех.....	300
6.c. Файлы SUID/SGID.....	300

6.d. SUID/SGID-файлы и жесткие ссылки.....	301
7. PAM.....	302
7.a. PAM.....	302
8. Упаковщики TCP.....	303
8.a. Упаковщики TCP.....	303
9. Безопасность ядра.....	303
9.a. Удаление функциональности.....	303
9.b. Файловая система proc.....	304
9.c. Grsecurity.....	305
9.d. Kerneli.....	306
9.e. Другие заплатки к ядру.....	306
10. Безопасность служб.....	306
10.a. Apache.....	306
10.b. Bind.....	306
10.c. Djbdns.....	307
10.d. FTP.....	307
10.e. Mysql.....	307
10.f. Proftpd.....	308
10.g. Pure-ftpd.....	308
10.h. Vsftpd.....	309
10.i. Netqmail.....	309
10.j. Samba.....	309
10.k. ssh.....	310
10.l. Использование xinetd.....	311
10.m. X.....	312
11. Изменение корневого каталога и виртуальные серверы.....	313
11.a. Изменение корневого каталога.....	313
11.b. Пользовательский режим Linux.....	314
12. Межсетевые экраны.....	315
12.a. Межсетевой экран.....	315
12.b. Фильтрация пакетов.....	315
12.c. Прокси.....	316
12.d. Программный шлюз.....	316
12.e. Iptables.....	317
12.f. Squid.....	324
12.g. Изучено.....	328
13. Обнаружение вторжения.....	328
13.a. AIDE (Advanced Intrusion Detection Environment).....	328
13.b. Snort.....	331
13.c. Обнаружение вредоносных программ с помощью chkrootkit.....	333
14. Регулярные обновления.....	333
14.a. Регулярные обновления.....	333
Обновление Gentoo Linux.....	336
Руководство по обновлению Gentoo.....	336
1. Gentoo и обновления.....	336
2. Гонка за новыми выпусками.....	337
3. Инструкции по обновлению профилей.....	338
HOWTO Полное обновление системы.....	340
Приведение в порядок /var/lib/portage/world .....	340
Обновление profile .....	340
USE-флаги .....	340
Запуск обновления системы (если не нужно обновлять toolchain) .....	340
Некоторые причины не использовать emerge -U вместо -u .....	341
Обновление одного из пакетов входящих в toolchain .....	341



Обновление безопасности.....	342
Удаление неиспользуемых пакетов.....	342
Обновление конфигов.....	343
Руководство Gentoo Linux по обновлению GCC.....	344
1. Введение.....	344
2. Общие указания по обновлению.....	344
3. Переход с GCC-3.3 на 3.4 или более новый.....	345
4. Обновление GCC на новой установке.....	347
5. Обычные грабли.....	349
Gentoo и X.Org.....	350
Описание настройки МОНОЛИТНОГО X-сервера версий 6.X.....	350
1. Что такое X Window Server?.....	350
2. Установка Xorg.....	351
3. Настройка Xorg.....	352
4. Настройка xorg.conf.....	354
5. Ресурсы.....	356
Описание настройки МОДУЛЬНОГО X-сервера версий 7.X (Англ.).....	357
Introduction.....	357
Preparation.....	357
Nvidia Drivers.....	357
Masking Xorg 7.1 (optional).....	357
Backup.....	358
Restoring the backup.....	358
Uninstall old X.org.....	358
Emerge Modular X.org.....	359
Device Drivers.....	359
Installation.....	359
GL Issues.....	360
Configuration file locations have changed.....	362
Other problems and solutions.....	362
XkbOptions.....	363
Параметры "XkbOptions".....	363
Пример.....	363
HOWTO Xorg and Fonts (Рус.).....	364
Назначение этого HOWTO.....	364
Общие сведения.....	364
Замечание по поводу DPI.....	364
Замечание по поводу сравнения шрифтов на Windows/Mac и Linux.....	365
Получение необходимых пакетов.....	365
Информация о хинтинге (Hinting).....	366
.....	367
Изменение конфигурационных файлов.....	367
/etc/X11/xorg.conf.....	370
Файл: /etc/X11/xorg.conf (Список путей к шрифтам).....	372
Перезагрузка служб.....	373
Mozilla Firefox and Thunderbird.....	373
Mozilla Firefox 1.0+.....	373
Mozilla Thunderbird 0.7, 1.0.2 и выше.....	374
Проблемы с размерами шрифтов.....	375
Что делать, если Linux шрифт в Konsole не работает.....	376
Что, если Ваши шрифты стали размытыми и не читаемыми.....	376
Ошибка при запуске startx.....	377
XEmacs.....	377
Полезные ресурсы.....	377

Смотрите также.....	377
Примечание (от JohnBat26), мой файл: ~/.fonts.conf:.....	378
Руководство по видеокартам nVidia в Gentoo Linux.....	380
1. Введение.....	380
2. Настройка карты.....	380
Конфигурирование ядра.....	380
Последующая настройка ядра.....	381
Установка подходящих драйверов.....	383
Настройка X-сервера.....	383
Добавление пользователей в группу video.....	384
Тестирование карты.....	384
Включение поддержки nvidia.....	385
Использование инструмента nVidia Settings.....	385
3. Проблемы.....	385
Работа 2D на компьютерах с ОЗУ 4 ГБ и более.....	385
Пытаясь загрузить модуль ядра, я в ответ получаю «no such device» .....	386
4. Настройка для опытных.....	386
Документация.....	386
Параметры модуля ядра.....	386
Расширенная настройка X-сервера.....	386
Распространенные вопросы об ATI в Gentoo Linux.....	388
1. Поддержка устройств.....	388
2. Установка.....	389
3. Дополнительные источники.....	390
HOWTO fbsplash.....	391
Введение .....	391
Что такое gensplash?.....	391
Что такое fbsplash (framebuffer splash)?.....	391
Что такое устройство frame buffer?.....	391
Установка fbsplash и splashutils.....	392
Установка заплат ядра.....	392
Конфигурация ядра.....	392
Компиляция ядра (Вручную).....	392
Конфигурация / компиляция ядра (Для пользователей genkernel).....	393
Установка splashutils.....	393
Конфигурация / компиляция ядра (ручная работа).....	393
Создание образа initramfs .....	393
Что такое initramfs?.....	393
Компиляция образа initramfs непосредственно в ядро.....	394
Загрузка образа initramfs динамически.....	394
Создание собственных тем для gensplash.....	395
Перевод тем bootsplash в темы gensplash.....	396
Использование собственных тем gensplash.....	396
Настройка загрузчика .....	397
Параметры ядра.....	397
Пример конфигурации Grub.....	397
Пример конфигурации Lilo.....	398
Добавление фона на все консоли .....	398
Gensplash для других дистрибутивов .....	398
Дополнительные шаги.....	399
Тут должно быть название пункта 3.....	399
Решение проблем .....	399
Если /usr на отдельном разделе.....	399
Установка фона во время загрузки.....	399

Ссылки.....	400
Hardware 3D Acceleration Guide (Англ.).....	401
1. Introduction.....	401
2. Install Xorg and configure your kernel.....	402
3. Install X11-DRM and configure direct rendering.....	403
4. Test 3D acceleration.....	404
5. Using the CVS sources.....	404
6. Tweak your performance.....	405
7. Troubleshooting.....	406
8. Acknowledgments.....	407
9. References.....	407
HOWTO по настройке KDE.....	408
1. Что же такое K Desktop Environment?.....	408
2. Установка KDE.....	408
3. Настройка KDE.....	411
Руководство по настройке GNOME.....	413
1. Что такое GNOME?.....	413
2. Установка GNOME.....	413
Использование UTF-8 в Gentoo.....	415
1. Предупреждение или Напутствие переводчика.....	415
2. Кодировки.....	415
3. Включение UTF-8 в Gentoo Linux.....	417
4. Поддержка приложениями.....	419
Руководство по русской локализации Gentoo Linux.....	426
1. Введение.....	426
2. Базовая кириллизация консоли.....	427
3. Установка русской locale.....	430
4. Установка часового пояса.....	432
5. Локализация оконной системы X.....	433
Листинг 5.1: Варианты выбора переключателя латиница/кириллица.....	434
Руководство Gentoo Linux ALSA.....	437
1. Введение.....	437
2. Установка ALSA.....	437
3. Настройка/тестирование ALSA.....	442
ALSA и USE.....	445
4. Другие возможности ALSA.....	447
Руководство по настройке Java в Gentoo.....	451
1. Установка JDK/JRE.....	451
2. Конфигурирование вашей JDK/JRE.....	451
3. Дополнительные ресурсы.....	453
4. Предостережения.....	453
Gentoo и GPRS.....	454
GPRS через USB (Руководство).....	454
Настройка GPRS в Linux.....	457
Введение.....	457
Подготовка.....	457
Сборка пакетов.....	457
Настройка.....	459
Установка соединения.....	460
Автоматическая установка соединения.....	460
Заключение.....	461
Экономим трафик, с помощью Toonel. (Linux).....	461
Сжимаем трафик электронной почты.....	462
GPRS через KPPP.....	463

GPRS через Bluetooth.....	465
HOWTO Соединение с Internet.....	466
Настройка VPN.....	466
Проверка параметров ядра .....	466
Установка пакетов .....	466
Настройка клиента.....	467
Настройка с помощью pptpconfig.....	467
Проверяем соединение .....	468
Через rppoe.....	469
Настройка выхода через DIAL-UP.....	470
Подключение модема.....	470
Последовательный порт.....	470
Подключение через USB.....	470
Win модемы.....	470
Для всех типов модемов.....	470
Проверка работоспособности, возможные грабли.....	471
Настройка программного обеспечения.....	471
Включение поддержки PPP в ядре.....	471
Установка необходимых пакетов.....	471
Настройка соединения с помощью chat-script.....	471
Обзор звонилки.....	472
chestnut-dialer.....	472
Настройка Call back.....	473
Настройка выхода через GPRS.....	473
Требуемая аппаратура.....	473
Подключение с помощью USB-кабеля.....	473
Подключение с помощью COM-кабеля.....	474
Подключение через Bluetooth.....	474
Подключение через IrDA.....	475
Настройка выхода через CDMA (SKYLINK).....	476
Подключение аппаратуры.....	476
Настройка программного обеспечения.....	476
Power Management Guide (Англ.).....	477
1. Introduction.....	477
2. Prerequisites.....	478
3. CPU Power Management.....	482
4. LCD Power Management.....	485
5. Disk Power Management.....	487
6. Power Management For Other Devices.....	489
7. Sleep States: sleep, standby, and suspend to disk.....	491
8. Troubleshooting.....	494
Руководство по udev в Gentoo.....	497
1. Что такое udev?.....	497
2. Как использовать udev в Gentoo.....	499
3. Известные проблемы.....	500
4. Ссылки на ресурсы и благодарности.....	502
Gentoo LDAP-DNS Guide (Англ.).....	503
1. Introduction.....	503
2. Configuring LDAP-DNS.....	503
3. Configuring OpenLDAP.....	504
4. Testing the Installation.....	505
Руководство по настройке Samba в режиме PDC с использованием LDAP.....	506
Устанавливаемые пакеты .....	506
OpenLDAP.....	506

Конфигурирование OpenLDAP.....	506
Запуск сервера OpenLDAP.....	507
SAMBA.....	508
Миграция OpenLDAP.....	508
Конфигурация SAMBA.....	509
Настройка системы на авторизацию в LDAP.....	512
Управление пользователями.....	514
Создание пользователя.....	514
Изменение пароля.....	515
Модификация пользователя.....	515
Удаление пользователя.....	516
Управление пользователями в оффтопике.....	516
Источники.....	516
Создание виртуальной почтовой системы.....	517
1. Введение.....	517
2. Начальная настройка postfix.....	518
3. Courier-imap.....	519
4. Cyrus-sasl.....	520
5. SSL-сертификаты для Postfix и Apache.....	520
6. Добавим SSL и SASL поддержку в Postfix.....	521
7. MySQL.....	522
8. Apache и phpMyAdmin.....	523
9. Vmail-пользователь.....	524
10. Настройка MySQL авторизации и виртуальных доменов.....	525
11. Squirrelmail.....	528
12. Mailman.....	528
13. Фильтрация содержимого и Anti-Virus.....	530
14. Окончание.....	530
15. Troubleshooting.....	531
Виртуальная почтовая система на основе qmail/vpopmail.....	533
1. Введение.....	533
2. qmail ("Разговаривая с собой").....	533
3. vpopmail.....	535
4. Courier POP/IMAP.....	536
5. qmail (разговаривая с миром).....	537
6. Веб клиент Horde / IMP.....	538
7. Дополнительные пакеты.....	539
8. Заключительные примечания.....	541
Руководство по электронной почте с использованием Mutt.....	542
1. Введение в E-Mail.....	542
2. Fetchmail.....	542
3. Procmail.....	543
4. Почтовый клиент Mutt.....	545
5. SMTP.....	547
Руководство по предварительному связыванию в Gentoo Linux.....	549
1. Введение.....	549
Что такое предварительное связывание (Prelink) и как оно может помочь мне? .....	549
Резюме.....	549
2. Инициализация предварительного связывания.....	550
Установка программ.....	550
Настройка.....	551
3. Предварительное связывание.....	551
Применение предварительного связывания.....	551

4. Известные проблемы и методы их решения.....	552
5. Заключение.....	554
Руководство по файловой системе для устройств.....	555
1. Что такое devfs?.....	555
Проблемы.....	556
Решения.....	556
2. Навигация через дерево устройств.....	557
3. Администрирование дерева устройств.....	558
4. Вопросы, относящиеся к правам доступа.....	559
Русское WIKI.....	562
Настройка дисковой подсистемы.....	562
Тонкая настройка IDE дисков с помощью hdparm.....	577
Работа с CD/DVD.....	581
Создание DVD.....	585
Система Portage.....	592
Portage через NFS.....	592
Настройка Portage.....	596
Portage Overlay.....	600
Экономия трафика с помощью deltap.....	603
Обновление портежей без доступа в интернет непосредственно из системы.....	604
Обновление пакетов без доступа в интернет непосредственно из системы.....	605
Полное обновление системы.....	607
Сборка на другом компьютере.....	611
Прочее.....	612
Разделы MS Windows по-русски.....	612
Udev.....	615
Использование CFLAGS для оптимизации собранных программ.....	617
Редактор nano.....	620
Udev и автоматическое монтирование носителей.....	621
Оптимизация glibc.....	624
Подключение USB-flash.....	629
Чтение-запись NTFS с использованием драйвера ntfs-3g.....	632
Сетевые сервисы.....	634
Установка Apache2.....	634
Настройка iptables для начинающих.....	639
Подробная настройка iptables.....	645
Установка почтовой системы.....	667
Настройка vsftpd.....	675
Jabber Server.....	678
Выделенный сервер Counter-Strike.....	679
Файл сервер Samba в домене Win2k.....	681
Руководство по настройке Samba в режиме PDC с использованием LDAP.....	685
Мониторинг работы системы.....	701
Настройка X.....	723
Настройка русских шрифтов в X.....	723
Настройка размера шрифтов в X.....	728
Xorg X11 и прозрачность.....	730
Xorg. Переключение раскладок.....	734
Установка Xgl.....	735
XGL.....	737
Автоматическое переключение раскладок.....	762
Cedega 5.1.....	763
Windows-эквиваленты.....	764
Ускорение загрузки системы.....	769

Секреты командной строки.....	773
Тонкость работы ссacche с emerge.....	781
Keymap.....	781
Часто задаваемые вопросы.....	783
2. Введение.....	783
Как произносится слово Gentoo, и что оно означает?.....	783
Что делает Gentoo особенным?.....	783
3. Установка.....	783
Все работает очень нестабильно, мои флаги оптимизации "-O9 -ffast-math -fomit-frame-pointer". В чем проблема? .....	783
Как изменить пароль root (или любого другого пользователя)?.....	784
Как добавить обычного пользователя?.....	784
Почему пользователь не может получить права root командой su?.....	784
Как отключить devfs?.....	785
Могу ли я обновить Gentoo без переустановки? .....	785
Ядро не загружается (правильно), что мне теперь делать?.....	785
Мой прокси-сервер требует аутентификации, что делать?.....	786
Как записать ISO файл на компакт-диск?.....	786
Какой диск/стадию надо использовать с моим процессором?.....	787
Я не могу выйти в интернет после перезагрузки. В чем может быть дело? .....	787
Я хочу загрузить Windows из GRUB или LILO, но вижу только черный экран. Что делать? .....	788
Как установить Gentoo, используя файл Stage1 или Stage2?.....	789
4. Управление пакетами.....	789
В каком формате хранятся пакеты?.....	789
Я хочу выполнить сценарий ./configure сам. Возможно ли это?.....	790
Как использовать emerge, если я за межсетевым экраном?.....	790
Что делать, если rsync мне не подходит?.....	790
У меня дома лишь слабое модемное подключение. Могу ли я скачать исходные коды в другом месте, а затем добавить их в систему? .....	790
Исходные коды хранятся в /usr/portage/distfiles. Безопасно ли удалить их? ....	791
Что находится в /var/tmp/portage? Можно ли удалить файлы и каталоги из /var/tmp/portage? .....	791
5. Использование.....	791
Как установить международную раскладку клавиатуры?.....	791
Почему мой пользователь не может использовать свой файл crontab?.....	792
Как включить numlock при загрузке?.....	792
Как сделать очистку терминала при выходе?.....	792
Не удается запускать X приложения из-под root'a (после su).....	793
6. Сопровождение.....	793
ReiserFS и порча файловых систем - как исправить, и т.п.....	793
7. Разработка.....	793
Где оставлять сообщения об ошибках?.....	793
Как часто появляются новые стабильные выпуски?.....	793
Мой динамик орет как сумасшедший. Как выключить динамик в консоли? ....	794
8. Ресурсы.....	794
Где найти информацию о Gentoo Linux?.....	794
Могу ли я купить CD с Gentoo Linux?.....	794
В этом списке нет ответа на мой вопрос. Что делать?.....	795
Наивные вопросы юных линуксоидов.....	795
Вступление.....	795
Как ставить программы? .....	795
Что еще за тарболл? .....	796
А кто такой root и как им стать?.....	797

Основные команды.....	798
Команда man.....	798
Команда cd.....	798
Команда ls.....	799
Команда cp.....	799
Команда rm.....	799
Команда df.....	800
А как мне быть если путь или имя очень длинные?.....	800
Как сделать чтобы программа запущенная из терминала работала после его закрытия?.....	800
А как работать с буфером?.....	801
Как заставить иксы стартовать при загрузке?.....	801
Термины.....	802
Вики (WiKi) (от гавайского: быстрый)— .....	802
ALSA, или Advanced Linux Sound Architecture — .....	802
X сервер (или Иксы) - .....	802
ebuild- .....	803
Оверлей - .....	803
MAN (справки по командам).....	814
Emerge.....	814
Примеры конфигурационных файлов.....	830
1. Мой make.conf.....	830
2. Файл xorg.conf для NVIDIA Geforce.....	832
3. Файл xorg.conf для ATI radeon:.....	836
4. Файл /etc/fstab.....	842
Важные команды Gentoo Linux.....	843
Полезные утилиты Gentoo Linux.....	844
Ресурсы с документацией.....	846
Ресурсы с программным обеспечением.....	846
Каналы IRC (freenode.net).....	846





# Новое и измененное (changelog)

## Версия 1.1 от 15.07.2007

1. По просьбе Mati\_maniak (linuxforum.ru):
  - а) emerge --depclean world - исправлено на emerge --depclean;
  - б) --oneshot (-i) – исправлено на --oneshot.
2. Дополнительное форматирование текста.
3. Добавлен сборник статей: [Linux не для идиотов](#), автор: Артем Капитула (no-dashi, dalth & viking).
4. Добавлен: [список IRC каналов](#).
5. Добавил: [примеры моих конфигурационных файлов](#).

## Версия 1.2 от 01.09.2007

1. Полностью переработан раздел по [обновлению Gentoo](#).
2. Произведено значительное форматирование текста.
3. Обновился раздел: [Полезные утилиты Gentoo Linux](#)
4. Полностью оформлен и дополнен [Handbook](#) !
5. По просьбе **inhab** (linuxforum.ru):  
добавлена статья: [Настройка Samba в режиме PDC с использованием LDAP](#).
6. Добавлено: [Руководство по настройке GNOME](#).
7. Добавлено: [HOWTO fbsplash](#)

## Версия 1.3 от 01.11.2007

1. Изменены цели написания данного сборника.
2. Появилось вступительное слово. :)
3. По просьбе **МАА** (<http://linuxforum.ru>) включена информация о Linux для новичков.
4. Сборник "Linux не для идиотов" перенесён в начало сборника !
5. Добавлен собственный перевод Wiki-статьи: [HOWTO Xorg and Fonts](#).
6. Оформлен новый раздел: "[Gentoo и GPRS](#)". Спасибо следующим авторам:
  - Nikoli (Linuxforum.ru).
  - Дмитрию a.k.a. doc (<http://doc.3dn.ru>).
  - Валентину Синицину.
7. Перенесены [XkbOptions](#).
8. Оформлен раздел: "[Gentoo и X.Org](#)"
9. Создан новый раздел: "[Файловые системы Linux](#)", состоящий из двух статей:
  - [Создание файловых систем](#)
  - [Linux и файловые системы: еще раз о проблеме выбора](#)Отдельное **СПАСИБО** Алексею Федорчуку ([alv@linux-online.ru](mailto:alv@linux-online.ru))
10. Заново импортирована: "[Настольная книга по безопасности Gentoo](#)" т.к. безопасность это очень важный вопрос. Теперь на русском языке.
11. Обновлено: "[Примеры конфигурационных файлов](#)"
12. Обновлено и дополнен: "[HOWTO Соединение с Internet](#)"

13. Продолжено форматирование текста.

## Составители:

- Батогов Е.В. а.к.а. JohnBat26 (ICQ(198710313)/Jabber.ru/Yahoo/AIM/Google/GG)  
e-mail: [JohnBat26@gmail.com](mailto:JohnBat26@gmail.com), [JohnBat26@yandex.ru](mailto:JohnBat26@yandex.ru)  
occupation: JEE developer  
обо мне:
  - <http://picasaweb.google.ru/JohnBat26>
  - <http://www.odnoklassniki.ru/user/182870623>
  - <http://evgeniy-batogov.moikrug.ru/>
- Конахин Э.В. а.к.а. BadEd (ICQ(757727756) – BadEd)  
e-mail: [baded@mail.ru](mailto:baded@mail.ru)
- Киселев М.А. а.к.а. Kismih05 (ICQ(77833996 – Mishel)/Yahoo/AIM)  
e-mail: [kismih05@yahoo.com](mailto:kismih05@yahoo.com)
- Dmeshr  
e-mail: [dmeshr@gmail.com](mailto:dmeshr@gmail.com)

### По материалам сайтов:

- <http://gentoo.ru>
- <http://gentoo.org>
- <http://ru.gentoo-wiki.com>
- <http://linuxforum.ru>
- <http://www.rugentoo.org>
- <http://myfotomx.com/dalth/linuxbook.odt>
- <http://linuxshop.ru>
- <http://citkit.ru>

Версия: 1.3

Последняя редакция: 01.11.2007

Лицензия: GPL v2

Создано с помощью *OpenOffice Writer 2.3.0. (Infra)*

## Цели данного сборника

- *Избавить от необходимости серфинга по различным ресурсам в поисках информации о Linux и Gentoo в частности.*
- *Увеличить количество пользователей Gentoo Linux в России.*
- *Создать лучшую документацию по Gentoo Linux на русском языке.*

*Пожелания, участие в написании и критика приветствуются. :)*

Просьба при копировании или дополнении данного сборника, согласовывать с:

1. Авторами статей.
2. Составителями данного сборника (см. вверху страницы).

***Всем авторам статей: ОГРОМНОЕ СПАСИБО ЗА ВАШ ТРУД !!!***

## Вступительное слово

**Автор:** Батогов Е.В. a.k.a JohnBat26

Добро пожаловать всем, кто решил вступить на путь освоения операционной системы Linux (и Gentoo в особенности). Немного забегаая вперед, хочется заметить, что путь предстоит не легкий, но очень интересный. Главное помните одну известную истину: "Лучше день подождать, а потом за 5 минут долететь!"

Расскажу немного из своего опыта использования Linux.

Когда 3 года назад мне надоело переустанавливать Windows по два и более раза в год, я стал искать альтернативу. Кто-то скажет сейчас, что мол я не умею использовать Windows, сразу отвечу, что работал в ней около 5 лет. Так как я являюсь человеком, который не может просто так сидеть и не проводить эксперименты с системой, то устанавливал более 200 программ, и это делало Windows очень медленной и нежизнеспособной (Сейчас у меня в Gentoo установлено > 1000 пакетов :)). Далее, в Windows, автоматически возникали проблемы обновления всего этого ПО. И это еще не говоря о вирусах...

Сначала я поставил SuSE Linux. В принципе все нравилось, НО это до необходимости совершения таких действий как, например, смена ядра, установка драйверов, обновление системы (хотя Smart в принципе, справлялся с этим).

Установка программ из RPM вызывала напряжение нервной системы, особенно когда не было хорошего интернета. Ох уж эти зависимости.... :((

Так как я являюсь человеком, который не остановится, пока не получит систему, близкую к идеальной, то мой взгляд, наконец-то, пал на Gentoo. Спустя год я уже установил Gentoo на 9 машин, из которых половина ноутбуки (Dell, Asus, Acer, Sony VAIO)! Процесс установки действительно захватывает тем, что ты полностью понимаешь, что полностью управляешь системой. Gentoo поразил меня следующим:

- очень быстрая скорость работы;
- ИМНО, лучшая системы управления ПО — portage;
- очень много документации, в том числе и на русском языке;
- никаких проблем с обновлением всей системы, включая ядро, gcc, glibc и т.п.;
- отличное сообщество пользователей;
- абсолютная контролируемость системы;
- никаких лишних пакетов.

Еще ни одна ОС не вдохновляла меня на написание книги, пусть она и составлена из трудов многих хороших людей. В нашей Великой России такие сборники должны быть очень актуальны, т.к. не у всех есть постоянный доступ к сети (сам работал через GPRS в течение двух лет).

В заключении хочется пожелать всем удачи на пути освоения Gentoo Linux. Как говорится: "Дорогу осилит идущий !" Вперед !!!

# Вводная информация о Linux

## UNIX

**Ссылка на оригинал:** <http://ru.wikipedia.org/wiki/UNIX>

**С версии: 1.3**

**UNIX** (читается юникс) — группа переносимых, многозадачных и многопользовательских операционных систем.

Первая система UNIX была разработана в конце 1960-х — начале 1970-х годов в подразделении Bell Labs компании AT&T. С тех пор было создано большое количество различных UNIX-систем. Юридически лишь некоторые из них имеют полное право называться «UNIX»; остальные же, хотя и используют сходные концепции и технологии, объединяются термином «UNIX-подобные» (англ. Unix-like). Для краткости в данной статье под UNIX-системами подразумеваются как истинные UNIX, так и UNIX-подобные ОС.

Некоторые отличительные признаки UNIX-систем включают в себя:

- использование простых текстовых файлов для настройки и управления системой;
- широкое применение командной строки;
- представление устройств и некоторых средств межпроцессного взаимодействия как файлов;
- использование конвейеров из нескольких программ, каждая из которых выполняет одну задачу.

В настоящее время UNIX используются в основном на серверах, а также как встроенные системы для различного оборудования. На рынке ОС для рабочих станций и домашнего применения UNIX уступили другим операционным системам, в первую очередь Microsoft Windows. UNIX-системы имеют большую историческую важность, поскольку благодаря им распространились некоторые популярные сегодня концепции и подходы в области ОС и программного обеспечения.

## История

### Предшественники

В 1957 году в Bell Labs была начата работа по созданию операционной системы для собственных нужд. Под руководством Виктора Высотского (русского по происхождению) была создана система BESYS. Впоследствии он возглавил проект Multics, а затем стал главой информационного подразделения Bell Labs.



В 1964 году появились компьютеры третьего поколения, для которых возможности BESYS уже не подходили. Высотский и его коллеги приняли решение не разрабатывать новую собственную операционную систему, а подключиться к совместному проекту General Electric и Массачусетского технологического института Multics. Телекоммуникационный гигант AT&T, в состав которого входили Bell Labs,

оказал проекту существенную поддержку, но в 1969 году вышел из проекта, поскольку он не приносил никаких финансовых выгод.

## Первые UNIX

Первоначально UNIX была разработана в конце 1960-х годов сотрудниками Bell Labs, в 1969 году Кен Томпсон, стремясь реализовать идеи, что были положены в основу MULTICS, но на более скромном аппаратном обеспечении (DEC PDP-7), написал первую версию новой операционной системы, а Брайан Керниган придумал для неё название — UNICS (UNIplexed Information and Computing System) — в противовес MULTICS (MULTIplexed Information and Computing Service). Позже это название сократилось до UNIX. В 1971 году вышла версия для PDP-11, наиболее успешного семейства миникомпьютеров 1970-х (в СССР оно известно как СМ ЭВМ). Эта версия получила название «первая редакция» (Edition 1) и была первой официальной версией. Системное время все реализации UNIX отсчитывают с 1 января 1970.

Первые версии UNIX были написаны на ассемблере и не имели встроенного компилятора с языка высокого уровня. Примерно в 1969 году Кен Томпсон при содействии Дениса Ритчи разработал и реализовал язык Би (B), представлявший собой упрощённый (для реализации на миникомпьютерах) вариант разработанного в 1966 языка BCPL. Би, как и BCPL, был интерпретируемым языком. В 1969—1973 годах на основе Би был разработан компилируемый язык, получивший название Си (C).

В 1973 году вышла третья редакция UNIX, со встроенным компилятором с Си. 15 октября того же года появилась четвёртая редакция, с переписанным на Си системным ядром (в духе системы Multics, также написанной на языке высокого уровня (ПЛ/1)), а в 1975 — пятая редакция, полностью переписанная на Си. С 1974 года UNIX стал бесплатно распространяться среди университетов и академических учреждений. С 1975 года началось появление новых версий, разработанных за пределами Bell Labs, и рост популярности системы.

В том же 1975 году вышла шестая редакция, известная по широко разошедшимся комментариям Джона Лайонса (Lions' Commentary on UNIX 6th Edition, with Source Code, [1], [2]). К 1978 г. система была установлена более чем на 600 машинах, прежде всего, в университетах. Седьмая редакция была последней единой версией UNIX. Именно в ней появился близкий к современному интерпретатор командной строки Bourne shell.ую очередь Кеном Томпсоном, Денисом Ритчи и Дугласом МакИлроем.

## Раскол

С 1978 года начинает свою историю BSD UNIX, созданный в университете Беркли. Его первая версия, была основана на шестой редакции. В 1979 выпущена новая версия, названная 3BSD, основанная на седьмой редакции. BSD поддерживал такие полезные свойства, как виртуальную память и замещение страниц по требованию. Автором BSD был Билл Джой.

В начале 1980-х компания AT&T, которой принадлежали Bell Labs, осознала ценность UNIX и начала создание коммерческой версии UNIX. Эта версия,



поступившая в продажу в 1982 году, носила название UNIX System III и была основана на седьмой версии системы.

Важной причиной раскола UNIX стала реализация в 1980 г. стека протоколов TCP/IP. До этого межмашинное взаимодействие в UNIX пребывало в зачаточном состоянии — наиболее существенным способом связи был UUCP (средство копирования файлов из одной UNIX-системы в другую, изначально работавшее по телефонным сетям с помощью модемов).

Было предложено два интерфейса программирования сетевых приложений: Berkley sockets и интерфейс транспортного уровня TLI (англ. Transport Layer Interface). Интерфейс Berkley sockets был разработан в университете Беркли и использовал стек протоколов TCP/IP, разработанный там же. TLI был создан AT&T в соответствии с определением транспортного уровня модели OSI и впервые появился в системе System V версии 3. Хотя эта версия содержала TLI и потоки, первоначально в ней не было реализации TCP/IP или других сетевых протоколов, но подобные реализации предоставлялись сторонними фирмами. Реализация TCP/IP официально и окончательно была включена в базовую поставку System V версии 4. Это, как и другие соображения (по большей части, рыночные), вызвало окончательное размежевание между двумя ветвями UNIX — BSD (университета Беркли) и System V (коммерческая версия от AT&T). Впоследствии, многие компании, лицензировав System V у AT&T, разработали собственные коммерческие разновидности UNIX, такие, как AIX, HP-UX, IRIX, Solaris.

В середине 1983 года была выпущена версия 4.2BSD, поддерживающая работу в сетях Ethernet и Arpanet. Система стала весьма популярной. Между 1983 и 1990 в BSD были добавлено много новых возможностей, таких как отладчик ядра, сетевая файловая система NFS, виртуальная файловая система VFS, и существенно улучшены возможности работы с файловыми сетями.

Тем временем AT&T выпускала новые версии своей системы, названной System V. В 1983 была выпущена версия 1 (SVR1 — System V Release 1), включавшая полноэкранный текстовый редактор vi, библиотеку curses, буферизацию ввода-вывода, кеширование inode. Версия 2 (SVR2), выпущенная в 1984, реализовывала монопольный доступ к файлам (file locking), доступ к страницам по требованию (demand paging), копирование при записи (copy-on-write). Версия 3 вышла в 1987 году и включала, среди прочего, TLI, а также систему поддержки удалённых файловых систем RFS. Версия 4 (SVR4), разработанная в сотрудничестве с фирмой Sun и вышедшая 18 октября 1988, поддерживала многие возможности BSD, в частности TCP/IP, сокеты, новый командный интерпретатор csh. Кроме того, там было много других добавлений, таких как символические ссылки, командный интерпретатор ksh, сетевая файловая система NFS (заимствованная у SunOS) и т. д.

Современные реализации UNIX, как правило, не являются системами V или BSD в чистом виде. Они реализуют возможности как System V, так и BSD.

## **Свободные UNIX-подобные операционные системы**

В 1983 году Ричард Столлман объявил о создании проекта GNU — попытки создания свободной UNIX-подобной операционной системы с нуля, без использования оригинального исходного кода. Большая часть программного обеспечения, разработанного в рамках данного проекта, — такого, как GNU toolchain, Glibc (стандартная библиотека языка Си) и Coreutils — играет ключевую роль в других свободных операционных системах. Однако работы по созданию замены для ядра UNIX, необходимые для полного выполнения задач GNU, продвигались крайне медленно. В настоящее время GNU Hurd — попытка создать современное ядро на основе микроядерной архитектуры Mach — всё ещё далека от завершения.

В 1991 году, когда Линус Торвалдс опубликовал ядро Linux и привлёк помощников, использование инструментов, разработанных в рамках проекта GNU, было очевидным выбором. Операционная система GNU и ядро Linux вместе составляют ОС, известную, как GNU/Linux. Дистрибутивы этой системы (такие как Red Hat и Debian), включающие ядро, утилиты GNU и дополнительное программное обеспечение стали популярными как среди любителей, так и среди представителей бизнеса.

В результате урегулирования юридического дела, возбуждённого UNIX Systems Laboratories против университета Беркли и Berkeley Software Design Inc., было установлено, что университет может распространять BSD UNIX, в том числе и бесплатно. После этого были возобновлены эксперименты, связанные с BSD-версией UNIX. Вскоре разработка дистрибутива BSD была продолжена в нескольких направлениях одновременно, что привело к появлению проектов, известных как FreeBSD, NetBSD, OpenBSD, TrustedBSD и DragonFlyBSD.

В настоящий момент GNU/Linux и представители семейства BSD быстро отвоевывают рынок у коммерческих UNIX-систем и одновременно проникают как на настольные компьютеры конечных пользователей, так и на мобильные и встраиваемые системы. Одним из свидетельств данного успеха служит тот факт, что, когда фирма Apple искала основу для своей новой операционной системы, она выбрала NEXTSTEP — операционную систему со свободно распространяемым ядром, разработанную фирмой NeXT и переименованную в Darwin после приобретения фирмой Apple. Данная система относится к семейству BSD и основана на ядре Mach. Применение Darwin BSD UNIX в Mac OS X делает его одной из наиболее широко используемых версий UNIX.

## **Современность**

После разделения компании AT&T, товарный знак UNIX и права на оригинальный исходный код неоднократно меняли владельцев, в частности, длительное время принадлежали компании Novell.

В 1993 году Novell передала права на товарный знак и на сертификацию программного обеспечения на соответствие этому знаку консорциуму X/Open, который затем объединился с Open Software Foundation, образовав консорциум The Open Group. Он объединяет ведущие компьютерные корпорации и государственные организации, в том числе IBM, Hewlett-Packard, Sun, NASA и многие другие. Консорциум занимается разработкой открытых стандартов в области операционных систем, самым важным из которых является Single UNIX Specification, ранее известный как POSIX. С точки зрения The Open Group, название UNIX могут носить только системы, прошедшие сертификацию на соответствие Single UNIX Specification.

В 1995 году Novell продала права на существующие лицензии и дальнейшую разработку System V компании Santa Cruz Operation. В 2000 году Santa Cruz Operation продала свой UNIX-бизнес компании Caldera, которая затем была переименована в SCO Group. Хотя это название похоже на аббревиатуру SCO, используемую Santa Cruz Operation, это две разные компании.

SCO Group заявила, что она также обладает правами на исходный код UNIX и развернула кампанию против различных пользователей и поставщиков UNIX-подобных систем, требуя выплаты лицензионных отчислений. Однако Novell утверждает, что права на исходный код не были переданы Santa Cruz Operation и, таким образом, не перешли к SCO Group, а остаются у Novell, что и подтвердил вердикт суда.

## **Влияние UNIX**

Идеи, заложенные в основу UNIX, оказали огромное влияние на развитие компьютерных операционных систем. В настоящее время UNIX-системы признаны одними из самых исторически важных ОС.

Как и Multics, UNIX была написана на языке высокого уровня, а не на ассемблере (доминировавшем в то время).

Она содержала значительно упрощённую, по сравнению с современными ей операционными системами, файловую модель. Файловая система включала как службы, так и устройства (такие как принтеры, терминалы и жёсткие диски) и предоставляла внешне единообразный интерфейс к ним, но дополнительные механизмы работы с устройствами (такие как IOCTL и биты доступа) не вписывались в простую модель «поток байтов».

UNIX популяризовала предложенную в Multics идею иерархической файловой системы с произвольной глубиной вложенности. Другие операционные системы того времени позволяли разбивать дисковое пространство на каталоги или разделы, но число уровней вложенности было фиксировано и, зачастую, уровень вложенности был только один. Позднее все основные фирменные операционные системы обрели возможность создания рекурсивных подкаталогов, также заимствованную из Multics.

То, что интерпретатор команд стал просто одной из пользовательских программ, а в качестве дополнительных команд выступают отдельные программы, является ещё одной инновацией Multics, популяризированной UNIX. Язык командной оболочки UNIX используется пользователем как для интерактивной работы, так и для написания скриптов, то есть не существует отдельного языка описания заданий, как, например, в системе JCL фирмы IBM. Так как оболочка и команды операционной системы являются обычными программами, пользователь может выбирать их в соответствии со своими предпочтениями, или даже написать собственную оболочку. Наконец, новые команды можно добавлять к системе без перекомпиляции ядра. Новый, предложенный в командной строке UNIX, способ создания цепочек программ, последовательно обрабатывающих данные, способствовал использованию параллельной обработки данных.

Существенными особенностями UNIX были полная ориентация на текстовый ввод-вывод и предположение, что размер машинного слова кратен восьми битам. Первоначально в UNIX не было даже редакторов двоичных файлов — система полностью конфигурировалась с помощью текстовых команд. Наибольшей и наименьшей единицей ввода-вывода служил текстовый байт, что полностью отличало ввод-вывод UNIX от ввода-вывода других операционных систем, ориентированного на работу с записями. Ориентация на использование текста для представления всего, что только можно, сделала полезными т. н. конвейеры (англ. pipelines). Ориентация на текстовый восьмибитный байт сделала UNIX более масштабируемой и переносимой, чем другие операционные системы. Со временем текстовые приложения одержали победу и в других областях, например, на уровне сетевых протоколов, таких как Telnet, FTP, SMTP, HTTP и других.

UNIX способствовала широкому распространению регулярных выражений, которые были впервые реализованы в текстовом редакторе ed для UNIX. Возможности, предоставляемые UNIX-программам, стали основой стандартных интерфейсов операционных систем (POSIX).

Широко используемый в системном программировании язык Си, созданный изначально для разработки UNIX, превзошёл UNIX по популярности. Си был первым «веротерпимым» языком, который не пытался навязать программисту тот или иной стиль программирования. Си был первым высокоуровневым языком, предоставляющим доступ ко всем возможностям процессора, таким как ссылки, таблицы, битовые сдвиги, приращения и т. п. С другой стороны, свобода Си приводила к ошибкам переполнения буфера в таких функциях стандартной

библиотеки Си, как `gets` и `scanf`. Результатом стали многие печально известные уязвимости, например, та, что эксплуатировалась в знаменитом черве Морриса. Первые разработчики UNIX способствовали внедрению принципов модульного программирования и повторного использования в инженерную практику. UNIX предоставлял возможность использования протоколов TCP/IP на сравнительно недорогих компьютерах, что привело к быстрому росту Интернета. Это, в свою очередь, способствовало быстрому обнаружению нескольких крупных уязвимостей в системе безопасности, архитектуре и системных утилитах UNIX. Со временем ведущие разработчики UNIX разработали культурные нормы разработки программного обеспечения, которые стали столь же важны, как и сам UNIX (см. Философия UNIX).

## Стандарты

Большое количество разных вариантов системы UNIX привело к необходимости стандартизовать её, чтобы упростить переносимость приложений и избавить пользователя от необходимости изучать особенности каждой разновидности UNIX. С этой целью ещё в 1980 была создана пользовательская группа /usr/group. Самые первые стандарты были разработаны в 1984—1985 гг. В настоящее время наиболее важными являются следующие стандарты:

- **POSIX 1003.1-1988**, определяющий программный интерфейс приложений (API, Application Programming Interface). Он используется не только в UNIX, но и в других операционных системах. В 1990 он был принят институтом IEEE как IEEE 1003.1-1990.
- **POSIX 1003.2-1992**, определяющий поведение утилит, в том числе командного интерпретатора.
- **POSIX 1003.1b-1993**, дополняющий POSIX 1003.1-1988. Определяет поддержку систем реального времени.
- **POSIX 1003.1c-1995**, дополняющий POSIX 1003.1-1988. Определяет нити (threads), известные также как pthreads.

### Все стандарты POSIX собраны в документе IEEE 1003.

В начале 1990-х годов The Open Group предложила другой, похожий на POSIX стандарт — Common API Specification, или Spec 1170. Стандарт приобрёл большую популярность, чем POSIX, поскольку был доступен бесплатно, в то время как IEEE требовало немалую плату за доступ к своему стандарту.

В 1998 году были начаты работы по объединению данных стандартов. Благодаря этому в настоящее время данные стандарты почти идентичны. Совместный стандарт называется Single UNIX Specification Version 3 и доступен бесплатно в интернете [3].

В целях совместимости несколько создателей UNIX-систем предложили использовать ELF-формат систем SVR4 для двоичных и объектных файлов. Единый формат полностью обеспечивает соответствие двоичных файлов в рамках одной компьютерной архитектуры.

Структура каталогов некоторых систем, в частности, GNU/Linux, определена в стандарте Filesystem Hierarchy Standard. Однако во многих отношениях этот тип стандарта является спорным, и он, даже внутри сообщества GNU/Linux, далеко не универсален.

## Канонические команды UNIX

- Создание и навигация по файлам и каталогам: ls, mv, rm, cp, ln, CD, pwd, mkdir, rmdir, find, du, df.
- Просмотр и редактирование файлов: touch, more, less, ed, ex, vi, emacs.
- Обработка текста: echo, cat, grep, sort, uniq, sed, awk, tee, head, tail, cut, tr, split, printf.
- Сравнение файлов: comm, cmp, diff, patch.
- Разнообразные утилиты командного интерпретатора: yes, test, xargs, expr.
- Системное администрирование: chmod, chown, ps, su, w, who, df, mount, umount.
- Коммуникации: mail, telnet, ftp, finger, ssh.
- Командные оболочки: sh, bash, csh, ksh, tcsh, zsh.
- Работа с исходным кодом и объектным кодом: cc, gcc, ld, nm, yacc, bison, lex, flex, ar, ranlib, make.
- Сжатие и архивация: compress, uncompress, gzip, gunzip, tar.
- Работа с двоичными файлами: od, strings.

Ниже приведён список 60 команд из раздела 1 первой версии UNIX:

ar, as, b, bas, bcd, boot, cat, chdir, check, chmod, chown, cmp, cp, date, db, dbppt, dc, df, dsw, dtf, du, ed, find, for, form, hup, lbppt, ld, ln, ls, mail, mesg, mkdir, mkfs, mount, mv, nm, od, pr, rew, rkd, rkf, rkl, rm, rmdir, roff, sdate, sh, stat, strip, su, sum, tap, tm, tty, type, un, wc, who, write

# GNU/Linux

Ссылка на оригинал: <http://ru.wikipedia.org/wiki/Линукс>

С версии: 1.3

**GNU/Linux** (произносится «гну слэш линукс») — свободная UNIX-подобная операционная система. Она основана на системных программах, разработанных в рамках проекта GNU, и на ядре Linux. Зачастую, по историческим причинам и для краткости, эту систему называют просто «Linux».

GNU/Linux работает на PC-совместимых системах семейства Intel x86, а также на IA-64, AMD64, PowerPC, ARM и многих других.

К операционной системе GNU/Linux также часто относят программы, дополняющие эту операционную систему, и прикладные программы, делающие её полноценной многофункциональной операционной средой.

В отличие от большинства других операционных систем, GNU/Linux не имеет единой «официальной» комплектации. Вместо этого GNU/Linux поставляется в большом количестве так называемых дистрибутивов, в которых программы GNU соединяются с ядром Linux и другими программами. Наиболее известными дистрибутивами GNU/Linux являются Slackware, Red Hat, Fedora, Mandriva, SuSE, Debian, Gentoo, Ubuntu. Из дистрибутивов российских разработчиков наиболее известны ALT Linux и ASPLinux.

В отличие от Microsoft Windows, Mac OS (Mac OS X) и коммерческих UNIX-подобных систем, GNU/Linux не имеет географического центра разработки. Нет и организации, которая владела бы этой системой; нет даже единого координационного центра. Программы для GNU/Linux — результат работы тысяч проектов. Некоторые из этих проектов централизованы, некоторые сосредоточены в фирмах, но большинство объединяют программистов со всего света, которые знакомы только по переписке. Создать свой проект или присоединиться к уже существующему может любой и, в случае успеха, результаты работы станут известны миллионам пользователей. Пользователи принимают участие в тестировании свободных программ, общаются с разработчиками напрямую, что позволяет быстро находить и исправлять ошибки и реализовывать новые возможности.

Именно такая гибкая и динамичная система разработки, невозможная для проектов с закрытым кодом, определяет исключительную экономическую эффективность GNU/Linux. Низкая стоимость свободных разработок, отложенные механизмы тестирования и распространения, привлечение людей из разных стран, обладающих разным видением проблем, защита кода лицензией GPL — всё это стало причиной успеха свободных программ. [1]

Конечно, такая высокая эффективность разработки не могла не заинтересовать крупные фирмы, которые стали открывать свои проекты. Так появились Mozilla (Netscape, AOL), OpenOffice.org (Sun), свободный клон Interbase (Borland) — Firebird, SAP DB (SAP). IBM способствовала переносу GNU/Linux на свои мейнфреймы.

С другой стороны, открытый код значительно снижает себестоимость разработки закрытых систем для GNU/Linux и позволяет снизить цену решения для пользователя. Вот почему GNU/Linux стала платформой, часто рекомендуемой для таких продуктов, как Oracle Database, DB2, Informix, SyBase, SAP R3, Domino.

Сообщество GNU/Linux поддерживает связь посредством групп пользователей Linux. Именно такая гибкая и динамичная система разработки, невозможная для проектов с закрытым кодом, определяет исключительную экономическую эффективность GNU/Linux. Низкая стоимость свободных разработок, отложенные механизмы тестирования и распространения, привлечение людей из разных стран, обладающих разным видением проблем, защита кода лицензией GPL — всё это стало причиной успеха свободных программ. [1]

Конечно, такая высокая эффективность разработки не могла не заинтересовать крупные фирмы, которые стали открывать свои проекты. Так появились Mozilla (Netscape, AOL), OpenOffice.org (Sun), свободный клон Interbase (Borland) — Firebird, SAP DB (SAP). IBM способствовала переносу GNU/Linux на свои мейнфреймы.

С другой стороны, открытый код значительно снижает себестоимость разработки закрытых систем для GNU/Linux и позволяет снизить цену решения для пользователя. Вот почему GNU/Linux стала платформой, часто рекомендуемой для таких продуктов, как Oracle Database, DB2, Informix, SyBase, SAP R3, Domino.

Сообщество GNU/Linux поддерживает связь посредством групп пользователей Linux.





# Linux (ядро)

Ссылка на оригинал: [http://ru.wikipedia.org/wiki/Linux\\_\(ядро\)](http://ru.wikipedia.org/wiki/Linux_(ядро))

С версии: 1.3

**Linux (Линукс)** — ядро операционной системы, разработка которого была начата финским студентом Линусом Торвальдсом в 1991 году.

В большинстве своём код написан на Си с некоторыми расширениями GNU C и на ассемблере (с использованием синтаксиса GNU Assembler AT&T).

Распространяется свободно на условиях GNU General Public License.

Торговая марка Linux зарегистрирована на Линуса Торвальдса.

## История

Начало проекту было положено в 1991 году с публикации сообщения в новостной группе Usenet comp.os.minix] следующего содержания[1]:

Привет всем, кто использует миникс — Я делаю (бесплатную) операционную систему (всего лишь хобби, не будет большой и профессиональной как gnu) для клонов 386 (486) AT...

К тому времени проект GNU уже создал множество составляющих для свободной операционной системы, но их ядро GNU Hurd ещё не было готово. BSD-системы в то время не могли быть использованы по юридическим причинам, связанным с использовавшимися лицензиями. Поэтому пустующее место ядра для свободной ОС занял Linux и, несмотря на ограниченную функциональность ранних версий, привлёк к себе множество разработчиков и пользователей.

**Linux** — это название только ядра, а не операционной системы. Часто системы, основанные на ядре Linux, называют просто Linux, но большинство из них на самом деле называются GNU/Linux, так как состоят из ядра Linux и множества системных библиотек и программ, написанных в рамках проекта GNU.

## Хронология

\* Апрель 1991 — 21-летний Линус Торвальдс начал работу над некоторыми механизмами операционной системы. Он начал с эмулятора терминала и планировщика задач.

\* 25 августа 1991 — Линус поместил следующее сообщение (русский перевод оригинального сообщения[1] на английском):

From: torvalds@klaava.Helsinki.Fi (Линус Бенедикт Торвальдс)  
Newsgroups: comp.os.minix  
Subject: Маленький опрос о моей новой операционной системе  
Message-ID:<1991Aug25.205708.9541@klaava.Helsinki.Fi>  
Date: 25 Aug 91 20:57:08 GMT  
Organization: Хельсинский Университет

Привет всем, кто использует миникс - Я делаю (бесплатную) операционную

систему (всего лишь хобби, не будет большой и профессиональной как gnu) для клонов 386 (486) АТ. Она ваялась с апреля, и скоро будет готова. Я хочу отзывов о том, что людям нравится/не нравится в миниксе, ибо моя система на неё похожа(такое же устройство файловой системы(по практическим соображениям) среди всего прочего).

Я уже включил bash (1.08) и gcc (1.40), и похоже всё работает. Это значит, что что-то полезное появится через несколько месяцев, и я хотел бы узнать, чего люди хотят. Любые советы принимаются, но я не обещаю, что всё исполню :-)

Линус (torvalds@kruuna.helsinki.fi)

PS. Да, у неё никакого миниксового кода, и многозадачная фс. Она НЕ переносима (применяет переключение задач 386-го, итп.), и скорее всего будет поддерживать только АТ-винчестеры, так как это всё, что у меня есть :-)

- 17 сентября 1991 — Linux версии 0.01. (10 239 строк кода);
- 5 октября 1991 — Linux версии 0.02 [1](англ.);
- Декабрь 1991 — Linux версии 0.11. Это была первая версия Linux, на которой можно было собрать Linux из исходных кодов;
- 19 января 1992 — Первое сообщение в группе новостей alt.os.linux(англ.);
- 31 марта 1992 — Создана группа новостей comp.os.linux(англ.);
- Апрель 1992 — Linux версии 0.96, на котором стало возможно запустить графический сервер X Window System;
- Весь 1993 и начало 1994 — 15 тестовых релизов версии 0.99.\* (в июле 1993 введено понятие BogoMips);
- 14 марта 1994 — Linux версии 1.0.0 (176 250 строк кода);
- Март 1995 — Linux версии 1.2.0 (310 950 строк кода);
- 9 мая 1996 — Выбран символ Linux — пингвин Tux;
- 9 июня 1996 — Linux версии 2.0.0 (777 956 строк кода);
- 25 января 1999 — Linux версии 2.2.0, изначально довольно недоработанный (1 800 847 строк кода);
- 4 января 2001 — Linux версии 2.4.0 (3 377 902 строк кода);
- 18 декабря 2003 — Linux версии 2.6.0 (5 929 913 строк кода).

## **Версии**

Торвальдс продолжает выпускать новые версии ядра, объединяя изменения, вносимые другими программистами, и внося свои. В дополнение к официальным версиям ядра существуют альтернативные ветки, которые могут быть взяты из различных источников. Как правило, разработчики дистрибутивов GNU/Linux поддерживают свои собственные версии ядра Linux, например, включая в них драйвера устройств, которые ещё не включены в официальную версию.

### **Нумерация версий**

Номер версии ядра Linux в настоящее время содержит четыре числа, следуя недавнему изменению в долго используемой до этого политике схемы версий, основанной на трёх числах. Для иллюстрации допустим, что номер версии составлен

таким образом: A.B.C[D] (например 2.2.1, 2.4.13 или 2.6.12.3).

\* Число A обозначает версию ядра. Оно изменяется наименее часто и только тогда, когда вносятся значительные изменения в код и концепцию ядра. Оно изменялось дважды в истории ядра: в 1994 (версия 1.0) и в 1996 (версия 2.0).

\* Число B обозначает старшую версию ревизии ядра. Чётные числа обозначают стабильные ревизии, то есть те, которые предназначены для промышленного использования, такие как 1.2, 2.4 или 2.6. Нечётные числа обозначают ревизии для разработчиков, такие как 1.1 или 2.5. Они предназначены для тестирования новых улучшений и драйверов до тех пор, пока они не станут достаточно стабильными для того, чтобы быть включёнными в стабильный выпуск.

\* Число C обозначает младшую версию ревизии ядра. В старой трёхчисловой схеме нумерации, оно изменялось тогда, когда в ядро включались заплатки связанные с безопасностью, исправления ошибок, новые улучшения или драйвера. С новой политикой нумерации, однако, оно изменяется только тогда, когда вносятся новые драйвера или улучшения; небольшие исправления поддерживаются числом D.

\* Число D впервые появилось в случае, когда смертельная ошибка, которая требовала незамедлительного исправления, была обнаружена в коде ядра 2.6.8, связанного с NFS. Однако, было недостаточно других изменений, для того чтобы это послужило причиной для выпуска новой младшей ревизии (которой должна была стать 2.6.9). Поэтому была выпущена версия 2.6.8.1 с единственным исправлением в виде исправления для этой ошибки. С ядра 2.6.11, эта нумерация была адаптирована в качестве новой официальной политики версий. Исправления ошибок и заплатки безопасности теперь управляются с помощью четвёртого числа, тогда как большие изменения выполняются в изменениях младшей версии ревизии ядра (число C).

## **Поддержка**

В то время как Торвальдс продолжает выпускать новые экспериментальные версии, руководство «старыми» стабильными версиями передаётся другим лицам:

Версии	Сопроводители
2.0	Дэвид Виенхал
2.2	Марк-Кристиан Петерсон (раньше Элан Кокс)
2.4	Марчело Тозатти
2.6	Эндрю Мортон / Линус Торвальдс

Другими программистами ядра Linux являются Роберт Лав и Инго Молнар. (См. Список сопроводителей Linux (англ.)).

## **Стабильные версии**

\* Версия 1.0 в марте 1994 — поддерживала только однопроцессорные i386-

машины.

- \* Версия 1.2 в марте 1995 — добавлена поддержка процессоров Alpha, SPARC и MIPS.

- \* Версия 2.0 в июне 1996 — добавлена поддержка других процессоров, а также многопроцессорных систем.

- \* Версия 2.2 в январе 1999 — [2](англ.).

- \* Версия 2.4 в январе 2001 — добавлена поддержка ISA Plug and Play, процессоров PA-RISC, шин USB и PC-Card (PCMCIA). Поддержка для процессоров Axis Communications ETRAX CRIS и файловой системы InterMezzo были добавлены чуть позже. [3](англ.)

- \* Версия 2.6 от 17 декабря 2003:

- о встроен uClinux (для микроконтроллеров);

- о добавлена поддержка для процессоров Hitachi серии H8/300, NEC v850, процессоры для встроенных систем Motorola m68k, новая архитектура доступа к памяти NUMA, поддержка NCR Voyager, технологии Intel hyperthreading и PAE;

- о добавлено:

- + поддержка файловой системы XFS SGI;

- + улучшена поддержка APIC;

- + увеличено максимальное количество пользователей и групп с 65 000 до более 4 млрд;

- + увеличено максимальное количество процессов с 32 000 до 1 млрд;

- + увеличено максимальное количество типов устройств (major device) с 255 до 4095 и максимальное количество устройств каждого типа (minor device) с 255 до более миллиона;

- + улучшена поддержка 64-битных систем и поддержка файловых систем размером более 16 Тбайт;

- + улучшено время реакции для процессов реального времени;

- + переписана реализация потоков под использование Native POSIX Thread Library (NPTL);

- + улучшен загрузчик модулей;

- + добавлена новая служебная файловая система sysfs;

- + интеграция User-mode Linux;

- + и др.

## **Архитектура**

Ядро Linux поддерживает многозадачность, виртуальную память, динамические библиотеки, отложенную загрузку, производительную систему управления памятью и многие сетевые протоколы.

На сегодняшний день Linux — монолитное ядро с поддержкой загружаемых модулей. Драйверы устройств и расширения ядра обычно запускаются на «кольце 0», с полным доступом к оборудованию. В отличие от обычных монолитных ядер, драйвера устройств легко собираются в виде модулей и загружаются или выгружаются во время работы системы.

То, что архитектура Linux не является микроядерной, вызвало обширнейшие прения между Линусом Торвальдсом и Эндрю Таненбаумом в конференции comp.os.minix(англ.) в 1992 г.

## Совместимость

Не задуманный изначально как многоплатформенное ядро, Linux на данный момент портирован на очень широкий круг архитектур, запускается на широком спектре оборудования от iPAQ (карманный компьютер) до IBM S/390 (высокопроизводительный мейнфрейм). Linux используется как ядро операционной системы на суперкомпьютере Blue Gene (англ.) фирмы IBM.

Изначально Linux разрабатывался для 32-битных x86-совместимых ПК; на сегодняшний день Linux запускается на следующих процессорных архитектурах:

- \* ARM
  - o Acorn: Archimedes, A5000, RiscPC
  - o StrongARM, Intel XScale и т. п.
  - o HP iPAQ
- \* Axis Communications CRIS
- \* DEC Alpha
- \* HP PA-RISC
- \* Hitachi: SuperH (SEGA Dreamcast), H8/300
- \* IBM S/390
- \* IBM zSeries-мэйнфреймы
- \* Intel 80386 и выше: IBM PC и совместимые с процессорами:
  - o 80386, 80486, а также AMD, Cyrix, TI и IBM-варианты;
  - o серия Pentium;
  - o Core, Core2 Duo в 32 и 64-х битных версиях.
  - o AMD Am5x86, K5, K6, Athlon (все 32-битные версии), Duron;
  - o AMD64: 64-битная технология AMD (также известная как x86-64);
  - o Cyrix 5x86, 6x86 (M1), 6x86MX и MediaGX (National/AMD Geode) серия;
  - o VIA C3 (англ.) и последующие процессоры;
  - o поддержка Intel 8086, 8088, 80186, 80188 и 80286 процессоров находится в разработке (см. проект ELKS(англ.));
  - o Microsoft Xbox (Pentium III).
- \* Intel IA-64
- \* MIPS
  - o Silicon Graphics, Inc.;
  - o Cobalt Qube, Cobalt Raq;
  - o Sony PlayStation 2, PlayStation 3;
  - o DECstation;
  - o некоторые другие.
- \* Motorola 68020 и выше:
  - o более новые Amiga: A1200, A2500, A3000, A4000
  - o Apple Macintosh II, LC, Quadra, Centris и ранняя серия Performa
  - o рабочие станции Sun Microsystems серии 3 (экспериментальная, с использованием Sun-3 MMU)
- \* NEC v850e
- \* Renesas M32R
- \* PowerPC и IBM POWER:
  - o все новые компьютеры Apple (все оснащённые PCI Power Macintoshes, ограниченная поддержка NuBus Power Macs)
  - o клоны PCI Power Mac, разработанные Power Computing, UMAX и Motorola
  - o IBM RS/6000, iSeries- и pSeries-системы
  - o Pegasos I и II системы
  - o некоторые встроенные системы PowerPC
- \* SPARC и UltraSparc: Sun 4-series, SPARCstation/SPARCserver, Ultra-, Blade- и

## **Лицензия**

Linux распространяется на условиях лицензии GNU General Public License, то есть свободно. Эту лицензию выбрал Линус Торвальдс практически сразу после того, как стало понятно, что его хобби начало получать распространение по всему миру. Обладателем торговой марки Linux™ является Линус, а помогает следить за соблюдением его прав и условий GPL Фонд свободного программного обеспечения.

## **Символ**

Официальным символом Linux является пингвин по имени Тух, отличающийся от «обычных» пингвинов жёлтым цветом клюва и лап.

## **BSD**

Ссылка на оригинал: <http://ru.wikipedia.org/wiki/BSD>

С версии: 1.3

**BSD (англ. Berkeley Software Distribution)** — система распространения программного обеспечения в исходных кодах, созданная для обмена опытом между учебными заведениями. Особенностью пакетов ПО BSD была специальная лицензия BSD, которую кратко можно охарактеризовать так: весь исходный код — собственность BSD, все правки — собственность их авторов.

В данный момент термин BSD чаще всего употребляется как синоним BSD-UNIX общего названия вариантов UNIX, восходящих к дистрибутивам университета Беркли.

**К семейству BSD относятся:** NetBSD, FreeBSD, OpenBSD, DragonFly BSD, PC-BSD, DesktopBSD, Darwin (ядро Mac OS X).

Отличия от «классической» системы UNIX состоят в отсутствии переключения уровней выполнения, системе печати, файловой системе и командного процессора. Пользовательские команды практически идентичны.

## **История BSD**

В начале 1978 года аспирант UCB Билл Джой (Bill Joy) начал создание дистрибутива программ Беркли (Berkeley Software Distribution — BSD). Первая лента (360м, ценой около \$50) содержала систему Pascal для UNIX и редактор ex; распространено около 30 копий. До конца 1978 года распространялся пакет 2BSD, продано около 75 копий. Приблизительно тогда же компания Interactive Systems выпускает первую коммерческую версию UNIX, а фирма Whitesmiths — первый клон системы UNIX — Idris.

Версия UNIX 7 была первой переносимой операционной системой. Однако её производительность была хуже, чем в версии шесть. После многих доработок в январе 1982 года вышла в свет модель 2.8.1BSD.

Впоследствии вышли версии:

- \* 4BSD — октябрь 1980

- \* 4.1BSD — июнь 1981

- \* 4.1a, 4.1b и 4.1c (1982—1983)

- \* 4.2 — сентябрь 1983 — действительно сильно переработанная система, включавшая поддержку сетей (протокол TCP/IP, новую файловую систему и возможность использования сигналов.

Последней университетской версией была 4.4BSD(июнь 1993). После суда права на BSD были переданы компании BSDI (Berkeley Software Design, Inc).

На базе BSD-UNIX существуют несколько операционных систем с открытыми исходными текстами:

- \* NetBSD

- \* FreeBSD

- \* OpenBSD

- \* DragonFlyBSD

- \* PC-BSD

- \* DesktopBSD

# GNU General Public License

Ссылка на оригинал: [http://ru.wikipedia.org/wiki/GNU\\_General\\_Public\\_License](http://ru.wikipedia.org/wiki/GNU_General_Public_License)

С версии: 1.3

**GNU General Public License** (Универсальная общедоступная лицензия GNU или Открытое лицензионное соглашение GNU) — возможно, наиболее популярная лицензия на свободное программное обеспечение, созданная в рамках проекта GNU в 1988 г. Её также сокращённо называют GNU GPL или даже просто GPL, если из контекста понятно, что речь идёт именно о данной лицензии (существует довольно много других лицензий, содержащих слова «general public license» в названии). Вторая версия этой лицензии была выпущена в 1991 году, третья версия, после многолетней работы и длительной дискуссии — в 2007 году. GNU Lesser General Public License (LGPL) — это модифицированная версия GPL, предназначенная для некоторых библиотек ПО.

Цель GNU GPL — предоставить пользователю права копировать, модифицировать и распространять программы (что по умолчанию запрещено законом об авторских правах), а также гарантировать, что и пользователи всех производных программ получат вышеперечисленные права. Принцип «наследования» прав называется «копилефт» (транслитерация английского «copyleft») и был придуман Ричардом Столлмэном. По контрасту с GPL, лицензии собственного ПО очень редко дают пользователю такие права и обычно, наоборот, стремятся их ограничить, например, запрещая восстановление исходного кода.

## Свобода

GPL предоставляет получателям компьютерных программ следующие права, или «свободы»:

- свободу запуска программы, с любой целью;
- свободу изучения того, как программа работает, и её модификации (предварительным условием для этого является доступ к исходному коду);
- свободу распространения копий;
- свободу улучшения программы, и выпуска улучшений в публичный доступ (предварительным условием для этого является доступ к исходному коду).

## История

GPL была написана Ричардом Столлмэном для использования с программами как часть проекта GNU. Она базируется на сходных лицензиях, использовавшихся для ранних версий GNU Emacs, GDB (отладчика GNU) и Коллекции компиляторов GNU (GCC), унифицирует и обобщает их.

Лицензии-прототипы содержали части, подобные частям GPL, но были специфичными для каждой программы. Целью Столлмэна являлось создание единой лицензии, которая могла бы использоваться для любого проекта, делая таким образом возможным совместное использование кода различными программами. Такой лицензией и стала первая версия GNU GPL, выпущенная в январе 1989 года.

В 1990 году стало очевидным, что требуется менее ограничивающая лицензия, которая могла бы использоваться для некоторых библиотек ПО; когда версия 2 GPL



была выпущена в июне 1991 года, вместе с ней была введена в обращение GNU Library General Public License, также получившая номер 2, для обозначения того, что эти две лицензии являются взаимодополняющими. Номера версий разошлись в 1999 году, когда была выпущена LGPL версии 2.1, которая была переименована в Lesser General Public License для уточнения её местоположения в философии GNU.

## **GPL v3**

В 2005 году Эбен Моуглен и Ричард Столлмэн написали черновик GPL версии 3. В разгоревшейся затем 7 апреля 2005 года дискуссии в Филадельфии, Столлмэн сделал несколько заявлений, касающихся патентов на ПО и DRM.

В 2006 году Free Software Foundation начал двенадцатимесячную консультацию о возможных изменениях в GPL. Этот процесс координируется Фондом свободного программного обеспечения, Правовым центром свободы программного обеспечения и Европейским Фондом свободного программного обеспечения. Целью консультаций является создание новой версии лицензии с учётом рекомендаций и опыта всех заинтересованных сторон, но с сохранением приверженности принципам свободного ПО.

**Первый черновик был опубликован 16 января 2006 года.**

Тем не менее, 25 января 2006 года Линус Торвальдс публично заявил, что ядро Linux, скорее всего, будет по-прежнему распространяться по лицензии GPL версии 2. (В отличие от многих других GPL-программ, Linux распространяется на условиях только второй версии GPL, а не «версии 2 или более поздней»).

В своём сообщении в почтовую рассылку для Linux-разработчиков Линус Торвальдс, автор ядра Linux, говорит о том, что ОС Solaris может инициировать переход Linux-ядра на новую готовящуюся версию лицензии на свободное программное обеспечение — GNU GPLv3.

«Если Sun действительно собирается выпустить OpenSolaris под GPLv3, это может стать хорошей причиной» для перехода Linux на новую лицензию, заявил Торвальдс.[1]

«Не думаю, что GPLv3 — такая же хорошая лицензия, как и GPLv2, но с другой стороны, я прагматик, и, если мы можем избежать ситуации с существованием двух ядер, распространяющихся под разными лицензиями, и с вызванными этим разногласиями, я по крайней мере вижу причину для перехода на GPLv3», — пояснил свою позицию главный разработчик Linux-ядра.

До этого Торвальдс уже выражал своё недовольство новой версией лицензии GNU GPL, однако после появления последнего чернового варианта GPLv3 стал лучше относиться к этому проекту. Несмотря на это, сам он до сих пор отдаёт предпочтение GPLv2.

Компании, распространяющие GPLv3-ПО, не могут предъявлять судебные претензии к пользователям GPLv3-продуктов.

Окончательная версия GPLv3 была опубликована 29 июня 2007. Черновой вариант перевода можно прочитать тут.

## Схема GNU GPL

Текст GNU GPL состоит из нескольких пронумерованных разделов. Ниже приведена схема версии 2.0 лицензии. Эта схема не имеет никакой юридической силы и служит только для краткого ознакомления.

0. Определения
  - \* (первый абзац) Определение термина «программа»
  - \* (второй абзац) Область действия лицензии
1. Право на копирование и распространение
2. Изменение программы
  - \* (первый абзац) Право на изменения при соблюдении следующих условий:
    - о а) добавление информации об изменении в модифицированных файлах;
    - о б) лицензирование модифицированных версий на условиях GNU GPL;
    - о с) условное требование интерактивного вывода информации об авторских правах и отсутствии гарантии.
  - \* (абзацы 2—4) Уточнение термина «производная работа»
3. Требование предоставления исходного кода
  - \* (первый абзац) Возможные варианты распространения исполнимого кода:
    - о а) распространение вместе с исходным кодом, или
    - о б) распространение с гарантией предоставления исходного кода, или
    - о с) (для некоммерческого использования) распространение вместе с такой гарантией, полученной от третьего лица.
  - \* (второй абзац) Определение термина «исходный код»
  - \* (третий абзац) Достаточность одинакового доступа для копирования исполнимого и исходного кодов
4. Прекращение действия лицензии при нарушении её условий
5. Акты, означающие принятие лицензии
6. Запрещение дополнительных ограничений при дальнейшем распространении
7. Внешние ограничения не снимают обязательства выполнять условия лицензии
8. Возможность географических ограничений
9. Будущие версии GNU GPL
10. Запросы на исключения из правил
11. Отказ от предоставления гарантий
12. Отказ от ответственности

# Linux не для идиотов: сборник рассказов и рецептов

Ссылка на оригинал: <http://myfotomx.com/dalth/linuxbook.odt>

Автор: Артем Капитула (no-dashi, dalth & viking).

Скачано с <http://myfotomx.com/dalth>: 15.07.07

С версии: 1.1

## Предисловие

Я много лет работал с Linux, и, общаясь со многими единомышленниками, сделал один странный вывод: нам катастрофически не хватает документации. Причем не инструкций вида «сделайте так и вот так» и не справочных руководств, а некоторого «мостика» между новичком, который видел только графическую оболочку подобную GNOME или KDE, и профессионалом, который может скомпилировать необходимый ему драйвер, даже если этот драйвер упорно сопротивляется.

Соответственно, я попытался сделать попытку написать книжку (хотя на книгу этот материал не тянет, скорее на методичку), которая позволила бы сравнительно просто перейти с пользовательского уровня знакомства с Linux на более высокий уровень, не проходя по типичным ошибкам, и за сравнительно короткое время.

Некоторое время я использовал фрагменты этой книги также как часть учебного курса в Челябинском Государственном университете (ЧелГУ) для того, чтобы мои студенты могли ориентироваться в системе несколько лучше, чем на уровне команд ls/ps/exit. Конечно, если честно – как учебное пособие эта маленькая книга непригодна, но, как мне кажется, она неплохо подходит как дополнительная литература.

Если у вас есть пожелания и дополнения – пишите мне почтой на [dalt74@gmail.com](mailto:dalt74@gmail.com), я постараюсь учесть ваши замечания в следующей редакции. Большое спасибо всем тем кто участвовал в рецензировании и помогал советами и замечаниями.

Искренне ваш, Артем Капитула (no-dashi, dalth & viking)

P.S. если вы будете распечатывать или выкладывать это пособие – пожалуйста, указывайте ссылку на автора и оригинальный источник, договорились?

P.P.S. пока что есть следующий to-do list: основы DNS, базовые настройки серверов и рабочих станций под типичные нужды. Ориентировочный срок следующей редакции – через три месяца.

## Ядро и модули

Ядро Linux является единственным процессом, имеющим непосредственный доступ к аппаратуре – все остальные процессы обращаются к устройствам только через ядро. В ядре Linux можно выделить несколько важных подсистем: подсистему управления памятью; планировщик задач; подсистему VFS – виртуальную файловую

систему и драйверы.

Подсистема управления памятью управляет распределением оперативной памяти между задачами, а также обслуживает файл подкачки, планировщик задач управляет разделением процессорного времени между задачами (процессами и нитями), подсистема VFS предназначена для обслуживания файловых операций.

Драйверы предназначены для управления устройствами и поддержки различных протоколов. Существует две разновидности драйверов – статически подключенные в ядро драйверы и загружаемые модули; первые всегда загружены, вторые могут быть загружены при необходимости и выгружены, когда необходимость в них отпала. Каждый модуль и само ядро содержат сигнатуру версии – специальную метку, которая описывает версию ядра и некоторые опции, использованные при компиляции ядра. Кроме того, ядра версии 2.6 могут поддерживать цифровую подпись модулей. Это было сделано для повышения надежности системы – по умолчанию ядро не будет загружать и использовать драйверы, предназначенные для другой версии ядра, или собранные с другими опциями, поскольку это может привести к возникновению проблем. Версию ядра можно узнать с помощью команды `uname`:

```
[dalth@inferno dalth]$ uname -r  
2.6.8.1
```

В принципе, утилиты для работы с модулями поддерживают возможность загрузки модулей, собранных для другого ядра – но пользоваться этой возможностью следует с крайней осторожностью, поскольку это может привести к непредсказуемым последствиями – от ошибок ядра (`kernel panic`) и вплоть до странных потерь данных и непонятных ошибок, взявшихся на пустом месте.

В большинстве дистрибутивов образ ядра располагается в каталоге `/boot`, а загружаемые модули ядра располагаются в `/lib/modules/<версия_ядра>`, там же располагается таблица зависимостей модулей, поскольку некоторые модули могут нуждаться для своей работы в других модулях (например, драйвер поддержки SCSI-дисков нуждается в драйвере поддержки SCSI – как следствие этого, если объекты какого-либо модуля используются другим драйвером, такой модуль невозможно выгрузить). Следующий листинг демонстрирует достаточно типичное содержание каталога модулей для ядер линейки 2.6:

```

[root@vikings dev]# ls -l /lib/modules/2.6.8.1/
total 616
lrwxrwxrwx    1 root root      18 Apr 27 15:36 build -> /usr/src/linux
drwxr-xr-x   10 root root    4096 Oct  1 13:55 kernel
-rw-r--r--    1 root root 108680 Oct  1 13:56 modules.alias
-rw-r--r--    1 root root   69 Oct  1 13:56 modules.ccwmap
-rw-r--r--    1 root root 153967 Oct  1 13:56 modules.dep
-rw-r--r--    1 root root    73 Oct  1 13:56 modules.ieee1394map
-rw-r--r--    1 root root   357 Oct  1 13:56 modules.inputmap
-rw-r--r--    1 root root  16658 Oct  1 13:56 modules.isapnpmap
-rw-r--r--    1 root root  85093 Oct  1 13:56 modules.pcimap
-rw-r--r--    1 root root  68078 Oct  1 13:56 modules.symbols
-rw-r--r--    1 root root 150781 Oct  1 13:56 modules.usbmap
lrwxrwxrwx    1 root root      18 Oct  1 13:22 source -> /usr/src/linux
[root@vikings dev]# find /lib/modules/2.6.8.1/kernel -type f | head
-20
/lib/modules/2.6.8.1/kernel/arch/i386/kernel/cpuid.ko
/lib/modules/2.6.8.1/kernel/arch/i386/kernel/microcode.ko
/lib/modules/2.6.8.1/kernel/arch/i386/kernel/msr.ko
/lib/modules/2.6.8.1/kernel/crypto/blowfish.ko
/lib/modules/2.6.8.1/kernel/crypto/deflate.ko
/lib/modules/2.6.8.1/kernel/crypto/md5.ko
/lib/modules/2.6.8.1/kernel/crypto/twofish.ko
/lib/modules/2.6.8.1/kernel/drivers/acpi/fan.ko
/lib/modules/2.6.8.1/kernel/drivers/acpi/processor.ko
/lib/modules/2.6.8.1/kernel/drivers/acpi/thermal.ko
/lib/modules/2.6.8.1/kernel/drivers/base/firmware_class.ko
/lib/modules/2.6.8.1/kernel/drivers/block/cryptoloop.ko
/lib/modules/2.6.8.1/kernel/drivers/block/loop.ko
/lib/modules/2.6.8.1/kernel/drivers/block/nbd.ko
/lib/modules/2.6.8.1/kernel/drivers/block/paride/epat.ko
/lib/modules/2.6.8.1/kernel/drivers/block/paride/paride.ko
/lib/modules/2.6.8.1/kernel/drivers/block/paride/pd.ko
/lib/modules/2.6.8.1/kernel/drivers/block/paride/pg.ko
/lib/modules/2.6.8.1/kernel/drivers/bluetooth/bcm203x.ko
/lib/modules/2.6.8.1/kernel/drivers/bluetooth/bfusb.ko
[root@vikings dev]#

```

Бинарные файлы модулей содержатся в подкаталоге *kernel*, и имеют расширение “.o” для ядер линейки 2.4 и расширение “.ko” для ядер линейки 2.6. В файлах *modules.\*.map* перечисляются символы (функции и переменные), экспортируемые модулями.

Часто в одном каталоге с модулями содержатся ссылки на каталоги, в которых хранились исходные тексты ядра и в котором производилась сборка ядра (это ссылки *source* и *build*, соответственно). Эти ссылки, как правило, используются для того, чтобы скомпилировать модули или программы, которые зависят от версии ядра (например, эти ссылки используются при инсталляции модуля поддержки видеокарт *nvidia*).

Учет взаимосвязей между загруженными модулями производится с помощью счетчика ссылок – модуль увеличивает свой счетчик ссылок как только какой-либо его объект начинает использоваться другими драйверами. Когда объекты модуля освобождаются, счетчик ссылок уменьшается. Модуль может быть выгружен, если число ссылок на него станет равно 0. В ядрах версии 2.6 существует возможность произвести принудительную выгрузку модуля даже если он используется, но этим пользоваться без крайней необходимости не рекомендуется, поскольку очень возможно возникновение ошибок.

Ядро содержит множество переменных и функций, которые используются различными драйверами, и соответственно, если какой-либо драйвер должен обратиться к такому объекту, он должен знать его адрес. Некоторые драйверы также содержат переменные и функции, которые должны быть доступны другим

драйверам, и адреса таких объектов тоже размещаются в специальной таблице. При загрузке модуля ядро и программа загрузки модулей устанавливает адреса всех объектов, в которых нуждается загружаемый модуль, и только после этого модуль может начать инициализацию.

Загрузка модулей и их выгрузка осуществляются утилитами *modprobe*, *insmod* и *rmmod*. Программы *modinfo* и *depmod* предназначены для получения служебной информации о загружаемых модулях. В процессе своей работы эти утилиты опираются на конфигурационные файлы */etc/modprobe.conf* (для ядер 2.6.X) или *modules.conf* (для ядер 2.4.X).

## **Загрузка операционной системы**

Для компьютеров архитектуры x86 последовательность загрузки хорошо описана в специализированной литературе, но мы все-таки кратко ее повторим. После включения компьютера первым загружается BIOS. Он тестирует аппаратуру и инициализирует устройства. После этого BIOS прочитывает начальный сектор загрузочного жесткого диска (MBR), убеждается что он содержит код первичного загрузчика, и передает управление прочитанному коду. Кроме кода первичного загрузчика, начальный сектор также может содержать таблицу разделов жесткого диска.

В задачи первичного загрузчика входит чтение основного кода загрузчика операционной системы и передача управления ему, после чего основная часть загрузчика может считать конфигурационный файл, загрузить ядро операционной системы, установить параметры для ядра и передать ядру управление. Ядро инициализирует драйверы, проверяет параметры и, опираясь на параметры, пытается смонтировать корневую файловую систему, после чего (если не было проинструктировано об ином) запускает программу */sbin/init*. Дальнейшая работа *init* подробно описана во множестве книг и статей.

В настоящий момент в мире Linux наиболее распространен загрузчик GRUB. Этот загрузчик состоит из нескольких частей – первичного загрузчика, собственно основного кода который организует интерфейс пользователя, и набора мини-драйверов различных файловых систем, позволяющих прочесть необходимые файлы с файловой системы в момент когда операционная система еще недоступна. Каждая из этих компонент работает на одним из двух этапов загрузки. Рассмотрим эти этапы:

**Этап 0** – здесь срабатывает первичный загрузчик GRUB. Он компактен и умещается в один блок жесткого диска, что позволяет при желании разместить его в MBR. В задачи кода *stage\_0* входит прочтение кода необходимого на следующем этапе (собственно кода загрузчика и мини-драйвера файловой системы где расположены основные файлы загрузчика), и передача управления прочитанному коду.

**Этап 1** – это на этом этапе первичным загрузчиком в память уже загружен основной код загрузчика, а также мини-драйвер файловой системы, на которой расположены конфигурационные файлы загрузчика, ядро и необходимые драйверы. Основной код, используя функции мини-драйвера, прочитывает конфигурационный файл и организует диалог с пользователем. В зависимости от выбора пользователя, используя мини-драйвер файловой системы, с диска прочитываются файлы ядра и необходимых драйверов, после чего управление передается ядру. Как вариант, пользователь может отказаться от загрузки Linux и инструктировать GRUB прочесть загрузочный сектор некоторого раздела жесткого диска и передать управление ему.

Первичный загрузчик из состава GRUB может быть расположен как в MBR, так и в загрузочном секторе какого-либо раздела жесткого диска – или даже храниться в файле и быть вызван из другого загрузчика (например NTLOADER).

Нередко случаются ситуации, когда корневая файловая система располагается на устройстве, чей драйвер скомпилирован в виде модуля, или драйвер корневой файловой системы скомпилирован в виде модуля. Получается замкнутый круг – чтобы смонтировать корневую файловую систему, необходимо прочесть драйвер, а чтобы прочесть драйвер – нужно смонтировать корневую файловую систему. Чтобы разорвать этот порочный круг, в Linux была введена поддержка *initrd* – INITial RamDisk.

Initial ramdisk – это файл, который прочитывается загрузчиком ОС и загружается в память вместе с ядром. Ядро интерпретирует фрагмент памяти, куда загружен этот файл, как блочное устройство с помощью специального драйвера, статически вкомпилированного в ядро. После инициализации статически скомпилированных драйверов ядро монтирует файловую систему, хранящуюся в `initrd` и загружает с нее драйверы и запускает программы, необходимые для монтирования корневой файловой системы.

Обычно файл с образом ядра хранится в каталоге `/boot` и называется `vmlinux-<версия>`, там же располагается файл `initrd-<версия>.img`, содержащий образ файловой системы `initrd`. Для каждой версии ядра необходим свой образ `initrd`, в который включены модули для этой версии ядра. В большинстве случаев образ `initrd` поставляется в бинарном пакете вместе с ядром, или автоматически создается в процессе построения ядра из исходных текстов в момент выполнения команды `make install`, если же возникает ситуация, когда необходимо повторно собрать образ `initrd` (например, в сервере сменили SCSI-контроллер), можно воспользоваться специальной командой `mkinitrd`, позволяющей произвести повторную генерацию `initrd`:

```
[root@viking dalth]# mkinitrd /tmp/initrd-2.4.8.1.img 2.6.8.1
[root@viking dalth]# cp /tmp/initrd-2.4.8.1.img /boot
[root@viking dalth]# reboot
```

Для Linux существует два основных загрузчика – LILO и GRUB. Второй является более поздней разработкой и немного удобней в использовании, а LILO используется по историческим или личным причинам (например, он нравится системному администратору), либо в некоторых случаях, когда требуются специфичные для LILO функции. Для более подробной справки лучше обратиться к справочному руководству (`man grub`, `man lilo`).

Из интересных особенностей GRUB и LILO следует отметить то, что и оба этих загрузчика, и ядро оперируют термином корневой файловой системы – но если с точки зрения ядра эта та файловая система, которая содержит программу `/sbin/init`, то с точки зрения обоих загрузчиков корневой файловой системой является та, которая содержит образ ядра и файл `initrd`.

## Организация памяти

Подсистема виртуальной памяти управляет распределением оперативной памяти между задачами (процессами). Каждая задача считает, что ей выделен непрерывный участок памяти максимального размера, поддерживаемого на соответствующей архитектуре (для архитектуры x86 это 4GB). Из них последний гигабайт резервируется для себя ядром, часть отдается под код программы и разделяемые библиотеки (оба этих фрагмента ядром защищаются), а оставшееся пространство отдается собственно программе под ее данные – но это только то, как видит это все программа.

На самом же деле программа занимает только тот объем памяти, с которым она реально работает. Большинство памяти существует только “на бумаге”, т.е. будет предоставлена программе в тот момент, когда она обратится в эту область. Ядро распределяет память страницами фиксированного размера. Процедура, когда страница оперативной памяти объявляется частью адресного пространства процесса, называется *отображением* этой страницы в адресное пространство процесса.

Соответственно, ядро отображает реально используемые страницы в виртуальное адресное пространство процесса. Когда процесс обращается к некоторой странице своего адресного пространства, ядро проверяет, имеет ли он право на доступ к этой странице, и если проверка пройдена и доступ получен, то ядро переадресовывает обращение на реальный адрес этой страницы. Если это первое обращение к

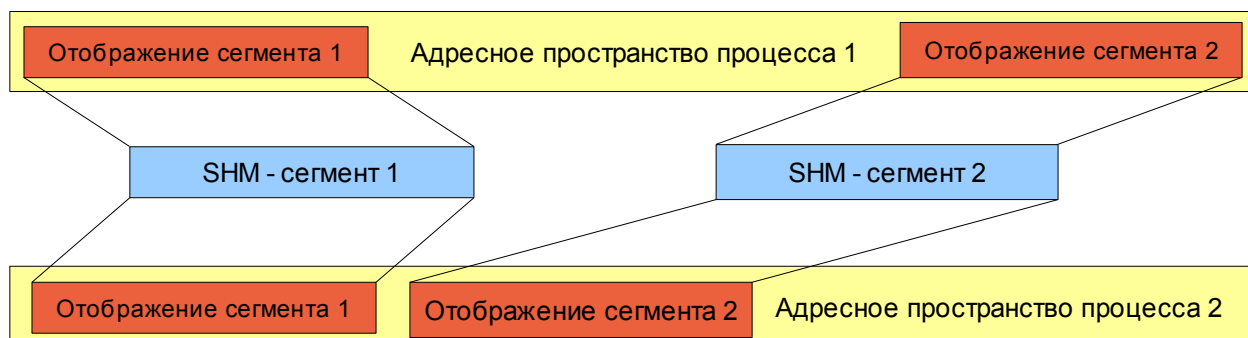
странице, ядро попытается найти свободную страницу и в случае успеха отобразит ее в адресное пространство соответствующего процесса. Размер страницы фиксирован архитектурой процессора, и для x86 ее размер составляет 4096 байт. Если случается ситуация, когда свободных страниц больше нет, но существует файл подкачки, ядро может убрать одну из наиболее долго не использовавшихся страниц в файл подкачки, и освободившуюся физическую страницу отдать запросившему память процессу. Если же нет ни незанятого пространства в файле подкачки, ни свободных страниц RAM, то развитие событий может быть следующим: либо запросивший память процесс прерван и “убит” системой, либо какой-то другой из процессов (это определяется специфическими алгоритмами) будет “убит” ядром, и освободившаяся память будет передана запросившему память процессу.

На самом деле большинством действий занимается одна из подсистем процессора, называемая MMU – Memory Management Unit, и в действительности ядро просто полагается на его работу и вмешивается в нее только для проведения операций пейджинга (подгрузки/выгрузки страниц в SWAP-файл), или когда возникает ошибка доступа к странице.

Ограничение адресного пространства в 4GB не означает, что система не сможет адресовать более этого объема памяти. На платформе x86 ядро Linux может использовать до 64GB, а ограничение в 4GB накладывается лишь на размер адресного пространства процесса.

## System V shared memory

Linux поддерживает стандартную для всех UNIX-подобных операционных систем организацию разделяемой памяти. Пользовательские приложения могут создавать *сегменты* разделяемой памяти, которые могут быть присоединены к некоторому фрагменту адресного пространства процесса. Любой процесс, имеющий достаточные права доступа, может присоединиться к сегменту разделяемой памяти, и отобразить его в свое адресное пространство, начиная с некоторого адреса.



Если в приведенной схеме любой из процессов изменит содержимое памяти в области, занимаемой отображением одного из сегментов, то же самое изменение произойдет в адресном пространстве другого процесса, поскольку соответствующий сегмент существует в одном экземпляре и отображен в адресное пространство обоих процессов.

Кроме System V IPC ядро Linux также поддерживает другие объекты IPC, в частности семафоры и очереди сообщений. Каждый объект System V IPC идентифицируется уникальным ключом. Просмотреть список всех объектов IPC можно командой `ipcs`. Команда `ipcrm` позволяет удалять объекты IPC, которые по каким-либо причинам остались не освобожденными после завершения создавшего их процесса – например, такая ситуация может возникнуть после аварийного завершения работы СУБД Oracle, Informix или DB2.

Соответственно, перед перезапуском процесса системный администратор с помощью команды `ipcrm` должен освободить неиспользуемые объекты IPC, поскольку стартующее приложение не сможет их повторно создать и не будет корректно работать.



Для каждого объекта IPC система устанавливает права доступа, как если бы это был файл (т.е. для каждого объекта IPC можно устанавливать набор прав ugo/rwx, но в отличие от обычных файлов сменить права доступа для IPC-объектов можно только вызывая специализированные функции, предназначенные для работы с такими объектами.

```
[dalth@viking dalth]$ ipcs

----- Shared Memory Segments -----
key          shmid          owner          perms          bytes          nattch
status
0x00000000  0              oracle         640            4194304        10
0x00000000  32769          oracle         640            20971520       10
0x00000000  65538          oracle         640            29360128       10
0x0d3c24a0  98307          oracle         640            29360128       50
0x00000000  13697028       root           777            49152          1
0x00000000  13729797       root           777            16384          1
0x000004d2  13795334       dalth          666            1008           2
0x00000000  14286866       root           644            790528         2          dest
0x00000000  21823507       dalth          600            393216         2          dest
0x00000000  21921814       root           644            122880         2          dest
0x00000000  14516249       root           644            151552         1          dest

----- Semaphore Arrays -----
key          semid          owner          perms          nsems
0x0b4f657c  262147         oracle         640            154
0x000004d2  458756         dalth          666            1

----- Message Queues -----
key          msqid          owner          perms          used-bytes      messages

[dalth@viking dalth]$
```

Поддержка System V IPC позволяет сравнительно легко переносить на Linux приложения, написанные для других UNIX-систем.

## Файловая система

VFS и драйверы файловых систем являются одной из важнейших составляющих ядра. Для того, чтобы получить доступ к файлам, хранящимся на каком-либо устройстве хранения данных, необходимо, чтобы в ядро был загружен драйвер соответствующей файловой системы, и файловая система была смонтирована. Драйвера всех файловых систем поддерживают набор стандартных функций: открыть файл по имени, записать данные в файл, прочитать данные из файла, закрыть файл, удалить файл и т.д.

Уточним, что драйвера файловых систем не занимаются кэшированием, этим занимается VFS.

При первоначальной загрузке драйвер файловой системы регистрирует в VFS имя файловой системы и те функции, которые предназначены для выполнения стандартных файловых операций. Впоследствии при обращении к файлу на какой-либо файловой системе VFS будет переадресовывать обращение на соответствующую функцию, если таковая была зарегистрирована драйвером. Посмотреть список обслуживаемых ядром файловых систем можно в файле `/proc/filesystems`:

```
[dalth@viking proc]$ cat /proc/filesystems
nodev    sysfs
nodev    rootfs
nodev    bdev
nodev    proc
nodev    sockfs
nodev    usbfs
nodev    usbdevfs
nodev    futexfs
nodev    tmpfs
nodev    pipefs
nodev    eventpollfs
nodev    devpts
nodev    ext2
nodev    ramfs
nodev    hugetlbfs
nodev    iso9660
nodev    devfs
nodev    mqueue
nodev    ext3
nodev    rpc_pipefs
nodev    nfsd
nodev    smbfs
```

Операция монтирования предназначена для того, чтобы сделать доступной файловую систему, расположенную на каком-либо блочном устройстве. Суть операции монтирования заключается в том, что ядро ассоциирует некоторый каталог (называемый точкой монтирования) с блочным устройством и драйвером файловой системы. Для этого оно передает ссылку на блочное устройство драйверу файловой системы, и в случае, если драйвер успешно проидентифицировал эту файловую систему, ядро заносит в специальную таблицу монтирования информацию о том, что все файлы и каталоги, чей полный путь начинается с указанной точки монтирования, обслуживаются соответствующим драйвером файловой системы и расположены на указанном блочном устройстве.

Некоторые файловые системы не нуждаются в блочном устройстве, поскольку хранят свои данные исключительно в памяти, например файловая система *procfs*, через файлы которой можно получить доступ к различным системным параметрам и таблицам.

Очень часто при монтировании файловой системы системный администратор имеет возможность задать *опции монтирования*. Опции монтирования – это специальные параметры, которые влияют на работу драйвера файловой системы, когда он работает с файловой системой на соответствующем блочном устройстве – например, с помощью опций монтирования можно управлять режимом кэширования данных, преобразованиями имен файлов и данных, включать и отключать поддержку ACL и т.д.

Посмотреть таблицу примонтированных файловых систем можно через файл */proc/mounts*:

```
[dalth@viking proc]$ cat /proc/mounts
rootfs / rootfs rw 0 0
/dev/root / ext3 rw 0 0
none /dev devfs rw 0 0
/proc /proc proc rw,nodiratime 0 0
/sys /sys sysfs rw 0 0
none /dev/pts devpts rw 0 0
usbdevfs /proc/bus/usb usbdevfs rw 0 0
/dev/chimera/var /var ext3 rw 0 0
/dev/chimera/temp /tmp ext3 rw 0 0
/dev/chimera/usr /usr ext3 rw 0 0
/dev/chimera/home /home ext3 rw 0 0
/dev/chimera/opt /opt ext3 rw 0 0
none /dev/shm tmpfs rw 0 0
```

Виртуальная файловая система Linux различает несколько типов файлов: каталоги, обычные файлы, именованные каналы, символичные ссылки, сокеты и специальные файлы. Каждая из этих разновидностей обрабатывается своим собственным образом:

- Обычные файлы могут быть прочитаны, записаны, удалены или отображены в память.
- Каталог можно представить как список имен файлов, и для этого списка определены операции добавления элемента в список, удаление элемента из списка, переименование элемента списка.
- Именованные каналы являются просто буферами – в него можно записать некоторый объем данных, и прочесть их в том же порядке, в каком они были записаны.
- Сокеты являются вариантом именованных каналов, но если у именованного канала буфер только один, то есть нельзя определить какой процесс записал данные в канал, то у сокетов может быть несколько клиентов, один из которых осуществляет управление сокетом и может обмениваться данными с любым из остальных клиентов, а те в свою очередь могут обмениваться данными с диспетчером канала – то есть сокет поддерживает множество независимых буферов, по одному на каждую пару сервер+клиент
- Специальные файлы предназначены для того, чтобы осуществлять прямой доступ к устройствам. Подробнее о специальных файлах будет говориться в главе **Секреты /dev**
- Символичные ссылки являются “ярлыками”, которые могут содержать имя другого файла – и тогда операции чтения, записи и открытия/закрытия файла автоматически переадресовываются к файлу, на который указывает символическая ссылка, но в отличие от “ярлыков” Windows (shortcuts), символичные ссылки (symbolic links) не требуют специальных функций при работе – все программы (кроме тех, которые специально предназначены для работы с ними) видят их как обычные файлы, и также их открывают, читают и записывают данные и т.д.

В некоторых файловых системах, которые изначально проектировались для UNIX-подобных систем, есть возможность создавать кроме символических ссылок еще и жесткие ссылки. Фактически, жесткая ссылка – это второе имя для файла. Жесткие ссылки возможно создавать только в пределах одной файловой системы.

Из-за того, что в VFS присутствует понятие кэширования, перед отключением системы необходимо делать обязательный сброс изменений на диск. Сброс кэша на диск осуществляется в момент размонтирования файловой системы. Кроме того, с помощью команды `sync` можно в любой момент принудительно сбросить на диск все закэшированные изменения в файловой системе (например, системные администраторы часто делают `sync` перед загрузкой нового драйвера).

Размонтировать файловую систему можно только тогда, когда ни один процесс не удерживает в открытом состоянии файлов с этой файловой системы, а также не находится ни один процесс не имеет рабочим каталогом каталога с размонтируемой файловой системы. При невыполнении этого условия размонтировать файловую систему не удастся:

```
[root@viking dalth]# umount /home/ftp/pub/linux/fedora/cd1
umount: /home/ftp/pub/linux/fedora/cd1: device is busy
umount: /home/ftp/pub/linux/fedora/cd1: device is busy
[root@viking dalth]#
```

Некоторые файловые системы поддерживают специальные опции, позволяющие принудительно синхронизировать файловую систему при каждой операции чтения или записи. Обычно опции, влияющие на синхронизацию файловой системы, содержат в своем названии слово **sync**, например приведенная ниже команда инструктирует операционную систему примонтировать некоторый раздел в режиме принудительной синхронизации:

```
[root@viking dalth]# mount -t ext3 -o sync,dirsync /dev/hda9 /home
```

Следует учесть, что принудительная синхронизация – это удар по производительности операций записи для файловой системы, смонтированной в таком режиме, поэтому использовать такой его следует осторожно.

## Права доступа

Кроме стандартных наборов прав доступа к файлам некоторые файловые системы Linux поддерживают т.н. **POSIX ACL** – списки контроля доступа POSIX. Эта возможность позволяет гибко управлять доступом к файлу, не ограничиваясь “классическим” набором `ugo/rwx`. Для того, чтобы использовать на файловой системе POSIX ACL, необходимо смонтировать файловую систему с опцией `acl`:

```
[root@inferno root]# mount -t ext3 -o acl /dev/inferno/opt /opt
```

Возможно также настроить соответствующий параметр для файловой системы, чтобы она поддерживала POSIX ACL по умолчанию:

```
[root@inferno root]# tune2fs -o acl /dev/inferno/opt
```

После установки соответствующей опции можно приступать к работе с POSIX ACL. Для работы с ними существует две базовых утилиты: `getfacl` для получения списка дополнительных атрибутов доступа, и `setfacl` для установки расширенных атрибутов контроля доступа. Если в выводе команды `ls` вы видите символ “+” рядом со списком стандартных прав доступа, это означает, что для файла также установлены расширенные атрибуты контроля доступа:

```
[root@inferno root]# ls -l /home/dalth/.bash_???????
-rw-r-----+ 1 dalth dalth 20034 Окт 11
22:48 /home/dalth/.bash_history
-rw-r--r-- 1 dalth dalth 191 Авг 23
21:51 /home/dalth/.bash_profile
[root@inferno root]#
```

Для просмотра значений расширенных атрибутов можно воспользоваться утилитой `getfacl`. Жирным шрифтом выделен дополнительный атрибут контроля доступа, позволяющий пользователю `kiki` получить доступ на чтение к файлу `.bash_history`:

```
[root@inferno dalth]# getfacl .bash_history
# file: .bash_history
# owner: dalth
# group: dalth
user::rw-
user:kiki:r--
group:---
mask::r--
other:---
```

Добавим пользователю oracle права на чтение и запись файла .bash\_history с помощью команды setfacl, и затем отберем дополнительные права на доступ к указанному файлу у пользователя kiki:

```
[root@inferno dalth]# setfacl -m u:oracle:rw .bash_history
[root@inferno dalth]# setfacl -x u:kiki .bash_history
[root@inferno dalth]# getfacl .bash_history
# file: .bash_history
# owner: dalth
# group: dalth
user::rw-
user:oracle:rw-
group:---
mask::rw-
other:---
```

Последним шагом сбросим все расширенные атрибуты с файла с файла .bash\_history:

```
[root@inferno dalth]# setfacl -b .bash_history
[root@inferno dalth]# ls -l .bash_history
-rw----- 1 dalth dalth 20034 Окт 11 22:48 .bash_history
```

Расширенные атрибуты позволяют гибко контролировать доступ к файловым объектам, обходя стратегию ugo/rwx пришедшую из “классического UNIX”. Права доступ на файловые объекты могут быть выданы не только пользователю, но и группе.

К сожалению, далеко не все утилиты и файловые системы поддерживают ACL, поэтому при резервном копировании или восстановлении файлов необходимо проверять корректность установки расширенных атрибутов и правильность их переноса.

## **Журналируемые файловые системы**

Для обеспечения сохранности данных и обеспечения целостности файловых систем при неожиданных сбоях были разработаны *журналируемые файловые системы*. Как правило, у этих файловых систем существует специальная область данных, называемая *журналом*. Все изменения, которые необходимо произвести с файловой системой, сначала записываются в журнал, и уже из журнала попадают в основную часть файловой системы.

В большинстве случаев журналируются только *метаданные* файловых систем (служебная информация, обеспечивающая целостность структуры файловой системы – например, изменения в каталогах или служебных таблицах размещения файлов). Некоторые файловые системы позволяют журналировать не только метаданные, но и данные файлов – такой шаг позволяет повысить надежность, но уменьшает скорость записи данных на файловую систему, поскольку каждый блок данных записывается на диск дважды – сначала в журнал, и затем из журнала в основную область файловой системы..

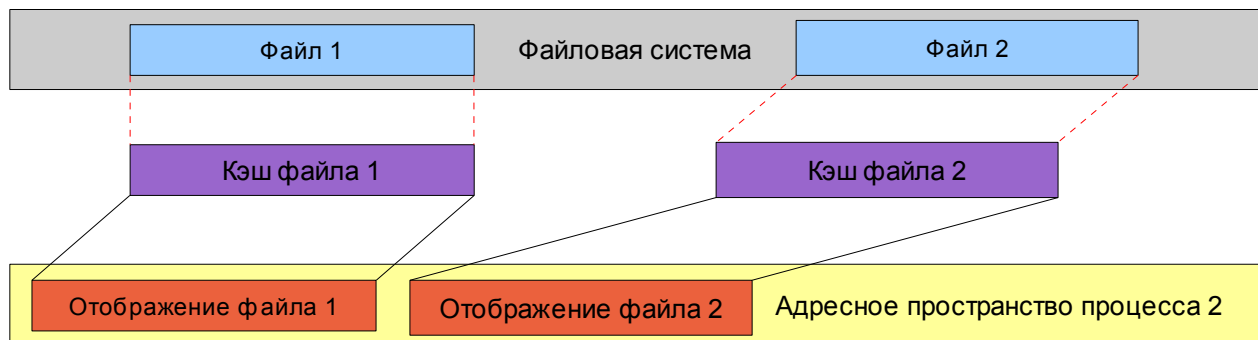
В большинстве случаев, журналируемые файловые системы способны решить проблемы с надежностью при неожиданных сбоях без тех потерь производительности, к которым может привести использование опции **sync** при монтировании.

В частности, к журналируемым файловым системам, например, относятся EXT3, ReiserFS, XFS, JFS и некоторые другие.

## **Отображенные в память файлы**

Объединение кэширования файлов и разделяемой памяти позволяет реализовать такое действие, как отображение файла в память. Для простоты можно представить, что файл загружается в кэш, и страницы кэша отображаются в адресное

пространство процесса, и в результате любое изменение в том фрагменте адресного пространства, которое занято отображением файла, автоматически попадает в закэшированные данные файла. Когда файл закрывается, закэшированные изменения сбрасываются на диск, изменяя сам файл. Кроме того, в свободное время ядро также постепенно сбрасывает изменившиеся кэшированные данные на диск.



На самом деле, механизм отображения файлов в память куда “хитрее” - при обращении по записи к странице, которая является отображением некоторого файла, ядро перехватывает обращение, производит запись в файл (в подавляющем большинстве случаев эта операция попадает в кэш). При обращении по чтению к такой странице ядро опять же перехватывает обращение и производит чтение из файла – в большинстве случаев это чтение производится из кэша. Для наших же целей проще будет считать, что страницы кэша отображены в память процесса.

Такая методика часто используется для загрузки разделяемых библиотек, когда выполняемый код библиотек и исполняемого кода программы хранится в кэше и через отображение файла в память становится виден в адресном пространстве процесса:

```
[dalth@viking dalth]$ cat /proc/1/maps
08048000-08050000 r-xp 00000000 03:01 75813      /sbin/init
08050000-08051000 rw-p 00008000 03:01 75813      /sbin/init
08051000-08072000 rw-p 08051000 00:00 0
40015000-40016000 rw-p 40015000 00:00 0
4c8ee000-4c903000 r-xp 00000000 03:01 92869      /lib/ld-2.3.3.so
4c903000-4c904000 r--p 00014000 03:01 92869      /lib/ld-2.3.3.so
4c904000-4c905000 rw-p 00015000 03:01 92869      /lib/ld-2.3.3.so
4c907000-4calc000      r-xp                00000000      03:01
92857      /lib/tls/libc-2.3.3.so
4calc000-4cale000      r--p                00115000      03:01
92857      /lib/tls/libc-2.3.3.so
4cale000-4ca20000      rw-p                00117000      03:01
92857      /lib/tls/libc-2.3.3.so
4ca20000-4ca22000 rw-p 4ca20000 00:00 0
4d201000-4d20f000 r-xp 00000000 03:01 92965      /lib/libselinux.so.1
4d20f000-4d211000 rw-p 0000d000 03:01 92965      /lib/libselinux.so.1
bffffd000-c00000000 rw-p bffffd000 00:00 0
fffffe000-ffffff000 ---p 00000000 00:00 0
```

На самом деле, драйверы любого устройства и любой файловой системы могут по-своему реализовывать операцию mmap, но большинство драйверов файловых систем полагаются в этом на VFS.

## Специальные файловые системы

Некоторые типы файловых систем являются специальными и предназначены для выполнения или реализации специфических задач операционной системы. К таким файловым системам относятся файловые системы *procfs* и *sysfs*, предоставляющие доступ к различным параметрам системы и системным объектам, “живущим” в ядре.

Файловая система `sysfs` в основном предоставляет доступ к объектам ядра и отображает взаимосвязи между ними. Файловая система `procfs` предоставляет доступ к различным параметрам ядра и драйверов и к пользовательским процессам, позволяя тем самым реализовать такие команды как `ps` или `sysctl`. Большинство файлов в `sysfs` двоичные, в `procfs` – текстовые.

Драйверы файловых системы `ramfs`, `tmpfs` и `shmfs` очень похожи, и после монтирования файловой системы такого типа на ней можно создавать файлы, хранящиеся в памяти и отличаются в основном небольшими особенностями работы (например, страницы, используемые `ramfs` под данные файлов, не вытесняются в `swarp`-файл в отличие от `shmfs` и `tmpfs`). В ядре 2.6 `shmfs` была заменена на `tmpfs`.

## Сетевые файловые системы

Сетевые файловые системы предназначены для получения доступа к файловым системам других компьютеров с использованием сетевых протоколов.

Наиболее часто используются сетевые файловые системы NCPFS (для доступа к серверам Novell NetWare), SMBFS (для доступа к серверам Windows) и NFS (для доступа к файловым системам других UNIX-систем).

Как правило, процедура монтирования сетевых файловых систем схожа с процедурой монтирования обычных файловых систем на блочных устройствах с тем отличием, что вместо блочного устройства указывается адрес сервера, чья файловая система монтируется, и имя монтируемого ресурса. Для примера рассмотрим процедуры монтирования ресурсов, доступных по SMB и по NFS:

```
# mount -t smbfs -o username=usr,workgroup=tst //server/share_name /mnt/smb_target
Password: *****
# mount -t nfs -o timeout=4 server:/export/home /mnt/nfs_target
```

В данном примере опция `-t` команды `mount` указывает тип файловой системы, опция `-o` позволяет задать дополнительные параметры для монтирования – для SMB мы задаем, например, имя пользователя, с правами которого производится подключение к серверу и имя рабочей группы или домена, для NFS мы указываем таймаут, по истечении которого операция ввода/вывода считается неудавшейся. Вместо блочного устройства мы указываем адрес сервера, ресурс которого хотим использовать, и имя ресурса на сервере. Последним параметром идет точка монтирования.

## Создание файловых систем

Для создания файловых систем в Linux используется команда `mkfs`:

```
[root@inefrno root]# mkfs -t ext3 /dev/hda6
```

На самом деле, `mkfs` является просто “оберткой” к реальным программам создания файловых систем, которые обычно именуются как `mkfs.<имя_ФС>`, например `mksf.ext2` или `mkfs.reiserfs`.

В большинстве случаев программы группы `mkfs` просто инициализируют специальную область раздела, называемую *суперблоком файловой системы*. Суперблок содержит ссылки на все значимые элементы файловой системы (например, ссылку на оглавление корневого каталога, ссылку список свободных блоков, ссылку на список файлов и т.д.)

Наличие суперблока (который практически всегда содержит в своем теле некоторое характерное значение) позволяет производить монтирование файловых систем без указания их типа. К сожалению, некоторые файловые системы вследствие своей архитектуры не содержат суперблока (в частности, FAT и большинство ее разновидностей) и для блочных устройств, содержащих такие файловые системы, автоматическое определение типа файловой системы затруднено.

## Интерфейс sysctl

Ядро содержит очень много параметров, от которых зависит его производительность и которые могут изменять алгоритмы его работы. Для того, чтобы иметь возможность узнавать и изменять эти параметры, в UNIX был разработан интерфейс sysctl.

Виртуальная файловая система procfs содержит каталог sys с деревом подкаталогов и файлов. Содержимое каждого из этих файлов можно прочесть, например, командой `cat`, или записать в такой файл новое значение командой `echo`:

```
[root@inferno root]# cat /proc/sys/kernel/shmmax
33554432
[root@inferno root]# echo 67108864 >/proc/sys/kernel/shmmax
[root@inferno root]# cat /proc/sys/kernel/shmmax
67108864
[root@inferno root]#
```

Команда `sysctl` предназначена для того, чтобы избежать необходимости использовать прямой доступ к этим файлам, и предоставить возможность автоматизации установки таких параметров при загрузке системы. На самом же деле, команда `sysctl` просто читает или записывает значения в файлы из каталога `/proc/sys`, т.е. если системный администратор устанавливает с помощью команды `sysctl` значение некоторого параметра, фактически он просто записывает это значение в соответствующий файл. Существует однозначное соответствие между именем параметра и именем файла, через который его можно изменить: если посмотреть вывод `sysctl -a`, можно увидеть, что параметры в большинстве своем именуются несколькими мнемоническими аббревиатурами, разделенными точками:

```
[root@inferno root]# sysctl -a | grep mem
net.ipv4.tcp_rmem = 4096      87380    174760
net.ipv4.tcp_wmem = 4096      16384    131072
net.ipv4.tcp_mem = 24576      32768    49152
net.ipv4.igmp_max_memberships = 20
net.core.optmem_max = 10240
net.core.rmem_default = 108544
net.core.wmem_default = 108544
```

Если в имени параметра заменить точки на символ разделителя пути (символ `/`), и к началу получившейся строки добавить `/proc/sys/` - то мы получим имя файла, через который можно изменить или прочесть значение соответствующего параметра.

Если системному администратору необходимо при каждой загрузке изменять некоторые параметры через интерфейс `sysctl`, то список параметров и их значений можно записать в конфигурационный файл `/etc/sysctl.conf`, который прочитывается при каждой загрузке системы.

## Статически и динамически собранные программы

В Linux исполняемые файлы можно условно поделить на две группы – те, которые содержат в себе весь код, необходимые для работы, и те, которым необходимы разделяемые библиотеки. Первые называют статически собранными бинарными файлами, вторые называют динамически собранными исполняемыми файлами.

Статически собранные программы характеризуются тем, что могут корректно функционировать в любых условиях, и не зависят от наличия или отсутствия разделяемых библиотек, что может оказаться полезным в ситуациях, когда возникают конфликты версий разделяемых библиотек, или когда системные библиотеки повреждены или недоступны (например во время восстановления операционной системы после серьезного сбоя). К недостаткам таких исполняемых файлов следует отнести то, что они имеют значительный размер и для обновления программы необходимо полностью заменить ее исполняемый файл – например,



если несколько статически собранных программ, которые работают с архивами ZIP, содержат ошибку, то для исправления ошибки необходимо заменить все эти программы, что может быть затруднено (например, будет трудно точно установить, какие именно программы содержат ошибочный код и нуждаются в обновлении). Кроме того, статически собранные программы не умеют совместно использовать совпадающие участки кода, что ведет к излишнему расходу системных ресурсов.

Динамически собранные исполняемые файлы для корректной работы требуют наличия файлов разделяемых библиотек, и соответственно при их отсутствии/повреждении не могут корректно функционировать, но зато для обновления программы и исправления ошибки часто оказывается достаточным просто заменить соответствующую разделяемую библиотеку, после чего ошибка исчезает во всех программах, которые эту библиотеку используют динамически. Динамически связанные программы также значительно меньше по объему, чем статически связанные, и код разделяемых библиотек может использоваться одновременно многими программами – что позволяет экономить системные ресурсы.

Подавляющее большинство программ в современных дистрибутивах Linux являются динамически собранными. Определить тип исполняемого файла (статический ли он либо с динамическим связыванием) можно, например, с помощью команды *ldd*:

```
[dalth@viking dalth]$ ldd /bin/su
linux-gate.so.1 => (0xffffe000)
libpam.so.0 => /lib/libpam.so.0 (0x4ce08000)
libpam_misc.so.0 => /lib/libpam_misc.so.0 (0x4cb3c000)
libcrypt.so.1 => /lib/libcrypt.so.1 (0x4e3a2000)
libdl.so.2 => /lib/libdl.so.2 (0x4ca49000)
libc.so.6 => /lib/tls/libc.so.6 (0x4c907000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x4c8ee000)
[dalth@viking dalth]$ ldd /sbin/devlabel
not a dynamic executable
[dalth@viking dalth]$
```

В данном случае мы видим, что исполняемый файл */bin/su* использует динамическое связывание, а исполняемый файл */sbin/devlabel* собран статическим образом.

## **Системная библиотека *GNU libc***

Основная системная библиотека, которая так или иначе используется практически всеми программами, называется *glibc* (GNU *libc*). Основными задачами *glibc* являются обеспечение взаимодействия между ядром и пользовательскими процессами, поддержка локализации и многие другие распространенные действия.

На нижнем уровне прикладные процессы могут обратиться к функциям ядра посредством системных вызовов (*syscall*). Фактически *syscall* – это вызов прерывания 80h с установленными параметрами, описывающими параметры для этого системного вызова. Те функции *glibc*, которые должны обратиться к ядру, в большинстве случаев просто устанавливают параметры для соответствующего системного вызова и вызывают *int80h*.

Большинство программ используют динамически загружаемую библиотеку *glibc*, но некоторые приложения, которые должны работать вне зависимости от наличия файловой системы, где расположена динамически загружаемая реализация *libc*, используют статическое связывание, когда весь программный код, необходимый для их работы, содержится в исполняемом файле программы. В основном к таким программам относятся утилиты, используемые при загрузке системы совместно с *initrd* – например, статические варианты утилит *insmod*, *lvm* или *devlabel*, а также командные оболочки для первичной загрузки или восстановления системы – например *sash* – standalone shell, часто используемый при восстановлении системы после серьезного сбоя или *nash*, используемый при выполнении сценариев загрузки после монтирования *initrd*, но до монтирования корневой файловой системы, когда

разделяемая версия glibc еще недоступна.

## ***LD, Shared Library, SO и много страшных слов***

Существует набор базовых действий, которые практически любая программа выполняет одинаково – открытие файла, чтение и запись данных и тому подобное. Разделяемые библиотеки предназначены для того, чтобы предоставить прикладным программам готовые интерфейсы функций для выполнения каких-либо более-менее стандартных действий. Разделяемая библиотека, как понятно из названия, может использоваться множеством программ. В настоящий момент стандартным форматом для разделяемых библиотек в Linux является ELF (Executable Linked Format).

Каждый файл ELF имеет заголовок, в котором описывается, какие секции содержит этот файл. Секции объединяют однотипные данные, и их детальное описание можно прочитать в справочном руководстве [man elf]. Мы же выделим следующую информацию: каждая библиотека содержит список имен переменных и функций, которые она содержит и предоставляет другим (экспортирует) и список переменных и функций, которые необходимо взять в других библиотеках, а также секции инициализации и деинициализации. Экспортируемые и импортируемые объекты (переменные и функции) называют *символами* библиотеки.

Большинство исполняемых файлов программ также имеют формат ELF, и на самом деле отличаются от библиотек в основном тем, что не имеют экспортируемых функций. Загрузчик ELF (он же dl, dynamic linker и dynamic loader) умеет загружать в память код ELF-файла, анализировать его структуру для определения списков экспортируемых и импортируемых символов и загружать необходимые для работы программы библиотеки.

Когда пользователь пытается запустить какую-либо программу, первым начинает работу загрузчик ELF. Он загружает в память процесса бинарный файл и выделяет, какие символы и из каких библиотек необходимо догрузить в память. После дозагрузки каждой библиотеки загрузчик связывает символы (проставляет реальные адреса) из загруженной библиотеки и повторяет цикл анализа на предмет того, какую библиотеку нужно загрузить. Когда все нужные библиотеки загружены, загрузчик передает управление коду инициализации каждой из загруженных библиотек в порядке, обратном загрузке, после чего передает управление коду программы. По завершении программы загрузчик снова “проходится” по всем библиотекам и вызывает их функции деинициализации. Если на этапе загрузки какой – либо библиотеке возникает ошибка, загрузчик сообщит об этом пользователю. Наиболее типичные ошибки dl – это не найденный файл библиотеки или неразрешимый символ (символ не был найден в библиотеке, в которой ожидался).

Вполне естественно, что загрузчик ищет библиотеки не по всей файловой системе, а только в определенных каталогах. Это каталоги /lib, /usr/lib и те, которые были перечислены системным администратором в файле /etc/ld.so.conf. Уточним, что этот файл на самом деле используется только системной утилитой ldconfig, сам же загрузчик использует кэш-файл /etc/ld.so.cache. Обновить этот кэш-файл можно путем простого запуска ldconfig без параметров. Следствием этого является то, что если вы установили в систему новые библиотеки, не мешает вызвать ldconfig.

В некоторых дистрибутивах есть возможность включать в ld.so.conf дополнительные файлы без его изменения. Для этого в ld.so.conf включается специальная строка вида:

```
include ld.so.conf.d/*.conf
```

Это приводит к тому, что каталоги, перечисленные в файлах с расширением conf, расположенных в каталоге /etc/ld.so.conf.d будут использованы для поиска разделяемых библиотек:

```
[root@viking dalth]# cat /etc/ld.so.conf
include ld.so.conf.d/*.conf
/usr/lib/mysql
/usr/X11R6/lib
/usr/lib/qt-3.3/lib
[root@viking dalth]# ls /etc/ld.so.conf.d/
oracle
[root@viking dalth]# cat /etc/ld.so.conf.d/oracle
/opt/oracle/9i/lib
[root@viking dalth]#
```

Нередко возникает ситуация, когда пользователю необходимо запустить какую-либо программу, которая не находится в каталогах, описанных в `/etc/ld.so.conf`. В таких ситуациях можно воспользоваться специальным “люком”, оставленным разработчиками `ld` специально для таких случаев: дело в том, что кроме загрузки библиотек с использованием данных из `ld.so.cache` загрузчик проверяет факт наличия библиотеки с указанным именем в каталогах, перечисленных в переменной среды `LD_LIBRARY_PATH`.

Разработчики часто используют еще одну возможность `ld`: если файл некоторой разделяемой библиотеки указан в переменной `LD_PRELOAD`, эта библиотека принудительно загружается и ее символы считаются более “приоритетными” и перекрывают одноименные символы, если таковые существуют в других библиотеках, загружаемых `ld` при запуске на выполнение бинарного файла ELF.

Попробуем рассмотреть примеры использования указанных возможностей `ld`: пусть есть некоторый программный продукт, в состав которого кроме собственно исполняемых программ входят разделяемые библиотеки (например, таковы практически все продукты, разработанные с помощью Borland Kylix). Если мы установим такой пакет, например, в `/opt/program`, его исполняемые файлы в `/opt/program/bin` а разделяемые библиотеки в `/opt/program/lib`, то программа, скорее всего, не будет запускаться, поскольку не сможет загрузить необходимых библиотек. Для того, чтобы программы пакета начали запускаться, мы должны “объяснить” `ld` где именно искать библиотеки. Рассмотрим возможные способы, которыми мы можем воздействовать на `ld` чтобы добиться нужного нам результата.

Первый способ – указать каталог с библиотеками перед запуском программы и уже затем запустить программу (`ld` воспользуется значением переменной для того, чтобы попытаться найти библиотеки по указанному пути):

```
$ export LD_LIBRARY_PATH=/opt/program/lib
$ /opt/program/bin/filename
```

Второй способ – добавить каталог `/opt/program/lib` в файл `/etc/ld.so.conf` и запустить `ldconfig`, решив проблему с невозможностью нахождения этих библиотек для всех программ сразу:

```
$ su -
# echo /opt/program/lib >>/etc/ld.so.conf
# ldconfig
# exit
$ /opr/program/bin/filename
```

Можно также воспользоваться возможностью принудительной загрузки тех библиотек, которые необходимы программе для запуска:

```
$ export LD_PRELOAD=/opt/program/lib/*
$ /opr/program/bin/filename
```

Большая часть кода разделяемых библиотек находится в кэше и становится доступна процессам через отображение файла в память. Это отображение делается с правами доступа “только чтение”, что защищает код библиотек от переписывания его неправильно работающими или просто злонамеренными программами.

## Информация о процессах и файловая систем /proc

Ядро и его подсистемы очень важны, но большинство пользы приносят прикладные задачи, поэтому мониторинг состояния задач (процессов) – очень важная часть работы системного администратора. В Linux получить информацию о процессах можно через файлы и каталоги файловой системы procfs, как правило монтируемой к каталогу /proc.

Каждому процессу сопоставляется в /proc отдельный каталог, имя которого совпадает со значением PID процесса. Файлы в этом каталоге предоставляют информацию о соответствующем процессе. Таблица приводит список файлов и их назначение:

Имя файла	Формат	Назначение
cmdline	строка, разделенная символами \0	Представляет командную строку, которой был запущен процесс. параметры командной строки отделяются друг от друга символами \0
environ	строка, разделенная символами \0	Представляет список переменных окружения для указанного процесса
exe	символьная ссылка	Ссылается на исполняемый файл процесса
maps	несколько строк	Список отображенных в память процесса файлов
mem	бинарный	Прямой доступ к адресному пространству процесса
mounts	несколько строк	Список примонтированных файловых систем, доступных процессу
stat	строка числовых значений	Статистика активности процесса
statm	строка числовых значений	Статистика по использованию памяти процессом
cwd	символьная ссылка	Ссылается на каталог, который является текущим для процесса
fd/*	символьные ссылки	Имена файлов подкаталога fd соответствуют открытым процессом дескрипторам файлов. Символьные ссылки указывают на соответствующие файлы
root	символьная ссылка	Ссылается на каталог, который процесс считает корневым
status	несколько строк	Описание состояния процесса

Все указанные данные полностью соответствуют тому, что показала бы программа *ps*, будучи запущенной в тот момент, когда просматривается соответствующий файл, поскольку утилита *ps* на самом деле просто читает данные из соответствующих файлов в /proc.

## Создание процессов

Linux на самом деле поддерживает только один внутренний механизм создания процессов – механизм *fork+exec*. Любой процесс, который хочет создать еще один процесс, должен сначала создать свою копию с помощью системного вызова *fork*, после чего порожденный процесс, который является полной копией предыдущего за исключением нескольких параметров, таких как PID (Process ID) и PPID (Parent Process ID) использует системный вызов *exec* для того, чтобы загрузить в свое

адресное пространство код новой программы и начать его выполнение. Соответственно, все процессы организуют дерево, когда у каждого процесса есть родительский процесс (исключение составляет процесс init, запущенный ядром на этапе загрузки).

Когда процесс завершается, код его завершения возвращается родительскому процессу. До тех пор, пока код завершения процесса не будет прочитан родительским процессом, запись об этом процессе продолжает существовать в таблице процессов в ядре. Такой процесс (уже завершившийся, но еще числящийся в таблице процессов) называют процессом – зомби (zombie process). Завершить zombie process может родительский процесс, прочтя код его завершения. Если у вас в системе появилось множество зомби – процессов, это скорее всего означает ошибку в программе, породившей этот процесс. Удалить процесс – зомби можно только удалив его родительский процесс.

Нередко бывает, что родительский процесс завершается раньше, чем дочерний, и тогда для дочернего процесса объявляется родительским процесс init, и поэтому в системе никогда не бывает процессов-”сирот”, т.е. тех, кто не имеет родителя и чей код завершения некому прочесть.

## Секреты /dev

Ядро Linux реализует поддержку двух типов устройств – символьных и блочных. Основное их отличие в том, что для блочных устройств операции ввода вывода осуществляются не отдельными байтами (символами), а блоками фиксированного размера.

В Linux вся работа с устройствами ведется через специальные файлы, которые обычно расположены в каталоге /dev. Специальные файлы не содержат данных, а просто служат точками, через которые можно обратиться к драйверу соответствующего устройства. У каждого специального файла есть три характеристики – тип устройства (character или block), старший номер устройства (major number) и младший номер (minor number). Для примера, посмотрим на содержимое каталога /dev:

[dalth@viking proc]\$ ls -lL /dev/hd* /dev/ttyS*									
brw-----	1	root	root	3,	0	Окт	1	20:16	/dev/hda
brw-----	1	root	root	3,	1	Окт	1	20:16	/dev/hda1
brw-----	1	root	root	3,	2	Окт	1	20:16	/dev/hda2
brw-----	1	dalth	disk	22,	0	Янв	1	1970	/dev/hdc
crw-----	1	root	root	4,	64	Янв	1	1970	/dev/ttyS0
crw-----	1	root	root	4,	65	Янв	1	1970	/dev/ttyS1
crw-----	1	root	root	4,	66	Янв	1	1970	/dev/ttyS2

Как видно, в листинге присутствует описание семи устройств, четырех блочных и трех символьных. Для каждого файла можно увидеть его тип (первая буква в списке прав доступа), пользователя-владельца, группу-владельца, major number, minor number, дату модификации и имя файла.

Для поддержки работы с устройствами в ядре хранятся две таблицы, одна для списка символьных устройств, другая для списка блочных устройств. Каждая строка таблицы сопоставлена какой-то разновидности устройств соответствующего типа – например, для типа “символьные устройства” можно выделить следующие разновидности: COM-порты, LPT-порты, PS/2-мыши, USB-мыши и т.д., для типа “блочные устройства” можно выделить SCSI-диски, IDE-диски, SCSI-CD-приводы, виртуальные диски которыми представляются RAID-контролеры и т.п.

Каждая ячейка в этих системных таблицах сопоставляется конкретному экземпляру устройства. Таким образом, с точки зрения ядра каждое устройство оказывается однозначно проидентифицировано тремя параметрами – типом устройства (блочное или символьное) и двумя числами – номерами строки и номером столбца таблицы, в которой хранится ссылка на драйвер этого устройства.

Пример таблицы символьных и устройств

	0	1	...	63	64	65	66	...	175	...
4					COM1	COM2	COM3			
6	LPT1	LPT2								
10		Мышь PS/2		Диспетчер томов LVM					Слот AGP	
14	Микшер первой зв. карты									
195	Первая видеокарта NVidia									

Пример таблицы блочных устройств

	0	1	2	...	16	...	64	65	...
3	IDE Primary Master	Раздел 1 на IDE Primary Master	Раздел 2 на IDE Primary Master		Раздел 16 на IDE Primary Master		IDE Primary Slave	Раздел 1 на IDE Primary Slave	
13	SCSI диск 1	Раздел 1 на SCSI-диске 1	Раздел 2 на SCSI-диске 1		SCSI диск 2		SCSI диск 4	Раздел 1 на SCSI-диске 4	

При попытке обращения к такому специальному файлу ядро переадресует обращение через нужный драйвер на устройство в соответствии с теми данными, которые указаны в таблице устройств, причем конкретная таблица устройств будет выбрана в зависимости от типа устройства, строка из таблицы будет выбрана по major number, и столбец будет выбран по minor number. Если мы посмотрим на примеры наших таблиц, то увидим, что обращение на специальный файл `/dev/ttyS1`, который представляет символьное устройство со старшим номером 4 и младшим номером 65 будет адресовано на последовательный порт COM2, а обращение к файлу `/dev/hda2` (блочное устройство со старшим номером 3 и младшим номером 2) будет адресовано на 2-й раздел жесткого диска IDE, работающего в режиме primary master.

### Статическая организация каталога `/dev`

В настоящее время существует два подхода к организации `/dev` – статическая организация и динамическая организация. В первом случае в каталоге `/dev` заранее создаются специальные файлы для всех возможных устройств вне зависимости от того, загружен драйвер соответствующего устройства или нет. Во втором случае специальные файлы в `/dev` создаются по мере инициализации устройств и загрузки драйверов, и удаляются при выгрузке соответствующего драйвера или удалении устройства.

Процесс работы со статическим `/dev` особых проблем не вызывает – системный администратор при необходимости просто создает отсутствующие файлы командой ***mknod*** или ***MAKEDEV***. В том случае, когда какая-либо программа обращается к устройству, чей драйвер не загружен (или загружен, но ни одного соответствующего устройства не было обнаружено), операционная система возвращает ошибку при попытке открытия файла такого “неверного” устройства. Ниже приведен пример создания специального файла, соответствующего блочному устройству с мажором 8 и минором 33 и попытка его использования (отметим, что этот специальный файл соответствует разделу на жестком диске, который не существует на тестовой машине, где выполнялись эти команды):

```
[root@vikings root]# cd /dev
[root@vikings dev]# ls -l /dev/hda33
ls: /dev/hda33: No such file or directory
[root@vikings dev]#
[root@vikings dev]# mknod hda33 b 8 33
[root@vikings dev]#
[root@vikings dev]# ls -l hda33
brw-r--r-- 1 root root 8, 33 Окт 11 09:27 hda33
[root@vikings dev]#
[root@vikings dev]# dd if=hda33 of=/dev/null
dd: opening `hda33': No such device or address
[root@vikings dev]#
```

Сообщение **No such device or address** как раз и означает, что записи для данного устройства в таблице блочных устройств не существует.

Ядро Linux совместно с некоторыми системными утилитами поддерживает такую интересную возможность, как загрузка драйверов “по требованию”. Реализуется это следующим образом – в момент, когда какая-либо программа пытается открыть специальный файл, не связанный ни с каким драйвером, ядро делает попытку подобрать соответствующий драйвер самостоятельно. Необходимый драйвер для каждого специального файла определяется в файле `/etc/modules.conf` путем задания специального алиаса (alias) для модуля. Для активизации автоматической загрузки драйвера какого-либо символьного устройства в большинстве случаев достаточно просто записать в `/etc/modules.conf` строку следующего вида:

```
alias char-major-X-Y имя_драйвера
```

Для блочных устройств соответствующая запись слегка меняет свою форму:

```
alias block-major-X-Y имя_драйвера
```

X и Y – это major и minor специального файла, попытка открыть который должна активизировать автоматическую загрузку драйвера. Владельцы видеокарт на чипе nVidia могут увидеть этот подход в действии – программа инсталляции драйвера nVidia автоматически прописывает в `modules.conf` запись для загрузки «по требованию» той части драйвера, которая работает в режиме ядра.

Вместо X или Y может также быть подставлен символ “\*”, означающий “любое число”. Например, пусть в `modules.conf` будет написан следующий текст:

```
alias char-major-81-* bttv
```

Тогда при обращении к любому символьному устройству с major number равным 81 и которое не ассоциировано ни с каким драйвером, система попытается загрузить драйвер bttv (драйвер TV-тюнера на основе чипа bt848).

Эта возможность обеспечивает Linux возможность плавной загрузки и эффективного использования ресурсов – драйвер не загружается, пока в нем не возникнет необходимости. К сожалению, за простоту этой схемы приходится платить большим количеством специальных файлов в `/dev`.

## Что такое DevFS

Для того, чтобы избавить администратора от ручного создания специальных файлов и для уменьшения количества файлов в `/dev` был реализован второй способ организации `/dev` – динамическое создание специальных файлов процессе загрузки драйверов. Реализовано это было следующим образом:

Ядро монтирует к каталогу `/dev` специальную файловую систему, называемую **devfs** – эта файловая система хранится целиком в оперативной памяти и не занимает никакого места на диске. Когда какой-либо драйвер в процессе загрузки или работы обнаруживает обслуживаемое им устройство, он регистрирует это устройство и сообщает о нем драйверу devfs. Драйвер devfs создает специальный файл, который виден прикладным программам и может быть корректно открыт. При выгрузке же драйвер устройства сообщает devfs о том, что соответствующее устройство уже не

активно, и драйвер `devfs` удаляет запись о соответствующем специальном файле из файловой системы `devfs`.

Файловая система `devfs` отличается тем, что как правило специальный файл для устройства создается с длинным путем – например, для раздела на SCSI-диске путь может выглядеть примерно так: **`/dev/scsi/host1/bus1/target3/lun4/partition2`**

Эта особенность является весьма важным плюсом `devfs`, поскольку она позволяет адресовать дисковые устройства путем указания логического пути их подключения и избежать смены имен SCSI-дисков в некоторых случаях (об этих случаях будет рассказано позднее).

Для того, чтобы организовать более прозрачную структуру каталогов и файлов устройств, используется специальный демон **`devfsd`**. Он взаимодействует с драйвером `devfs` и ядром и в процессе активизации и деактивизации устройств он создает и удаляет символьные ссылки вида `/dev/disks/disc0` или `/dev/hda1`.

Надо отметить, что схема динамического `/dev` в некотором смысле близка к той организации каталога `/dev`, которая используется некоторыми коммерческими UNIX-системами (например, в Solaris), когда есть виртуальная файловая система `/devices`, и на ее файлы создаются ссылки из `/dev`, только в Linux роль программы `cfgadm` играет демон `devfsd`, и все изменения в состав `/dev` вносятся автоматически.

С помощью `devfsd` файловая система `devfs` также реализует автоматическую загрузку модулей, но в этом случае выбор модуля идет не через комбинацию `type/major/minor`, а путем указания имени запрошенного файла – когда приложение пытается открыть несуществующий файл устройства, `devfs` передает имя запрошенного файла демону `devfsd`, и последний загружает необходимые модули, например такой код в файле `modules.devfs`:

```
alias /dev/nvidia* nvidia
```

Приведет к тому, что при попытке обращения к любому файлу, чей полный путь начинается строкой `/dev/nvidia`, будет произведена попытка загрузить драйвер `nvidia.o` (для ядра 2.6 `nvidia.ko`)

В принципе, на сегодняшний день выбор того, каким именно образом необходимо организовывать `/dev`, остается за пользователем и создателем дистрибутива. Например, в Mandrake Linux используется `devfs`, а в RedHat, Fedora и SUSE каталог `/dev` организован статическим образом, а опытные пользователи часто меняют способ организации `/dev` в зависимости от своих предпочтений.

## Немного о UDEV

В современных дистрибутивах и ядрах поддержка `devfs/devfsd` отключена, и на смену этой паре пришел специальный демон, называемый `udev`. В отличие от `devfsd`, который требовал поддержки со стороны ядра, `udev` такой поддержки не требует. При инициализации устройства ядро подает сигнал через файловую систему `sysfs`, и демон `udev`, получив сигнал об этом событии, самостоятельно создает соответствующий специальный файл устройства в каталоге `/dev` в соответствии с правилами, описанными в его конфигурационных файлах. При необходимости в этих файлах можно указать например вызов некоторой внешней программы, создание символьной ссылки и так далее.

Например, если некоторое устройство после подключения перед началом работы требует дополнительной настройки с использованием внешних программ, можно создать соответствующее правило для `udev`, в котором будет указано какую программу вызвать и какие параметры ей необходимо передать – в частности, это может потребоваться для data-кабелей к некоторым мобильным телефонам Nokia, для устройств которым для корректной работы требуется `firmware`, или для сохранения или восстановления текущих настроек устройства.

Тем не менее, несмотря на внешние отличия между статической организацией `/dev`, `devfs` и `udev`, следует помнить что это всего лишь способ заполнения каталога `/dev`, и



во всех случаях в конечном итоге на файловой системе создаются те же самые файлы символьных и блочных устройств.

## Блочные устройства

Любое устройство, подключенное к компьютеру, имеет свое назначение, и блочные устройства в большинстве своем предназначаются для хранения информации. Как организована работа с блочными устройствами в Linux?

Во-первых, следует определиться с типами блочных устройств. Их следует поделить на две категории: к первой отнесем логические (виртуальные) устройства (loop-устройства, software RAID-устройства, устройства Volume Management, поддержка различных таблиц разделов), ко второй категории – физические устройства (SCSI диски и CD-ROM'ы, IDE-диски, USB-storage, RAM-диск).

Виртуальные устройства являются на самом деле просто оберткой, дополнительным слоем. В реальности драйверы логических устройств не работают с периферийными устройствами напрямую, они лишь переадресовывают запросы на драйверы других логических или физических устройств.

Драйверы физических устройств работают совместно с драйверами контроллеров, позволяя производить доступ к соответствующим устройствам на блочном уровне и предоставляя тем самым фактически прямой доступ к носителю – но, поскольку в большинстве случаев дисковые устройства имеют значительный объем, они часто делятся на *разделы*. Раздел является постоянным непрерывным фрагментом дискового пространства, местоположение которого на жестком диске записано в специальной области диска – таблице разделов.

Существует множество различных форматов разбиения диска на разделы – например, DOS partition table, BSD disklabels, UnixWare slices и многие другие. Как правило, во всех случаях соответствующая спецификация предусматривает возможность перечисления ограниченного количества разделов путем указания номеров первой и последней дорожек, занимаемых каждым из разделов. Каждый раздел видится как отдельное блочное устройство.

По традиции имена блочных устройств, соответствующих IDE-дискам и созданным на них разделам начинаются с *hd* и имеют вид */dev/hd<N>[<M>]* где **N** – это буква, зависящая от контроллера и канала IDE, к которому подключено устройство, и режима устройства (master/slave). **M** – это некоторое число от 1 до 63 (фактически номер раздела на диске). Если число не указано, подразумевается весь диск. SCSI-дискам в */dev* присваиваются имена *sda*, *sdb*, *sdc* и т.д. Ниже приводится небольшая таблица соответствия устройств и имен специальных файлов для IDE-дисков:

Контроллер	Канал	Режим		Имя файла
1	1	master	primary master	hda
1	1	slave	primary slave	hdb
1	2	master	secondary master	hdc
1	2	slave	secondary slave	hdd
2	1	master	tertiary master	hde
2	1	slave	tertiary slave	hdf
2	2	master	quaternary master	hdg
2	2	slave	quaternary slave	hdh

Когда драйвер блочного устройства, поддерживающего разбиение на разделы, в процессе загрузки или работы обнаруживает обслуживаемое им устройство, он

считывает с него таблицу разделов, определяет ее разновидность и составляет таблицу разделов, запоминая начало и конец каждого из разделов. Впоследствии, если какая-либо программа производит обращение не непосредственно к физическому устройству, а к разделу на этом устройстве, драйвер с использованием построенной таблицы разделов определяет реальный адрес блока, над которым нужно произвести запрошенную операцию ввода/вывода.

Особой разновидностью раздела можно назвать расширенный (*extended*) раздел DOS. Расширенный разделы DOS может быть разбит на произвольное количество вложенных разделов, но в настоящий момент без использования LVM ядро Linux поддерживает до 63 разделов на IDE-диске и до 15 разделов на SCSI-диске. Такое ограничение связано с распределением мажоров и миноров блочных устройств.

Когда используется разбиение диска на разделы с использованием таблицы разделов DOS, следует помнить, что на жестком диске может быть не более 4 первичных разделов. Если администратору нужно, чтобы на жестком диске было более 4 разделов, необходимо объявить один из первичных разделов как расширенный раздел. Первичные разделы при использовании таблицы разделов DOS нумеруются от 1 до 4, логические разделы нумеруются начиная с 5, вне зависимости от количества первичных разделов. Только один из первичных разделов может быть объявлен расширенным.

Рассмотрим вывод команды `fdisk`, которая обычно используется в Linux для разбиения диска на разделы:

```
[root@stend root]# fdisk /dev/hda -l

Disk /dev/hda: 6442 MB, 6442450944 bytes
16 heads, 63 sectors/track, 12483 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

   Device Boot      Start         End      Blocks    Id  System
/dev/hda1  *           1           203       102280+   83  Linux
/dev/hda2             204          2032        921816   83  Linux
/dev/hda3          2033          3072        524160   82  Linux swap
/dev/hda4          3073          12483       4743144   f  W95 Ext'd (LBA)
/dev/hda5          3073          4088        512032+   83  Linux
/dev/hda6          4089         12483       4231048+   83  Linux

[root@stend root]#
```

Как видно, на жестком диске IDE созданы 6 разделов, из них 4 первичных (разделы с номерами от 1 до 4) и два логических раздела с номерами 5 и 6, созданных внутри *extended*-раздела `hda4`.

Таблица разделов диска не может быть изменена в том случае, если хотя-бы один из разделов этого диска используется. В этом случае ядро продолжает использовать “старую” разметку (с которой оно работало до изменения), а изменения записываются на диск и вступают в силу после перезагрузки компьютера.

Если нет возможности запустить программу `fdisk`, то для получения данных о разметке блочных устройств на разделы и о состоянии блочных устройств, доступных в настоящий момент, можно использовать некоторые файлы из файловой системы `/proc`:

```
[dalth@viking dalth]$ cat /proc/partitions
major minor  #blocks  name

    3        0   78150744 hda
    3        1   1084356 hda1
    3        2   77063805 hda2
  253        0    1048576 dm-0
  253        1    1048576 dm-1
  253        2    1048576 dm-2
  253        3    1048576 dm-3
  253        4    1048576 dm-4
  253        5   41943040 dm-5
  253        6    360448 dm-6
  253        7    258048 dm-7
  253        8    258048 dm-8
  253        9    258048 dm-9
  253       10     53248 dm-10
    7        0    651884 loop0
    7        1    650198 loop1
    7        2    653336 loop2
    7        3    198962 loop3
```

Например, эти данные могут быть использованы для восстановления таблицы разделов, если системный администратор по ошибке ее исправил.

Интересной особенностью таблицы разделов диска является то, что она не всегда изменяема. Это приводит к тому, что во многих случаях невозможно изменить размер какого-либо раздела или внести какие-либо другие изменения в таблицу разделов без перезагрузки – т.е. системный администратор может делать любые изменения, но они вступят в силу только после перезагрузки системы.

## Распределение мажоров и миноров IDE дисков

IDE-диски в настоящее время наиболее часто используются в офисных и домашних компьютерах, поэтому знать особенности распределения мажоров и миноров для этих типов устройств достаточно важно. IDE-устройства отличаются низкой ценой и неплохой скоростью передачи данных, но у этой шины есть архитектурные недостатки – например, все устройства работают со скоростью самого медленного из них, на один шлейф (на один канал) можно подключить только 2 устройства.

Всем устройствам, находящимся на одном канале IDE, присвоен один мажор. В настоящий момент ядро Linux выделяет для каждого устройства 64 минора, из которых первый минор зарезервирован для всего диска, и 63 минора остается для идентификации разделов. Таким образом, для IDE-дисков мажор идентифицирует канал, а по минору можно определить номер устройства на канале (режим master/slave) и номер раздела. Более детально это можно увидеть в следующей таблице:

Распределение номеров устройств для IDE-дисков					
Канал	Устройство	Раздел	Major number	Minor number	Имя в /dev
1	1	Весь диск	3	0	/dev/hda
		Раздел 1		1	/dev/hda1
		Раздел 2		2	/dev/hda2
		Раздел 3		3	/dev/hda3
		Раздел 4		4	/dev/hda4
		...			
		Раздел 63		63	/dev/hda63
	2	Весь диск		64	/dev/hdb
		Раздел 1		65	/dev/hdb1
		Раздел 2		66	/dev/hdb2
		Раздел 3		67	/dev/hdb3
		...			
		Раздел 63		127	/dev/hdb63
2	1	Весь диск	22	0	/dev/hdc
		Разделы		1-63	/dev/hdc[1..63]
	2	Весь диск		64	/dev/hdd
		Разделы		65-127	/dev/hdd[1..63]

По таблице становится видно, что на каждый 64-й минор происходит смена физического диска. Драйвер IDE в текущей версии ядра Linux поддерживает до четырех каналов IDE – т.е. до 8 устройств, по два устройства на канал..

Рекомендации, которую можно дать владельцам компьютеров с интерфейсом IDE: устройства по возможности рекомендуется держать на разных каналах. Если нет свободных каналов, то “быстрые” устройства лучше подключать на один канал, медленные – на другой.

## Распределение мажоров и миноров для SCSI-дисков

Шина SCSI свободна от некоторых недостатков IDE, например количество устройств на одном канале может быть 15 (на самом деле 16, но одним устройством считается сам контроллер), все SCSI устройства работают на своей максимальной скорости, и ограничены только возможностями шины и контроллера – но SCSI-устройства и дорожке, и поэтому шина SCSI используется в основном на серверах и рабочих станциях. Для SCSI-дисков ситуация немного меняется – мажоры не привязаны к контроллерам (т.е. один major number может использоваться дисками с разных хост-адаптеров). На каждом SCSI-диске система поддерживает до 16 разделов, а нумерация дисков производится в порядке их подключения по схеме, аналогичной IDE-дискам – но только переход на следующий диск происходит на каждом 16-м миноре (т.е. разделы на SCSI-дисках нумеруются от 1 до 15). Увидеть это можно в следующей таблице и листинге /dev:

Распределение номеров устройств для SCSI-дисков				
Номер диска в порядке подключения	Major number	Minor number	Раздел	Имя файла
1	8	0	Весь диск	sda
		1	Первый первичный	sda1
		2	Второй первичный	sda2
		3	Третий первичный	sda3
		4	Четвертый первичный	sda4
		5	Первый логический	sda5
		...		
		15	Одиннадцатый логический	sda15
2	8	16	Весь диск	sdb
		17	Первый первичный	sdb1
		18	Второй первичный	sdb2
		19	Третий первичный	sdb3
		20	Четвертый первичный	sdb4
		21	Первый логический	sdb5
		...		
		31	Одиннадцатый логический	sdb15

В текущей версии ядро Linux поддерживает до 4096 SCSI-дисков – и для абсолютного большинства компьютеров этого должно быть достаточно. Очень важной особенностью SCSI-дисков является то, что мажор устройства, соответствующего физическому диску, не зависит от контроллера. В результате, если у вас есть 3 диска SCSI, то они всегда именуются sda, sdb и sdc, и если вы выключите компьютер и отключите первый диск, то после перезагрузки второй диск (который ранее назывался sdb) станет называться sda, а третий диск (который назывался sdc) станет называться sdb, поэтому при работе со SCSI-устройствами можно использовать devfs (которая позволяет адресовать диски через путь их подключения), либо использовать специализированные средства управления дисковым пространством, которые помечают носители и впоследствии правильно их идентифицируют даже после переименования устройств - например, средства md (software RAID) или LVM. Надо заметить, что в свете удаления devfs из основной ветви ядра, использование LVM стало фактически обязательным на серверах.

## Устройства SATA и переход с IDE на PATA

SATA, «осовремененная» версия интерфейса IDE, по своей структуре приблизилась к подсистеме SCSI. Поэтому в целях унификации подсистемы ввода-вывода в ядре Linux поддержка SATA была реализована через интерфейс SCSI., соответственно SATA-диски и контроллеры видятся ядром (и пользователем) как SCSI-устройства.

В новейших ядрах линейки 2.6 также появилась возможность работы с IDE-дисками через подсистему SCSI., и в новых дистрибутивах, таких как Fedora 7, даже обычные

Parallel ATA (они же IDE) диски и контроллеры представляются как SCSI-устройства, что привело к унификации подсистемы дискового ввода-вывода, и теперь все диски и CD/DVD приводы для пользователя представляются как `/dev/sdX` или `/dev/scdX`. Драйверы для обычных контроллеров IDE, которые работают по новой схеме, начинаются с префикса `pata`, например: `pata_via` – это драйвер IDE-контроллеров с чипсетом VIA, а `pata_piix` – это драйвер для IDE-контроллеров Intel, работающие через подсистему SCSI.

А говоря проще, это означает следующее – если у вас новый дистрибутив или имеются SATA-диски, вы можете смело работать со всеми дисками как со SCSI-устройствами.

## Logical Volume Manager

Использование таблиц разделов для управления дисковым пространством – достаточно часто используемое решение. К сожалению, оно не свободно от определенных недостатков – например, нет возможности расширить раздел или уменьшить его размер, нет возможности создать один раздел на нескольких дисках и т.д. Решить эту задачу призван LVM (Logical Volume Manager).

LVM работает следующим образом: пользователь может пометить какие-либо блочные устройства как разделы, используемые LVM. Каждое из таких помеченных блочных устройств (их называют физическими томами, или `physical volumes`) может быть присоединено к какой-либо группе логических томов (`logical volume groups`). Внутри групп логических томов могут создаваться уже собственно логические тома (`logical volumes`). Дисковое пространство любого физического тома из некоторой группы может быть выделено любому логическому тому из этой группы. Реализовано это через так называемые “экстенты” (`extents`) дискового пространства. Физические тома LVM разбиваются на экстенты, после чего из экстенентов и составляются логические тома. Именно за счет этого можно динамически менять конфигурацию дискового пространства – экстенмент может быть удален из одного тома, и добавлен к другому. Каждый объект LVM – будь то логический том, физический том или группа томов, имеет свой уникальный идентификатор.

Осуществляется это комплексно драйвером *device mapper* и специализированными программами из пакета `lvmd`. Эти программы читают файлы конфигурации и служебную информацию из заголовков физических томов, и на основании этой информации сообщают драйверу инструкции о том, из каких фрагментов каких блочных устройств каким именно образом должны быть скомбинированы логические тома, после чего драйвер для каждого логического тома создает отдельное блочное устройство. При обращении к такому блочному устройству *device mapper* определяет, на основании ранее переданных параметров к какому блоку какого блочного устройства на самом деле должен быть переадресован запрос, и запрашивает соответствующее блочное устройство для окончательного выполнения операции, после чего возвращает результат выполнения операции (прочитанные данные, сообщение об ошибке, код завершения операции) обратившейся программе.

Использование LVM позволяет гибко управлять распределением дискового пространства и избежать ограничений, связанных с классическим распределением дискового пространства путем создания разделов на жестких дисках. Единственное правило, которое я бы советовал соблюдать при использовании LVM – не создавать корневой раздел системы на логическом томе LVM: инициализации тома LVM, на котором находится корневая файловая система, необходимо вмешательство некоторых утилит, которые находятся на еще не смонтированной корневой файловой системы. Это решаемая проблема, но она потребует некоторого опыта.

Ниже идет пример создания и инициализации физического тома, группы томов и примеры нескольких операций с логическими томами. На первом фрагменте

протокола продемонстрирована инициализация таблицы разделов для использования LVM. Порядок действий для использования LVM в общем случае следующий: один или несколько разделов жесткого диска с помощью fdisk помечаются как разделы LVM. Затем эти разделы инициализируются и передаются в группы томов, после чего их дисковое пространство можно использовать для создания логических томов:

```
[root@inferno dalth]# fdisk /dev/hdb

The number of cylinders for this disk is set to 79408.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
 1) software that runs at boot time (e.g., old versions of LILO)
 2) booting and partitioning software from other OSs
    (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/hdb: 40.9 GB, 40982151168 bytes
16 heads, 63 sectors/track, 79408 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hdb1           1         79408     40021600+   8e  Linux LVM

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
[root@inferno dalth]#
```

В приведенном выше выводе fdisk видно, что на IDE-диске primary slave создан один раздел типа LVM (код типа раздела 0x8E). В следующем листинге показан процесс инициализации физического тома и создания группы томов **aurora**, в которую включается инициализированный командой **pvccreate** физический том **/dev/hdb1**:

```
[root@inferno dalth]# pvcreate /dev/hdb1
No physical volume label read from /dev/hdb1
Physical volume "/dev/hdb1" successfully created
[root@inferno dalth]# vgcreate aurora /dev/hdb1
Volume group "aurora" successfully created
[root@inferno dalth]#
```

Третий фрагмент демонстрирует создание нескольких томов и изменение размеров томов, сделанное с помощью LVM. В этом примере создается логический том размеров в 20GB, затем размер этого тома увеличивается до 30GB, создается еще один логический том, и после этого оба созданных логических тома удаляются.

```

[root@inferno dalth]#
[root@inferno dalth]# lvcreate -L 20G -n ftpdata aurora
Logical volume "ftpdata" created
[root@inferno dalth]#
[root@inferno dalth]# lvscan
ACTIVE                               '/dev/aurora/ftpdata' [20,00 GB] next free
(default)
[root@inferno dalth]#
[root@inferno dalth]# lvresize -L +10G /dev/aurora/ftpdata
Extending logical volume ftpdata to 30,00 GB
Logical volume ftpdata successfully resized
[root@inferno dalth]#
[root@inferno dalth]# lvscan
ACTIVE                               '/dev/aurora/ftpdata' [30,00 GB] next free
(default)
[root@inferno dalth]# lvcreate -L 8G -n home_dirs aurora
Logical volume "home_dirs" created
[root@inferno dalth]#
[root@inferno dalth]# lvscan
ACTIVE                               '/dev/aurora/ftpdata' [30,00 GB] next free
(default)
ACTIVE                               '/dev/aurora/home_dirs' [8,00 GB] next free
(default)
[root@inferno dalth]#
[root@inferno dalth]# lvremove /dev/aurora/ftpdata
Do you really want to remove active logical volume "ftpdata"? [y/n]:
y
Logical volume "ftpdata" successfully removed
[root@inferno dalth]#
[root@inferno dalth]# lvremove /dev/aurora/home_dirs
Do you really want to remove active logical volume "home_dirs"?
[y/n]: y
Logical volume "home_dirs" successfully removed
[root@inferno dalth]#

```

Еще один фрагмент демонстрирует удаление группы томов и очистку физического тома:

```

[root@inferno dalth]# vgremove aurora
Volume group "aurora" successfully removed
[root@inferno dalth]# pvremove /dev/hdb1
Labels on physical volume "/dev/hdb1" successfully wiped
[root@inferno dalth]#

```

Каждый логический том LVM имеет свой собственный minor number, а major number для всех томов LVM равен 253. Для доступа к томам LVM можно создавать блочные устройства с помощью команды *mknod*, а можно воспользоваться возможностями, предоставляемыми утилитой *devlabel*. Эта утилита создает символичные ссылки и каталоги внутри подкаталога */dev*, причем для каждой группы томов в */dev* создается каталог с именем этой группы, а логические тома представляются символическими ссылками из этих каталогов на блочные устройства, обслуживаемые драйвером *device mapper*, и тогда любой том LVM можно адресовать следующим путем: */dev/<имя\_группы>/<имя\_тома>*. На листинге ниже показан пример того, как можно распределить дисковое пространство с помощью LVM:



```

[root@viking root]#
[root@viking root]# ls -la /dev/chimera
lr-xr-xr-x 1 root root 23 Okt  8 13:53 opt -> /dev/mapper/chimera-
opt
lr-xr-xr-x 1 root root 24 Okt  8 13:53 swap -> /dev/mapper/chimera-
swap
lr-xr-xr-x 1 root root 24 Okt  8 13:53 temp -> /dev/mapper/chimera-
temp
lr-xr-xr-x 1 root root 23 Okt  8 13:53 usr -> /dev/mapper/chimera-
usr
lr-xr-xr-x 1 root root 23 Okt  8 13:53 var -> /dev/mapper/chimera-
var
[root@viking root]#
[root@viking root]# lvscan
ACTIVE          '/dev/chimera/swap' [1,00 GB] next free (default)
ACTIVE          '/dev/chimera/temp' [1,00 GB] next free (default)
ACTIVE          '/dev/chimera/usr' [10,00 GB] next free (default)
ACTIVE          '/dev/chimera/opt' [10,00 GB] next free (default)
ACTIVE          '/dev/chimera/var' [1,00 GB] next free (default)
[root@viking root]#
[root@viking root]# mount | grep chimera
/dev/mapper/chimera-var on /var type ext3 (rw)
/dev/mapper/chimera-temp on /tmp type ext3 (rw)
/dev/mapper/chimera-usr on /usr type ext3 (rw)
/dev/mapper/chimera-opt on /opt type ext3 (rw)
[root@viking root]#

```

Таким образом, возможности LVM позволяют системному администратору максимально эффективно использовать дисковое пространство, оперативно реагируя на меняющиеся условия эксплуатации. Еще одной интересной возможностью LVM является так называемый multipath I/O. В случае активации соответствующей опции в ядре device mapper знает о том, что физический том с некоторым UUID может быть доступен через несколько контроллеров, и в случае отказа одного контроллера динамически происходит переключение ввода-вывода на другой. Опытные системные администраторы также оценят такую возможность, как создание снимка (snapshot) логического тома: при создании снимка создается моментальная копия логического тома, которая начинает «жить» независимо от того тома, на основе которого она была создана:

```

# xfs_freeze /home
# lvcreate -s -L 10G -n home_snapshot /dev/chimera/home
# xfs_freeze -u /home
# dd if=/dev/chimera/home_snapshot of=/dev/st0
# lvremove /dev/chimera/home_snapshot

```

В приведенном примере системный администратор «замораживает» файловую систему XFS, при этом драйвер XFS сбрасывает все закэшированные операции на диск, и после этого блокирует все процессы, которые пытаются писать на «замороженную» файловую систему. Затем системный администратор создает снимок тома home из группы томов chimera, на котором «живет» файловая система /home, и этот снимок называет home\_snapshot, при этом на удержание копии измененных данных выделяется 10 гигабайт дискового пространства. После создания снимка файловая система /home размораживается, но каждый раз, когда будет переписываться какой-либо блок логического тома home, первоначальная версия изменяемого блока будет копироваться в те 10GB дискового пространства, которые мы выделили под снимок тома, и мы можем считать содержимое тома home\_snapshot неизменным, и скопировать его на ленту. В процессе чтения, если читаемый блок не изменялся с момента создания снимка, то он читается из исходного тома (home), если же блок менялся с момента создания снимка, то используется его копия, хранящаяся в зарезервированном при создании снимка пространстве. После окончания копирования мы удаляем снимок командой lvremove.

## Software RAID

Ядро Linux содержит средства для организации software raid (программных RAID-устройств). Эта возможность поддерживается драйвером устройств *md*. В отличие от device mapper, драйвер *md* умеет работать в “самостоятельном” режиме, получая конфигурацию из параметров, которые пользователь указал ядру при загрузке системы, что позволяет организовывать загрузку системы с RAID-устройств. Все устройства *md* имеют мажор 254 и миноры от 0 и до 16383.

В отличие от LVM, основной задачей которого является динамическое распределение дискового пространства (деление разделов на фрагменты и построение из фрагментов новых блочных устройств), задачей подсистемы RAID является построение новых блочных устройств путем объединения существующих. Каждое из устройств, входящих в создаваемый дисковый массив, может определенным образом помечаться. Впоследствии эти метки (их также называют *array superblocks*) используются для повторной сборки массива. В частности, например, суперблок массива содержит его уникальный идентификатор, который можно использовать при сборке ранее созданного массива после перезагрузки. Если программный RAID-массив был помечен в процессе создания (т.е. на нем был создан суперблок массива), это дает возможность автоматической сборки массива вне зависимости от того, поменялся или нет порядок следования устройств. Например, такая необходимость может возникнуть в ситуации, когда порядок нумерации блочных устройств изменился – например, один из SCSI-дисков, участвовавших в построении массива, был удален.

Естественно, суперблок не является обязательным – то есть можно создавать массивы без суперблока, но управления ими может быть затруднено вследствие необходимости «руками» контролировать корректность указания устройств при переконфигурации массива.

Из интересных особенностей драйвера *md* стоит отметить то, что он поддерживает разбиение *md*-устройств на разделы, при этом минорные номера присваиваются разделам аналогично тому, как они присваиваются разделам на дисках IDE, т.е. на каждом RAID-устройстве можно создать до 63 разделов. Определить минор раздела, созданного на RAID-устройстве, можно с помощью вычисления значения следующего выражения:  $64 * N + M$ , где *N* – это номер массива (номер RAID-устройства) из диапазона 0 ... 255, а *M* – это номер раздела из диапазона 1 ... 63.

Следует сказать, что по умолчанию в большинстве дистрибутивов специальные файлы для разделов на *md*-устройствах не создаются, и их необходимо создать вручную командой *mknod*. В настоящий момент оптимальным, наверное, следует считать комбинирование использования LVM и *md*, что позволяет достигнуть надежности за счет дублирования данных средствами *md*, и гибкости распределения дискового пространства за счет возможностей LVM.

Драйвер *md* хорошо подходит для создания RAID-устройств уровней 0, 1 или 0+1, но не будет являться оптимальным вариантом в случае использования, например RAID уровня 5 (чередование данных по устройствам с вычислением контрольной суммы и кодом исправления ошибок), поскольку это создаст значительную нагрузку на процессор при большом объеме передаваемых данных. Возможно, что в таких случаях стоит подумать о приобретении аппаратного контроллера RAID - например, HP NetRaid (сделан на основе AMI MegaRAID) или Compaq Smart Array (сейчас называется HP Smart Array).

На листинге демонстрируется пример создания, активизации и остановки программных RAID-устройств уровня 0 и уровня 1 средствами драйвера *md* и системной утилиты *mdadm*. Утилита *mdadm* имеет конфигурационный файл */etc/mdadm.conf*, но для того, чтобы проделать некоторые тесты и демонстрационные примеры нет необходимости его изменять.

В первом примере будем считать, что на жестком диске *hdb* создано два раздела, с

которыми мы и будем экспериментировать. Для начала необходимо произвести инициализацию md-устройства. Соответственно, для успешной необходимо указать тип RAID-массива, специальный файл md-устройства, которое мы хотим инициализировать, и список блочных устройств, на которых будет располагаться получившийся массив:

```
[root@inferno dalth]#
[root@inferno dalth]# mdadm --create \
>                               /dev/md0 --level=0 \
>                               --raid-devices=2 /dev/hdb1 /dev/hdb2
mdadm: array /dev/md0 started.
[root@inferno dalth]#
[root@inferno dalth]# mdadm -Q /dev/md0
/dev/md/d0: 983.25MiB raid0 2 devices, 0 spares.
[root@inferno dalth]#
[root@inferno dalth]# mdadm -S /dev/md0
[root@inferno dalth]#
[root@inferno dalth]# mdadm --create \
>                               /dev/md0 --level=1 \
>                               --raid-devices=2 /dev/hdb1 /dev/hdb2
mdadm: array /dev/md0 started.
[root@inferno dalth]#
[root@inferno dalth]# mdadm -Q /dev/md0
/dev/md0: 491.63MiB raid1 2 devices, 0 spares.
[root@inferno dalth]#
[root@inferno dalth]# mdadm -S /dev/md0
[root@inferno dalth]#
[root@inferno dalth]# mdadm --assemble /dev/md0 /dev/hdb1 /dev/hdb2
mdadm: /dev/md0 has been started with 2 drives.
[root@inferno dalth]#
```

Последняя команда демонстрирует активизацию массива путем указания имени md-устройства и нескольких блочных устройств, на которых оно базируется. Все остальные параметры (размеры блоков, разновидность RAID и т.д.) утилита mdadm извлекла из суперблока массива. Как уже отмечалось, суперблок массива содержит еще и уникальный идентификатор массива, что дает возможность проидентифицировать каждое исходное блочное устройство на предмет его принадлежности к какому-либо массиву. Ниже приведен пример вывода утилиты **mdadm**, демонстрирующий как можно получить некоторые полезные данные о массиве, а строка, содержащая UID массива выделена жирным текстом:

```
[root@viking root]# mdadm -Q -D /dev/md0
/dev/md0:
    Version : 00.90.01
    Creation Time : Fri Oct  8 14:29:21 2004
    Raid Level : raid1
    Array Size : 102336 (99.94 MiB 104.79 MB)
    Device Size : 102336 (99.94 MiB 104.79 MB)
    Raid Devices : 2
    Total Devices : 2
    Preferred Minor : 0
    Persistence : Superblock is persistent

    Update Time : Fri Oct  8 14:32:22 2004
    State : clean, no-errors
    Active Devices : 2
    Working Devices : 2
    Failed Devices : 0
    Spare Devices : 0

    Number   Major   Minor   RaidDevice State
     0         253      24         0      active sync  /dev/hdb1
     1         253      25         1      active sync  /dev/hdb2
    UUID : 8696ffc0:52547452:ba369881:d1b252d0
    Events : 0.3
[root@viking root]#
```

Впоследствии этот идентификатор может быть использован в файле конфигурации для утилиты mdadm. В конфигурационном файле /etc/mdadm.conf можно указать список устройств и правила их построения, после чего описанные в нем md-устройства будут автоматически собираться и разбираться без указания списка исходных устройств:

```
[root@viking root]# cat /etc/mdadm.conf
MAILADDR root
ARRAY /dev/md0 UUID=8696ffc0:52547452:ba369881:d1b252d0
DEVICE /dev/hdb*
[root@viking root]#
```

В листинге видно, что устройство массив md0 имеет указанный идентификатор, а также указано, что для построения массивов могут быть использованы все разделы диска hdb. В этом примере если системный администратор напишет команду **mdadm --assemble /dev/md0**, то mdadm просканирует все файлы устройств с именами, совпадающими с шаблоном /dev/hdb\* и подключит к массиву md0 те из них, на которых будет найдена суперблок массива с тем идентификатором, который указан в параметре **ARRAY** для устройства /dev/md0.

Системный администратор, который хочет расположить корневую файловую систему на md-устройстве, должен указать ядру при загрузке какие именно физические блочные устройства должны входить в md-устройство, на котором содержится корневая файловая система. Обычно это делается путем загрузки ядра командной строкой с опциями следующего вида:

```
linux md0=0,/dev/hdb1,/dev/hdb2 root=/dev/md0
```

Данный пример приведен скорее как иллюстративный, поскольку в зависимости от опций, использованных при создании RAID-устройства, на котором расположена корневая файловая система, командная строка ядра может меняться. В современных дистрибутивах при необходимости инициализации md-устройств для загрузки системы, как правило код и утилиты инициализации устройства помещаются в initrd.

Суперблок массива записывается не в начале блочного устройства, а ближе к его середине или концу. Сделано это было для того, чтобы можно было создать RAID-массив с boot-сектором, который сможет быть прочитан не только ядром Linux с

драйвером *md*, но и базовым загрузчиком BIOS, вследствие чего можно объединить в RAID-массив не *разделы* жестких дисков, а непосредственно *физические диски*. Тогда загрузчик, установленный в начало RAID-устройства, окажется установленным в начало жесткого диска, после чего можно использовать при загрузке ядра следующую командную строку:

```
linux md0=d0,/dev/hda,/dev/hdb root=/dev/md_d0p1
```

Драйвер *md* также поддерживает возможность задания hotswap-устройств для массивов, т.е. резервных устройств, которые могут быть активизированы в при сбое одного из основных устройств в массиве.

Поддержка устройств software RAID в Linux дает возможность создавать серверы с высокой отказоустойчивостью и быстродействием.

## Device mapper

В ядрах линейки 2.6 появилась еще одна подсистема по некоторым функциям аналогичная подсистеме MD, и называемая **device-mapper**. Это модульная компонентная подсистема, позволяющая с помощью специальных команд создать одно блочное устройство из нескольких кусков других блочных устройств, а также определить правила, по которым производится запись на эти «нижележащие» блочные устройства.

LVM работает именно через подсистему device mapper, и на самом деле все утилиты LVM на самом деле просто передают инструкции о том из каких фрагментов каких блочных устройств состоит какой том LVM в драйвер device mapper, в каком порядке осуществляется запись и чтение данных, и впоследствии при записи на том LVM или чтении с него, работа на самом деле ведется с устройствами, обслуживаемыми драйвером device mapper, который и делает всю работу.

Данная многоуровневая архитектура позволяет значительно упростить и таким образом значительно повысить стабильность работы системы, поскольку реализация нескольких небольших узкофункциональных компонентов в общей сложности содержит меньше ошибок, чем реализация всех этих функций в одной подсистеме.

## Host-RAID, или дешевых RAID-контроллеров не бывает

Сегодня даже для дешевых современных материнских плат фирмы-производители часто декларируют «аппаратную поддержку RAID» и у многих пользователей этот факт вызывает недоумение – как же так, мой Linux не умеет работать с RAID?! **На самом деле все проще – задекларированная и разрекламированная поддержка RAID-массивов на материнских платах для офисных и домашних компьютеров – это миф.**

Вся поддержка RAID в таких «контроллерах» на самом деле представляют собой просто небольшое расширение в BIOS и без специальных драйверов в 32/64-битных операционных системах не работают, а все функции RAID для них выполняются драйвером. Если в Windows для каждого из таких контроллеров фирма-производитель пишет драйвер, то в Linux ситуация немного иная.

Для реализации возможности работы с такими псевдо-RAID контроллерами (иногда называемыми fake-RAID) была разработана утилита **dmraid**. При запуске она сканирует жесткие диски в поисках специальных блоков (функционально аналогичных суперблокам уже знакомых нам md-устройств), записываемых такими fake-RAID контроллерами, и если ей удалось распознать формат этого специального блока, то dmraid инструктирует подсистему device-mapper о том, в каком порядке следует считывать блоки с жестких дисков.

После этого device-mapper создает блочное устройство, при записи или чтении данных с которого данные автоматически читаются и пишутся так, как сделал бы это драйвер от производителя материнской платы или контроллера.

## Сетевая подсистема

Ключевым (с нашей точки зрения) объектом сетевой подсистемы Linux является интерфейс. Сетевой интерфейс в Linux – это абстрактный именованный объект, используемый для передачи данных через некоторую линию связи без привязки к ее (линии связи) реализации. Конечно, сказано мудрено – но попробуем объяснить «на пальцах».

Например, если в системе существует интерфейс *eth0*, то в большинстве случаев на современных компьютерах он сопоставлен Ethernet-адаптеру, встроенному в материнскую плату. Интерфейс с именем *ppp0* отвечает за некоторое соединение «точка-точка» с другим компьютером. Интерфейс с именем *lo* является виртуальным и представляет как бы замкнутый сам на себя (вход непосредственно подключен к выходу) сетевой адаптер.

Основная задача интерфейса – абстрагироваться от физической составляющей канала. То есть программы и система будут использовать один и тот же метод «отправить пакет» для отправки данных через любой интерфейс – хоть *lo*, хоть *ethX*, хоть *pppY*, и точно так же использовать один и тот же метод «принять пакет» – то есть создается унифицированный API передачи данных, независимый от носителя.

Для того, чтобы ознакомиться с интерфейсами, можно воспользоваться командой *ifconfig*:

```
$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:11:2F:A8:DE:A4
          inet  addr:172.23.2.114      Bcast:172.23.2.255    Mask:
          255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:11 Base address:0x4000

lo        Link encap:Local Loopback
          inet  addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:60 errors:0 dropped:0 overruns:0 frame:0
          TX packets:60 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4707 (4.5 KiB)  TX bytes:4707 (4.5 KiB)
```

В данном случае мы видим в системе два активных интерфейса, *eth0* и *lo*, а также некоторую информацию об их состоянии, настройках и параметрах и состоянии аппаратуры для интерфейса *eth0*. В частности, поле *Link encap* характеризует тип интерфейса, *HWAddr* – аппаратный адрес устройства (например, MAC-адрес для Ethernet), *MTU* – максимальный размер передаваемого пакета. Также могут представлять интерес поля статистики – они сообщают, какой объем данных был передан и получен через соответствующий интерфейс.

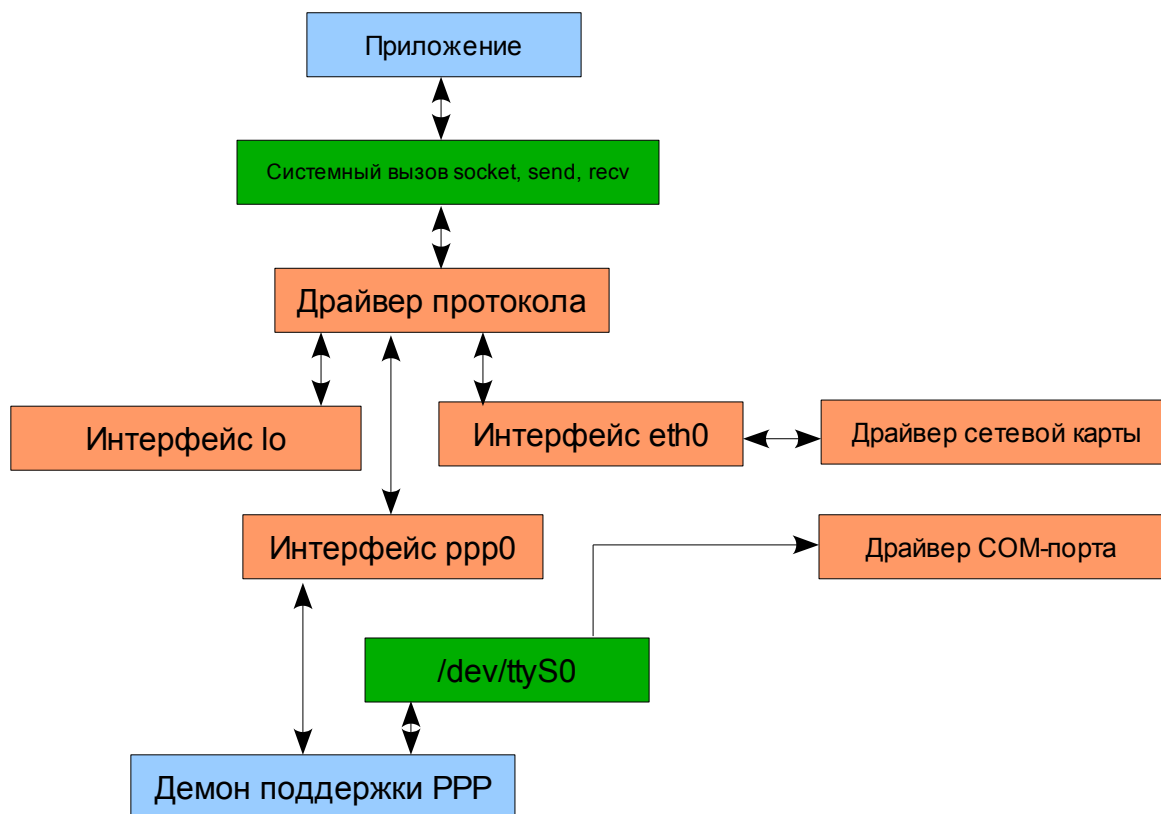
Наиболее часто встречающиеся типы интерфейсов:

- *eth* (Ethernet) – обычно соответствует отдельному сетевому адаптеру
- *ppp* (Point-To-Point) – соединение точка-точка, например при коммутируемом доступе, но очень часто используется и при организации VPN
- *slip* (Serial Line IP) – соединение точка-точка, устаревший протокол
- *wl* (Wireless) – беспроводной интерфейс, обычно сопоставлен соответствующему адаптеру
- *lo* (Loopback) – интерфейс-«петля», то есть что послал, то и получил - используется для общения между сетевыми приложениями в рамках одного компьютера

Команда *ifconfig* может также использоваться для остановки или активации интерфейса, а также изменения его параметров, связанных с протоколом IP:

```
# ifconfig eth0 down
# ifconfig eth0 up
# ifconfig eth0 inet 192.168.2.210 netmask 255.255.255.0
# ifconfig eth0 mtu 296
```

Для управления параметрами других протоколов используются другие команды – например, *ipx\_config* для управления параметрами, связанными с протоколом IPX. Рассмотрим также картинку, на которой приведена приблизительная схема взаимодействия различных драйверов и сетевых подсистем ядра:



Предположим, что приложение пытается отправить пакет. Перед отправкой через системные вызовы группы *socket* (*bind*, *connect* и пр.) приложение настраивает специальный файловый дескриптор. После окончания настройки каждый записанный в этот дескриптор пакет должен быть отправлен по сети получателю. Как движется пакет в нашей системе? Прежде всего, пакет попадает в драйвер протокола. Этот драйвер определяет через какой интерфейс должна производиться отправка, дописывает к пакету необходимые заголовки и отдает пакет на обработку соответствующему интерфейсу (точнее, ставит пакет в очередь, связанную с этим интерфейсом).

Что драйвер сделает с пакетом, это уже его дело. Например, драйвер интерфейса *loopback* этот пакет вынет из очереди и сразу поставит в очередь «принятых», откуда его впоследствии заберет драйвер протокола (левая цепочка на схеме). Драйвер интерфейса *eth0* допишет к пакету заголовки Ethernet и передаст пакет драйверу сетевого адаптера, и уже тот непосредственно проинструктирует сетевой адаптер, откуда взять и как отправить пакет (правая цепочка). В средней же цепочке мы видим схему работы PPP, когда пакет помещается в очередь интерфейса *ppp0*, откуда его заберет демон *pppd*. Демон допишет в пакет нужные заголовки, и через символьный специальный файл */dev/ttyS0* передаст пакет драйверу COM-порта, а тот непосредственно будет работать с аппаратурой. Соответственно, при приеме

данных цепочки проходятся в обратном порядке.

## Маршрутизация IP и форвардинг

Маршрутизация транзитных IP-пакетов (не предназначенных для этого компьютера), или IP-форвардинг, является опциональной возможностью IP-стека Linux. По умолчанию функция форвардинга не активируется, и система не пересылает транзитные пакеты через свои интерфейсы, а только обрабатывает адресованные ей пакеты. Включение форвардинга IP-пакетов производится через параметр `net.ipv4.ip_forward` интерфейса `sysctl`. Если значение этого параметра равно 0, то форвардинг отключен, если же значение параметра не равно 0, форвардинг включен:

```
[dalth@viking dalth]$ sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0
[dalth@viking dalth]$
```

Кроме того, возможно разрешать или запрещать участие в форвардинге для каждого интерфейса индивидуально:

```
[dalth@viking dalth]$ sysctl -a | grep forward | grep v4
net.ipv4.conf.vmnet1.mc_forwarding = 0
net.ipv4.conf.vmnet1.forwarding = 0
net.ipv4.conf.eth0.mc_forwarding = 0
net.ipv4.conf.eth0.forwarding = 0
net.ipv4.conf.lo.mc_forwarding = 0
net.ipv4.conf.lo.forwarding = 0
net.ipv4.conf.default.mc_forwarding = 0
net.ipv4.conf.default.forwarding = 0
net.ipv4.conf.all.mc_forwarding = 0
net.ipv4.conf.all.forwarding = 0
net.ipv4.ip_forward = 0
[dalth@viking dalth]$
```

По умолчанию форвардинг включается и выключается для всех интерфейсов одновременно, но для отдельных интерфейсов возможно сменить флаг участия в форвардинге. Изменять параметры форвардинга может только системный администратор или пользователь, который имеет право записи в необходимые файлы интерфейса `sysctl`. Следующий листинг демонстрирует включение форвардинга через все интерфейсы путем вызова программы `sysctl`:

```
[root@viking dalth]# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
[root@viking dalth]# sysctl -a | grep forward | sort
net.ipv4.conf.all.forwarding = 1
net.ipv4.conf.all.mc_forwarding = 0
net.ipv4.conf.default.forwarding = 1
net.ipv4.conf.default.mc_forwarding = 0
net.ipv4.conf.eth0.forwarding = 1
net.ipv4.conf.eth0.mc_forwarding = 0
net.ipv4.conf.lo.forwarding = 1
net.ipv4.conf.lo.mc_forwarding = 0
net.ipv4.conf.vmnet1.forwarding = 1
net.ipv4.conf.vmnet1.mc_forwarding = 0
net.ipv4.ip_forward = 1
[root@viking dalth]#
```

В процессе маршрутизации для выбора интерфейса и следующего узла для доставки пакета (next hop) ядро использует *таблицу маршрутизации*. Эта таблица представляет список критериев, в соответствии с которыми выбирается следующий узел. В частности, в таблице маршрутизации фигурируют следующие условия: адрес сети получателя пакета, маска подсети получателя пакета, IP-адрес следующего узла, метрика маршрута и служебные поля (например, тип и возраст записи). Таблица маршрутизации используется не только в IP-форвардинге, но и даже при



простой отсылке IP-пакета для выбора интерфейса, через который будет производиться отсылка пакета.

Запись о сети с адресом 0.0.0.0 и маской подсети 0.0.0.0 называют маршрутом по умолчанию, или *default route*. Узел, чей адрес указан в поле *gateway* для маршрута по умолчанию, называют маршрутизатором по умолчанию, или *default gateway* или *default router*. В системе может быть произвольное количество маршрутов по умолчанию, но они должны быть как минимум с разными метриками. Для просмотра таблицы маршрутизации можно воспользоваться командой *route*. Эта команда позволяет оперировать с таблицей маршрутов, добавляя и удаляя из нее записи.

```
[root@inferno dalth]# route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use
Iface
10.80.1.113      0.0.0.0         255.255.255.255 UH    0      0      0 ppp0
127.0.0.0        0.0.0.0         255.0.0.0        U      0      0      0 lo
0.0.0.0          10.80.1.113    0.0.0.0          UG    0      0      0 ppp0
[root@inferno dalth]# route del default
[root@inferno dalth]# route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use
Iface
10.80.1.113      0.0.0.0         255.255.255.255 UH    0      0      0 ppp0
127.0.0.0        0.0.0.0         255.0.0.0        U      0      0      0 lo
[root@inferno dalth]# route add default dev ppp0
[root@inferno dalth]#
```

В данном выводе таблица упорядочена по маске подсети, что соответствует порядку ее просмотра ядром. Столбцы *Destination* и *Genmask* содержат адрес и маску сети получателя пакета, столбец *Metric* фактически указывает приоритет маршрута (маршрут с меньшей метрикой более приоритетен), поле *Gateway* указывает IP-адрес следующего узла для передачи пакета. Некоторые типы интерфейсов (в частности, интерфейсы типа точка-точка, или point-to-point) подразумевают, что на принимающем конце линии связи всегда находится не более одного узла, и поэтому в этой ситуации IP-адрес следующего узла можно не указывать. В данном случае мы видим, что в приведенном примере некоторые узлы доступны через интерфейс *ppp0* типа точка-точка. В частности, именно из-за этого свойства приведенная выше таблица оказывается эквивалентна следующей ниже. Жирным шрифтом помечена измененная строка, демонстрирующая “точечную” природу PPP-соединения:

```
[root@inferno dalth]# route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use
Iface
10.80.1.113      0.0.0.0         255.255.255.255 UH    0      0      0
ppp0
127.0.0.0        0.0.0.0         255.0.0.0        U      0      0      0 lo
0.0.0.0          0.0.0.0         0.0.0.0          U      0      0      0
ppp0
```

Специфика использования протокола PPP (обычно используемого при модемных соединениях) такова, что любой PPP-интерфейс является интерфейсом типа точка-точка, более того – PPP и расшифровывается как *Point-to-Point Protocol*. Также интерфейсами точка-точка являются интерфейсы *SLIP* (Serial Line IP) и практически все разновидности туннельных интерфейсов.

При деактивизации интерфейса из таблицы маршрутизации автоматически исключаются все маршруты, для которых в поле *Iface* был указан отключившийся интерфейс. Для некоторых типов интерфейсов при активизации в таблице маршрутизации также создаются служебные записи о маршрутах, которые нельзя удалить.

В большинстве случаев таблица маршрутизации имеет не слишком большой

размер, но в некоторых ситуациях (в частности, на шлюзовых машинах в больших сетях) таблица может иметь весьма значительный размер и изменяется не “вручную” с помощью команды `route`, а специальными программами – демонами поддержки протоколов динамической маршрутизации. С некоторыми упрощениями алгоритм работы этих демонов можно описать следующим образом: демон “слушает” приходящие пакеты для обслуживаемых протоколов динамической маршрутизации, и по получении (или неполучении) такого пакета отдает ядру команду на изменение таблицы маршрутизации.

Следует также заметить, что команда `route` в режиме вывода таблицы маршрутизации фактически просто фильтрует и форматирует данные, содержащиеся в специальном файле, называемом `/proc/net/route`, используемом для доступа к таблице маршрутизации, ведущейся ядром.

## **Фильтры пакетов**

Возможность инсталляции фильтров пакетов является очень интересной возможностью, предоставляемой стеком TCP/IP ядра Linux. Не углубляясь в детали просто отметим, что IP-стек Linux позволяет различным модулям установить “ловушки” (*hooks*) для пакетов. При этом каждый пакет, попадающий в такую ловушку, передается для обработки драйверу, установившему эту ловушку. Драйвер, в свою очередь, может проанализировать пакет, проделать какие-либо действия с пакетом, после чего вернуть код обработки, инструктируя таким образом ядро о том, что следует делать с пакетом дальше – вернуть ли отправителю сообщение об ошибке, прервать ли обработку и уничтожить пакет, либо продолжать обработку пакета обычным образом.

В настоящее время для фильтрации пакетов наиболее часто используются средства *iptables*. *iptables* – это название утилиты, которая позволяет настроить множество драйверов сетевой подсистемы NETFILTER ядра Linux, позволяющих осуществить анализ, произвести преобразование, или изменить обработку IP-пакетов. Основным объектом в *iptables* является цепочка правил (*chain*). Каждое правило в цепочке содержит набор условий совпадения (*condition matches*) и действие (*action*). Цепочки сгруппированы в таблицы (*tables*).

Действий есть две разновидности – прерывающие обработку пакета в цепочке, например действия DROP или ACCEPT, и не прерывающие обработку пакета цепочкой – например, LOG или MARK.

Цепочки используются для проверки пакетов – то есть пакет поочередно последовательно сравнивается с каждым из правил цепочки, и если он удовлетворяет всем условиям в правиле, к пакету применяется действие, указанное в этом правиле. Если действие является прерывающим, то на этом обработка пакета этой цепочкой заканчивается, если действие не прерывающее, то пакет продолжает проверяться этой же цепочкой.

Стандартные цепочки также содержат специальное неявное действие по умолчанию, называемое политикой цепочки (*chain policy*). Действие, указанное как политика цепочки, применяется ко всем пакетам, которые не попали ни под одно правило с “прерывающим” действием.

## **Стандартные таблицы и цепочки**

Подсистема пакетного фильтра содержит три таблицы, в каждой из которых содержатся несколько цепочек (наборов правил). Кроме того, администратор может создавать собственные цепочки правил. Ниже перечисляются стандартные таблицы и цепочки:

Таблица	Назначение	Цепочки	Назначение
mangle	Модификация пакетов	PREROUTING	Модификация всех пришедших пакетов
		INPUT	Модификация пакетов, пришедших на адрес компьютера
		FORWARD	Модификация пакетов, которые должны быть отмаршрутизированы (пересланы на другой хост)
		OUTPUT	Модификация пакетов, сгенерированных процессами данного хоста
		POSTROUTING	Модификация всех переданных пакетов
filter	Фильтрация пакетов – принятие решения об их дальнейшей обработке или отказе от обработки)	INPUT	Фильтрация адресованных этому компьютеру пакетов
		OUTPUT	Фильтрация сгенерированных этим компьютером пакетов
		FORWARD	Фильтрация маршрутизируемых (транзитных) пакетов
nat	Трансляция адресов	PREROUTING	Трансляция адресов всех принимаемых пакетов
		FORWARD	Трансляция адресов транзитных пакетов
		POSTROUTING	Трансляция адресов всех передаваемых пакетов

Таблица *nat* обладает “двойственным” действием, т.е. если вы включите преобразование для входящих пакетов, исходящие пакеты также будут модифицироваться, и наоборот. Таблицы *mangle* и *nat* изменяют пакеты, но *mangle* не ведет список изменений, т.е. является “однонаправленной” таблицей.

## Порядок применения стандартных таблиц и цепочек

Рассмотрим, какой путь проходит пакет в цепочках и таблицах *iptables*. Для входящих пакетов (адресованных компьютеру, на котором активизирована поддержка *iptables*) верна следующая последовательность применения цепочек:

1. `mangle.PREROUTING`
2. `nat.PREROUTING`
3. `mangle.INPUT`
4. `filter.INPUT`

Для пакетов, отправляемых с компьютера, реализуется следующая цепочка обработки:

```
1. filter.OUTPUT
2. mangle.OUTPUT
3. nat.POSTROUTING
4. mangle.POSTROUTING
```

Пакеты, являющиеся транзитными (маршрутизируемыми), т.е. не адресованные фильтрующему компьютеру и не сгенерированные фильтрующим компьютером, проходят по следующей последовательности цепочек и таблиц:

```
1. mangle.PREROUTING
2. nat.PREROUTING
3. mangle.FORWARD
4. filter.FORWARD
5. nat.FORWARD
6. nat.POSTROUTING
7. mangle.POSTROUTING
```

## Стандартные действия

Каждое правило в любой цепочке может ссылаться на одно из стандартных или дополнительных действий, либо на какую-либо пользовательскую цепочку. Основное различие между стандартными и дополнительными действиями в том, что стандартные действия могут указываться в правилах любых цепочек любых таблиц, а дополнительные действия можно указывать только для некоторых цепочек некоторых таблиц. Перечислим стандартные действия:

ACCEPT	Прервать проверку пакета цепочкой и перейти к следующей в порядке обработки пакета стандартной цепочке
DROP	Прервать обработку пакета, сам пакет удалить
RETURN	Прервать проверку пакета цепочкой и вернуться к проверке вышестоящей цепочкой, а если действие встретилось в одно из стандартных цепочек, поступить с пакетом так, как предписано в политике цепочки (chain policy)
QUEUE	Передать пакет некоторому процессу для дальнейшей обработки

Интересной особенностью также является то, что существуют так называемые target extensions, которые реализованы как модули и также могут использоваться для указания проводимого над пакетом действия. В частности, к таким действиям, например, относятся действия LOG – запротоколировать факт получения пакета, MASQUERADE – подменить IP-адрес отправителя пакета, MARK – пометить пакет и многие другие. Некоторые действия могут встречаться не во всех цепочках, а только в некоторых цепочках некоторых таблиц.

Еще одним специфическим действием можно назвать переход к пользовательской цепочке. При этом, если пакет в процессе обработки попадает под действие правила, у которого в качестве действия указан переход к другой цепочке, то пакет начинает проверяться ее правилами. Это часто используется, чтобы сходным образом обрабатывать некоторые виды пакетов.

## Условия отбора

Условия отбора делятся на две группы – стандартные условия, которые применимы ко всем пакетам, и расширенные условия, называемые также match extensions. Расширенные условия могут применяться не для всех пакетов, а только для некоторых из них – например, дополнительные условия для протокола UDP включают в себя адреса портов отправителя и получателя, а для ICMP – тип и код ICMP-сообщения.

## Примеры конфигураций iptables

Попробуем рассмотреть несколько простых примеров, достаточно часто используемых в реальных конфигурациях. Стоит заметить, что самостоятельная конфигурация пакетного фильтра требует некоторых (точнее, достаточно значительных) знаний сетевых протоколов, поскольку в конфигурации необходимо задавать множество критериев, которые сильно зависят от ситуации и от используемых сервисов.

**Пример 1:** простейшая конфигурация iptables для домашнего компьютера, подключенного к Internet. В этой конфигурации мы запретим все входящие соединения, а также все исходящие пакеты UDP кроме тех, которые необходимы для нормальной работы в Internet с использованием PPP-соединения. В этой конфигурации мы разрешаем передачу всех пакетов в рамках локальной машины, разрешаем исходящие TCP-пакеты, разрешаем входящие пакеты TCP для уже установленных соединений, и разрешаем передачу и прием UDP-пакетов для службы DNS и пакетов автоматической конфигурации соединения PPP (пакетов DHCP). Кроме того, следует разрешить прием управляющих пакетов ICMP и отправку запросов и прием ответов PING:

```
# iptables -P INPUT DROP
# iptables -A INPUT -j ACCEPT -i lo
# iptables -A INPUT -j ACCEPT -p tcp ! --syn
# iptables -A INPUT -j ACCEPT -p udp --source-port 53
# iptables -A INPUT -j ACCEPT -p udp --source-port 67 --destination-port 68
# iptables -A INPUT -j ACCEPT -p icmp --icmp-type destination-unreachable
# iptables -A INPUT -j ACCEPT -p icmp --icmp-type time-exceeded
# iptables -A INPUT -j ACCEPT -p icmp --icmp-type parameter-problem
# iptables -A INPUT -j ACCEPT -p icmp --icmp-type echo-reply
# iptables -P OUTPUT DROP
# iptables -A OUTPUT -j ACCEPT -p tcp
# iptables -A OUTPUT -j ACCEPT -p udp --destination-port 53
# iptables -A OUTPUT -j ACCEPT -p udp --destination-port 67 --source-port 68
# iptables -A OUTPUT -j ACCEPT -p icmp --icmp-type echo-request
```

**Пример 2:** то же самое, что в примере 1, но все “отбитые” пакеты протоколируются. Для того, чтобы добиться такого эффекта, нужно создать дополнительную цепочку, которая будет протолировать и удалять пакеты. Эту цепочку мы назовем KILLER – вполне обоснованно, не так ли? Кроме того, мы исправим политики стандартных цепочек так, чтобы запрещенные пакеты не удалялись, а “забрасывались” в созданную нами цепочку KILLER, а нашей основной цели (сначала протолировать, потом удалить пакет) можно добиться просто указав два действия – сначала LOG, затем DROP. Поскольку действие LOG не является прерывающим обработку, мы получим требуемый нам эффект:

```
# iptables -N KILLER
# iptables -A KILLER -j LOG
# iptables -A KILLER -j DROP
# iptables -P INPUT KILLER
# iptables -A INPUT -j ACCEPT -i lo
# iptables -A INPUT -j ACCEPT -p tcp ! --syn
# iptables -A INPUT -j ACCEPT -p udp --source-port 53
# iptables -A INPUT -j ACCEPT -p udp --source-port 67 --destination-port 68
# iptables -A INPUT -j ACCEPT -p icmp --icmp-type destination-unreachable
# iptables -A INPUT -j ACCEPT -p icmp --icmp-type time-exceeded
# iptables -A INPUT -j ACCEPT -p icmp --icmp-type parameter-problem
# iptables -A INPUT -j ACCEPT -p icmp --icmp-type echo-reply
# iptables -P OUTPUT KILLER
# iptables -A OUTPUT -j ACCEPT -p tcp
# iptables -A OUTPUT -j ACCEPT -p udp --destination-port 53
# iptables -A OUTPUT -j ACCEPT -p udp --destination-port 67 --source-port 68
# iptables -A OUTPUT -j ACCEPT -p icmp --icmp-type echo-request
```

**Пример 3:** маскарад пакетов. Маскарадом называют преобразование IP-адресов проходящих пакетов так, чтобы они выглядели как отправленные с системы-маршрутизатора, а не с какого-либо узла “за” маршрутизатором. Достигается это путем изменения IP-адреса (и, возможно, номера порта) в “транзитных” пакетах. Собственно преобразование задается путем указания действий SNAT – замена адреса отправителя, DNAT – замена адреса получателя, или MASQUERADE – функционально аналогично SNAT, но без указания конкретного IP-адреса (IP-адресом для замены назначается IP-адрес интерфейса через который уходит пакет, со всеми отсюда вытекающими – например, если интерфейс меняет IP-адрес или просто деактивируется все «маскированные» через него соединения сбрасываются). Предположим, что наш “внешний” интерфейс имеет адрес 193.267.14.6, а внутренняя сеть имеет адрес 192.168.0.0/24. Тогда для того, чтобы дать всем компьютерам нашей сети доступ по протоколу TCP “наружу”, мы должны подать примерно следующую команду:

```
# iptables -A POSTROUTING -t nat -j SNAT -o ppp0 \
> --to-source 193.267.14.6 -p tcp \
> --source 192.168.0.0/24 \
> --destination ! 192.168.0.0/24
```

Если у нас внешний адрес динамический, а не статический (мы работаем по dialup – соединению), то мы можем использовать динамический маскарад без привязки к внешнему адресу – ну или с использованием динамической привязки, кому как больше нравится:

```
# iptables -A POSTROUTING -t nat -j MASQUERADE -o ppp0 \
> --source 192.168.0.0/24 \
> --destination ! 192.168.0.0/24
```

Действие SNAT более эффективно, MASQUERADE проще в использовании, но обладает рядом существенных недостатков (не вдаваясь в подробности, просто заметим, что на системе с несколькими интерфейсами и сложной таблицей маршрутизации проблемы почти наверняка будут). Особое внимание нужно обратить на указание -o ppp0, то есть действие применяется ТОЛЬКО для пакетов, отправляемых через интерфейс ppp0. Еще вы можете увидеть, что мы указываем это правило только один раз, и обратного к нему правила не строим - об этом позаботится функция connection tracking (отслеживание состояния соединений), и обратная замена адресов в отправляемых в ответ на наши запросы пакетах будет произведена системой автоматически.

**Пример 4:** “проброс” пакетов во внутреннюю сеть. Обычно это используется, если мы хотим перебросить пакеты, пришедшие на адрес маршрутизатора, на какую-либо из машин внутренней сети (например, так можно предоставить доступ ко

внутреннему WWW-серверу). Достигается это использованием действия DNAT (destination NAT). В нашем случае мы перебрасываем все TCP-пакеты, пришедшие на интерфейс маршрутизатора ppp0 на порт 80, на порт 85 компьютера с адресом 192.168.0.6:

```
# iptables -A PREROUTING -t nat -j DNAT -i ppp0 \  
> --to-destination 192.168.0.6:85 -p tcp --destination-ports 80
```

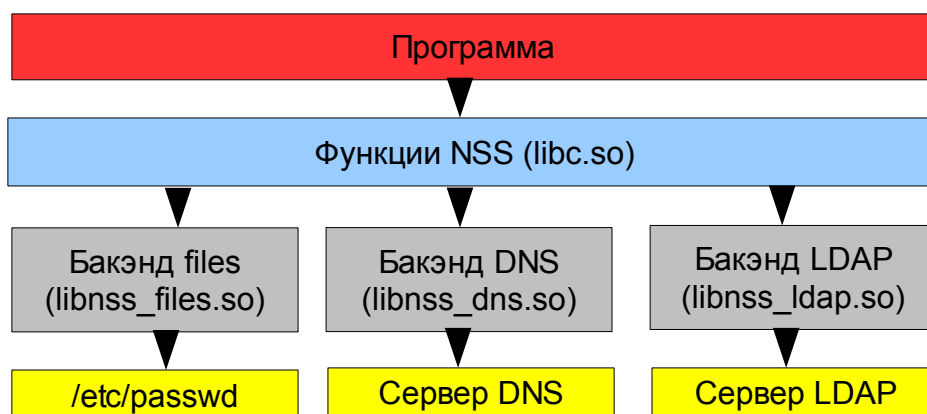
Конечно, приведенными примерами возможности iptables не исчерпываются, скорее даже наоборот — это лишь малая часть того, что умеет эта подсистема. Приведенные же примеры демонстрируют наиболее простые случаи, позволяющие составить некоторое представление об iptables и возможностях этой технологии.

В системах, основанных на RedHat Linux и его вариациях, есть специальный стартовый сценарий загрузки, также называемый iptables. Расположен он как правило в каталоге `/etc/rc.d/init.d/`. Этот сценарий при загрузке системы устанавливает правила, созданные администратором, и его можно использовать для сохранения конфигурации iptables. В процессе конфигурации системный администратор задает конфигурацию подсистемы iptables, используя утилиту `/sbin/iptables`, а после окончания настройки дает команду `/etc/rc.d/init.d/iptables save`, после чего текущая конфигурация сохраняется в файл `/etc/sysconfig/iptables`. Существует также множество “фронтендов” для настройки iptables, которые могут использоваться начинающими пользователями и не слишком опытными администраторами, но “ручной” способ настройки все-таки предпочтительней, поскольку позволяет очень точно настроить подсистему iptables.

## Управление пользователями, NSS и PAM

Linux — система многопользовательская. По умолчанию, большинство дистрибутивов используют «классический» набор файлов в которых хранится информация о пользователях и группах: `/etc/passwd`, `/etc/group`, `/etc/shadow`, `/etc/gshadow`. Во многих ситуациях этого вполне достаточно, но иногда возникает необходимость в интеграции Linux в более или менее чужеродное, либо просто распределенное окружение, и именно в этот момент к нам на помощь приходят такая интересная подсистема, как NSS — Name Service Switch. Основная задача NSS — создать модульное окружение для управления пользователями. Реализовано это посредством загружаемых библиотек. Основные вызовы NSS реализованы в библиотеке `libc`, а `libc` в свою очередь загружает и вызывает бэкэнды

NSS:



При инициализации программы, так или иначе связанной с NSS, загружаются основная библиотека `libc.so`, которая считывает конфигурацию из файла `/etc/nsswitch.conf`, после чего также загружаются те библиотеки NSS, которые указаны в этом файле.

Впоследствии при работе программы, если программе требуется работать с именованными сущностями, соответствующие вызовы функций `glibc` будут обращаться к функциям NSS и использовать в те источники данных, которые

указаны в nsswitch.conf.

В частности, через NSS можно разрешать (определять) имена и идентификаторы протоколов, номера портов служб (сервисов), имена и идентификаторы пользователей и групп, IP-адреса и имена компьютеров и некоторые другие данные.

Пример файла nsswitch.conf:

```
[viking@alpha etc]$ cat /etc/nsswitch.conf
passwd:      files ldap
shadow:      files ldap
group:       files ldap
hosts:       files dns
bootparams:  nisplus [NOTFOUND=return] files
ethers:      files
netmasks:   files
networks:    files
protocols:   files
rpc:         files
services:    files
netgroup:    nisplus
publickey:   nisplus
automount:   files nisplus
aliases:     files nisplus
[viking@alpha etc]$
```

В данном примере указано, что для определения имен пользователей и групп используются сначала текстовые файлы и затем LDAP, для определения имен компьютеров и IP-адресов используются сначала текстовые файлы и затем DNS, для определения алиасов и настроек автоматического монтирования каталогов используются сначала текстовые файлы и затем служба NIS+.

Библиотеки бэкэндов системы NSS хранятся в файлах libnss\_XXX.so, где XXX – это имя бэкэнда. Например libnss\_files.so – это бэкэнд NSS использующий в качестве источника данных текстовые файлы, libnss\_db.so – бэкэнд использующий файлы BerkeleyDB, libnss\_ldap.so – бэкэнд позволяющий хранить данные в каталоге LDAP. Как правило, каждый бэкэнд имеет свои дополнительные конфигурационные файлы. Как следствие, если у вас возникла необходимость использовать вашу Linux-систему в сетевом или чужеродном окружении и обеспечить ее интеграцию с ним, вы можете воспользоваться NSS и получить доступ к информации через соответствующий бэкэнд – например, для интеграции в среду Solaris, вы можете воспользоваться бэкэндом NIS/NIS+, для интеграции в ActiveDirectory – бэкэндом LDAP.

При этом модульность NSS позволяет вам объединять различные источники данных – например, использовать текстовые файлы и DNS для определения имен компьютеров, NIS для определения имен протоколов и сервисов, и текстовые файлы и LDAP для определения имен и идентификаторов пользователей и групп.

Подсистема PAM (Pluggable Authentication Modules) идейно очень схожа с NSS, но отличается от нее назначением. Основная задача PAM – аутентификация пользователей (проверка паролей, прав доступ, ограничений и так далее). Как и NSS, PAM состоит из набора основных библиотек и бэкэндов, причем необходимые бэкэнды, порядок их вызова и некоторые опциональные параметры определяются в конфигурационных файлах PAM, обычно они расположены в каталоге /etc/pam.d. Главным отличием PAM от NSS (кроме естественно назначения) является то, что PAM является не составной и неотъемлемой частью libc, а отдельным множеством библиотек.

Основная часть стандартных утилит UNIX для управления пользователями и группами и получения информации о них, в большинстве дистрибутивов Linux общего назначения, адаптирована и собрана с поддержкой NSS и PAM. К таким утилитам относятся passwd, chsh, chfn, id, who и другие. NSS также используется даже такими утилитами как ls, find, ps – то есть всеми теми программами, которые отображают имя пользователя. Соответственно, если программа запрашивает у



пользователя пароль – скорее всего она использует и NSS, и PAM (например XDM или GDM). Большинство программ в чьи функции входит обработка почты также используют NSS. Соответственно, можно уверенно говорить что подсистемы NSS и PAM и базовые знания об их предназначении на сегодняшний день являются необходимыми для администратора Linux-систем.

## **X11 и все-все-все**

Большинство нынешних дистрибутивов по умолчанию устанавливают для пользователя графическую среду X11 (X11 Windows System), под управлением которой и выполняются все графические приложения. Как «внутри» устроена X11? Прежде всего, X11 – это распределенная модульная среда, состоящая из двух основных компонентов: X-сервера и X-клиента.

### **Клиент-серверная архитектура X11**

**X-сервер** – это программа, которая организует работу с устройствами ввода/вывода, производит отрисовку видимых элементов, запущена у пользователя и предоставляет свои ресурсы (те же самые устройства ввода-вывода) для X-клиентов. X-сервер загружает драйверы устройств (например видеокарты, мыши или клавиатуры), он же управляет переключением раскладок клавиатуры и т.п. Кроме того, X-сервер частично берет на себя функции работы со шрифтами. Задача использования аппаратного ускорения для отрисовки также является прерогативой X-сервера.

В современных дистрибутивах как правило используется открытый свободно распространяемый X-сервер называемый Xorg. Его конфигурационный файл называется `xorg.conf` и расположен в каталоге `/etc/X11`. В конфигурационном файле описываются все устройства ввода, которые будет использовать X-сервер, настройки клавиатуры, драйвер видеокарты и многое другое. Более подробную информацию можно получить из справочного руководства [`man Xorg`, `man xorg.conf`].

**X-клиент** – это собственно пользовательская программа – браузер, почтовый клиент, видеоплеер, клиент мгновенных сообщений, игры, графические редакторы и просмотрщики и т.д.

Когда пользователь запускает графическое приложение, оно соединяется с X-сервером по стандартному протоколу X11, получает от X-сервера события о перемещении мыши, нажатиях кнопок клавиатуры и соответственно на них реагирует. Когда необходимо провести отрисовку, X-клиент отправляет соответствующие инструкции X-серверу, и уже X-сервер производит непосредственную отрисовку используя драйвер видеокарты. Команды протокола X11 могут передаваться как через разделяемую память или локальное соединение (если X-клиент и X-сервер запущены на одном компьютере), так и по сети – при этом X-клиент и X-сервер могут быть запущены на разных компьютерах.

X-сервер никаким образом не отвечает за внешний вид окна X-клиента – он отрисовывает все в точности так, как это указал клиент, более того – сам по себе X-сервер он не отрисовывает ни обрамлений окон, ни каких либо управляющих элементов, не обрабатывает никаких горячих клавиш – то есть он занимается только отрисовкой картинки присланной X-клиентом, а также чтением данных из устройств ввода и передачей соответствующих событий X-клиенту.

Обрамление окон, иконки рабочего стола, панели и кнопки – это все отрисовывается X-клиентами. Рассмотрим пример окна некоторого приложения (в нашем случае файлового менеджера Nautilus из состава среды GNOME):

На этой картинке на самом деле показан результат работы двух X-клиентов: собственно файлового менеджера `nautilus` (именно он управляет отрисовкой меню, строки статуса, панелей, иконок и так далее). Второй X-клиент – менеджер окон (`window manager`) под названием `metacity` нарисовал рамку окна и кнопки на этой

рамке. При этом горячие клавиши для закрытия окна, сворачивания, перемещения и других операций с окном обрабатывает именно metacity, а горячие клавиши копирования файла, перехода по каталогам, навигации по меню, подсвечивание иконок и тому подобное обрабатывает уже сам nautilus.

## Графическая среда пользователя

Графической средой мы будем называть набор программ, для пользователя для выполнения им повседневных функций. Каждая из этих программ как правило является самостоятельным X-клиентом и может работать сама по себе, даже без своих «коллег по окружению», но будучи собранными вместе, они начинают предоставлять пользователю цельный и органичный интерфейс. Достигается это обычно следующим образом:

- 1) Все программы данного графического окружения используют одну и ту же библиотеку для отрисовки своих элементов управления
- 2) В графическое окружение включается какой-либо менеджер окон
- 3) Все программы для данного окружения разрабатываются с соблюдением определенных общих требований
- 4) В состав окружения включаются самостоятельные программы для выполнения базовых функций - текстовые и табличные процессоры, графический редактор, браузер, IM-клиент и другие
- 5) В состав окружения включаются служебные программы, предоставляющие пользователю возможность вызова других программ – например, приложение которое отрисовывает панель, на которой размещается кнопка для вызова меню со списком установленных программ и список окон
- 6) В состав окружения включают утилиты для настройки оборудования и дополнительных функций

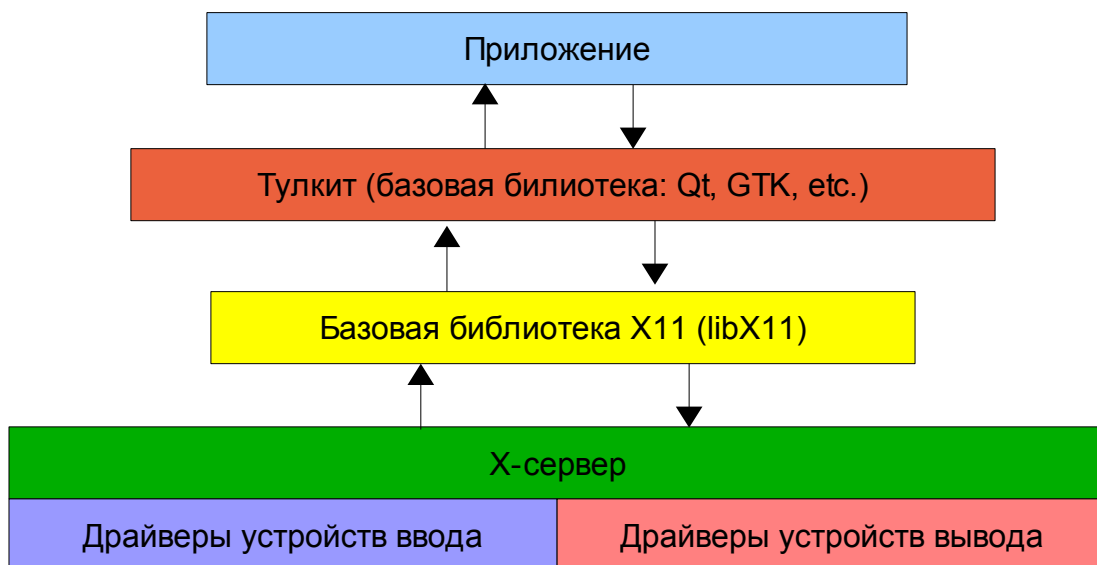
В результате внешний вид окружения становится единообразным, а тот факт что оно состоит из множества небольших программ делает графическую среду гибкой в настройке и стабильной в работе. В то же время обширный набор программ для различных функций делает позволяя графическому окружению удовлетворить большинство потребностей пользователя, а привычная вам картина GNOME или KDE, которую вы видите на экране в процессе работы, является результатом совместной работы множества X-клиентов соответствующего окружения... Или нескольких окружений одновременно.

Наиболее распространенные графические среды в Linux – это GNOME и KDE. И та и другая среда имеют свою библиотеку для отрисовки элементов управления внутри окна (для GNOME это GTK, для KDE это Qt), включают в свой состав почтовый клиент, браузер, мультимедиа-проигрыватель, клиент мгновенных сообщений и графический редактор, файловый менеджер (он же по совместительству отрисовывает иконки на рабочем столе и отвечает за фон рабочего стола), программу которая отрисовывает боковые панели и набор маленьких программ-апплетов, которые встраиваются при необходимости в панели, игры, программы управления для настройки окружения и менеджер окон.

Некоторые X-клиенты не входят в состав какого-либо окружения – например, браузеры Firefox и Opera, медиапроигрыватель Mplayer, офисный пакет OpenOffice.Org. Самое главное что следует запомнить – что графическое окружение, или графическая среда – это всего лишь набор программ, а не одна большая программа вида «все-в-одном», и вы можете, работая в основном в одном окружении, абсолютно спокойно использовать программы другого.

## Внутри X11. Упрощенная схема

X11 – система многослойная, но внутренне логичная и понятная. Схематически ее можно изобразить следующим образом:



Когда приложение отрисовывает что-либо, оно обращается к своей базовой библиотеке, она в свою очередь переадресовывает вызов (или уже цепочку вызовов) в библиотеку `libX11`, которая используя X11 Core Protocol передает команды на X-сервер. X-сервер интерпретирует команды и передает их на отрисовку драйверам устройств вывода. Воздействие же пользователя на устройства ввода считываются X-сервером через драйвера устройств ввода, через X11 Core Protocol передаются X-клиенту, `libX11` переводит команды протокола в события и передает события в тулkit, и уже тулkit передает события программе для того чтобы та на них отреагировала.

То как приложение выглядит – то есть внешний вид строк ввода, полос скроллинга, панелей и кнопок – за все это отвечает библиотека-тулkit. Два наиболее распространенных на сегодняшний день тулkit, Qt и GTK, примерно равны по возможностям – они обеспечивают переносимость приложений, поддержку «тем» внешнего вида, предоставляют программисту объектно-ориентированный интерфейс и некоторый набор служебных функций и возможностей.

Базовая библиотека X11 (`libX11`, или `Xlib`) позволяет программе абстрагироваться и не связываться с низкоуровневыми функциями протокола X11, и таким образом обеспечивает сетевую прозрачность X11, то есть позволяет программам работать как с локальным X-сервером, так и с находящимся на другом компьютере, причем делается это незаметно для программы.

## Внутри X11. Растровые шрифты и их отображение

С момента разработки в среде X11 отображение шрифтов управлял X-сервер. Приложение, когда ему необходимо вывести некоторый текст, просто инструктировало X-сервер «отобрази вот этот текст вот таким шрифтом в указанном месте». В ответ на это X-сервер выбирал из своей базы шрифтов наиболее подходящий, и использовал его для выполнения инструкций клиента, причем изначально эти шрифты были растровыми, то есть фактически содержали наборы заранее отрисованных глифов (символов).

Эта технология, называемая X Core Fonts, поддерживается в X11 и сейчас, поэтому в наборе пакетов любого дистрибутива всегда можно встретить наборы широко распространенных семейств растровых шрифтов – `fixed`, `hevetica`, `times`, `courier`. При этом каждый шрифт представляется множеством файлов, для различных сочетаний размера, ширины и начертания – например, если шрифт имеет десять вариантов размеров (от 8 до 18), две ширины (обычный и жирный) и два начертания (стандартное и курсив), то он будет представляться 40 файлами – по одному файлу для каждого из сочетаний размера, начертания и ширины.

Поскольку количество файлов получается очень большим, чтобы не устанавливать все эти шрифты на все компьютеры где запущены X-серверы, в систему X11 был введен такой объект как сервер шрифтов (font server). Системный администратор может настроить и запустить один сервер шрифтов для всей локальной сети, и указать X-серверу при запуске использовать шрифты с соответствующего сервера, что позволяет поддерживать на всех серверах один и тот же набор шрифтов с минимальными усилиями и избежать ситуации когда шрифт есть на одном компьютере, но его нет на другом.

В большинстве дистрибутивов Linux сервер шрифтов включен в поставку X11 по умолчанию, и называется xfs. Его конфигурационный файл как правило находится в каталоге /etc/X11/fs. В конфигурационном файле сервера шрифтов перечисляются каталоги с растровыми шрифтами, а в конфигурационном файле X-сервера указано, что основным источником шрифтов является сервер шрифтов, запущенный на этом же компьютере.

Такая методика позволяла добиться достаточно качественного отображения шрифта в большинстве случаев, но в то же время такие она имеет определенные недостатки – как известно, растровые шрифты очень плохо «вращаются» на угол не кратный 90 градусам, и плохо масштабируются, а с появлением устройств высокой четкости (LCD-мониторов) возникла необходимость еще и отрисовывать шрифты в «сглаженном» виде, с мягкими переходами цвета, чего растровые шрифты также предоставить не могли.

Наиболее распространенное приложение, использующее методику X Core Fonts – это графический эмулятор терминала xterm, который есть в составе всех дистрибутивов, или простейший менеджер графического входа в систему XDM.

## **Внутри X11. FreeType и XFT**

Для устранения этого недостатка растровых шрифтов была разработана специальные библиотеки XFT и FreeType, которые обеспечивают отображение векторных шрифтов и реализацию таких возможностей как сглаживание шрифтов, их поворот на произвольный угол и хинтинг (подгонку символов шрифта друг к другу оптимальным для данного шрифта способом). Поскольку данная методика плохо согласовывалась с уже сложившейся шрифтовой архитектурой X11, отображение векторных шрифтов было возложено на X-клиента.

При этом схема отображения шрифта меняется: X-клиент не передает на X-сервер запрос на вывод текста указанным шрифтом, а самостоятельно отрисовывает необходимые глифы шрифта используя функции FreeType и XFT, и передает на X-сервер уже сформированную картинку, которую X-сервер и отображает. Вполне логично, что при использовании этой методики X-клиент должен иметь доступ к оригиналу векторного шрифта (обычно это файл) – поэтому современные дистрибутивы также содержат еще и наборы векторных шрифтов PostScript и TrueType.

Новые версии тулкитов GTK и QT используют именно эту методику, но поскольку существуют еще и устаревшие тулкиты, такие как Motif или Xview (OpenLook) и значительное количество приложений на них основанных, да и с точки зрения необходимой полосы пропускания сети технология поддержки растровых шрифтов имеет явное преимущество перед XFT и FreeType, поддержка технологии растровых шрифтов по-прежнему входит в X11 и будет оставаться в ней еще долго.

## **Внутри X11. Расширения**

Протокол X11 не является замершим в своем развитии, но его развития ведется не путем изменения самого протокола, а путем внесения в протокол расширений – то есть дополнительных опциональных наборов команд и инструкций. Например, для поддержки непрямоугольных окон было введено расширение XShape, для поддержки проигрывания видеороликов введено расширение XVideo, для поддержки OpenGL было введено расширение GLX и так далее.

## Внутри X11. Полезные утилиты

В штатную поставку X11 входит множество полезных программ, позволяющих получать служебную информацию об ваших настройках среды X11. Перечислим некоторые из них:

**xterm** – эмулятор терминала

**xfontsel** – интерактивный просмотр растровых шрифтов

**xdpyinfo** – вывод информации о ваших настройках среды X11 и задействованных расширениях

**xwininfo** – просмотр информации об указанном окне (положение, размер, класс окна и т.п.)

**xwd** – моментальный «снимок» окна

**xwud** – показ результатов работы xwd

**xhost** – управление контролем доступа X-клиентов к X-серверу

## Внутри X11. Менеджер окон

Существует особая разновидность X-клиентов, называемых менеджерами окон. Их основная функция – обеспечивать управление другими окнами – перемещением, изменением размеров, сворачиванием и разворачиванием окон, отрисовкой обрамлений окон, управление передачей фокуса от окна к окну а также управление Z-порядком размещения окон. Соответственно, в один момент времени для одного экрана может быть использован только один менеджер окон.

GNOME и KDE имеют свои собственные менеджеры окон – это **metacity** и **kwin**. Менеджеры окон являются взаимозаменяемыми – то есть вы можете выбрать и использовать тот из них, который вам больше нравится или кажется более удобным. Еще один распространенный (то есть наличествующий почти на всех UNIX-системах) менеджер окон – это **twm**. Крайне простой и примитивный, он тем не менее предоставляет пользователю базовые функции управления окнами и доступ к меню приложений.

## Локализация

Широкое распространение UNIX-систем привело к необходимости введения системы локализации – поддержки предпочитаемых пользователем стандартов на запись даты и времени, формата чисел, языка сообщения программ, обозначений национальной валюты и так далее.

В Linux стандартом на методику локализации является спецификация i18n. Не вдаваясь в подробности, мы попытаемся получить некоторое представление о том как это выглядит для пользователя.

Как конечные пользователи, мы можем ознакомиться с текущими настройками своей локали используя команду locale:

```
$ locale
LANG=ru_RU.UTF-8
LC_CTYPE="ru_RU.UTF-8"
LC_NUMERIC="ru_RU.UTF-8"
LC_TIME="ru_RU.UTF-8"
LC_COLLATE="ru_RU.UTF-8"
LC_MONETARY="ru_RU.UTF-8"
LC_MESSAGES=en_US
LC_PAPER="ru_RU.UTF-8"
LC_NAME="ru_RU.UTF-8"
LC_ADDRESS="ru_RU.UTF-8"
LC_TELEPHONE="ru_RU.UTF-8"
LC_MEASUREMENT="ru_RU.UTF-8"
LC_IDENTIFICATION="ru_RU.UTF-8"
```

На самом деле, программа locale в корректно настроенной системе просто выведет

вам значение соответствующих переменных окружения. Каждая из этих переменных отвечает за свой собственный аспект локализации, перечислим наиболее важные:

**LANG** – локаль по умолчанию, ее значение для различных аспектов может быть перекрыто путем установки отдельных переменных в нужное вам значение

**LC\_CTYPE** – отвечает за классификацию символов и различия в их регистре

**LC\_NUMERIC** – отвечает за представление и форматирование чисел

**LC\_TIME** – отвечает за формат даты

**LC\_COLLATE** – определяет настройки сравнения строк и символов, влияет на сортировку

**LC\_MONETARY** – отвечает за представление национальной валюты

**LC\_MESSAGES** – определяет язык сообщений и интерфейса

В примере мы видим что локалью по умолчанию через переменную LANG выбрана локаль ru\_RU.UTF-8 (русский язык, соответствие российским стандартам, кодировка символов UTF-8), а в качестве языка интерфейса назначен английский язык в варианте используемом в США. Фактически это означает что все программы будут иметь английский интерфейс, но показывать дату, валюту, десятичную точку и сортировать строки они должны так как это принято в России.

В отличие от мира Windows, где язык сообщений программы как правило жестко забит в ресурсах, а ресурсы включены в исполняемый файл, в UNIX и Linux сообщения как правило содержатся в отдельных файлах и программа загружает текстовые строки для сообщений интерфейса основываясь на значениях переменных LANG и LC\_MESSAGES. Это дает возможность обеспечить работу пользователей из разных стран на привычных им языках, не прибегая к установке нескольких копий операционной системы или нескольких экземпляров программы, каждый для своего языка.

Системному администратору Linux достаточно обеспечить наличие файлов сообщений нужной локали для используемых пользователем программ, и правильно установить переменные окружения, после чего пользователь оказывается в привычной для него языковой обстановке и работать с привычными форматами даты, валюты, в привычных единицах измерения и т.д.

Более того – можно установить значение некоторых переменных окружения в нужные значения непосредственно перед запуском программы – и тогда эта программа примет будет использовать указанные настройки локали только для этого запуска, например:

```
[viking@alpha ~]$  
[viking@alpha ~]$ LC_MESSAGES=ru_RU.UTF-8 ls -l something  
ls: невозможно получить доступ к something: Нет такого файла или каталога  
[viking@alpha ~]$  
[viking@alpha ~]$ LC_MESSAGES=en_US.UTF-8 ls -l something  
ls: cannot access something: No such file or directory  
[viking@alpha ~]$  
[viking@alpha ~]$ LC_MESSAGES=fr_CA.UTF-8 ls -l something  
[viking@alpha ~]$  
ls: ne peut accéder something: Aucun fichier ou répertoire de ce type  
[viking@alpha ~]$
```

Таким образом, в Linux-системе возможно обеспечить одновременную работу пользователей в разных языковых окружениях, изменяя только изменение значений переменных окружения. В системах RedHat/Fedora общесистемные настройки локали хранятся в файле **/etc/sysconfig/i18n**, а пользователь может самостоятельно перекрыть их путем записи соответствующих значений в файле **~/i18n**:

```
[viking@alpha ~]$  
[viking@alpha ~]$ cat /etc/sysconfig/i18n  
LANG="ru_RU.UTF-8"
```

```
SYSFONT="latacyrheb-sun16"  
[viking@alpha ~]$ cat /homed/viking/.i18n  
LC_MESSAGES="en_US"  
[viking@alpha ~]$
```

В данном примере системная локаль **ru\_RU.UTF-8**, но пользователь viking предпочитает работать в англоязычном интерфейсе, что и записал в своем персональном файле настроек локали.

# Файловые системы Linux

## Создание файловых систем

**Ссылка на оригинал:** <http://linuxshop.ru/unix4all/?cid=&id=160>

**Автор:** Алексей Федорчук ([alv@linux-online.ru](mailto:alv@linux-online.ru))

**С версии:** 1.3

Создание файловых систем на дисковых разделах (или, в терминах DOS/Windows, форматирование последних) - второй этап подготовки диска к инсталляции Linux. Само по себе это действие - не из самых сложных, однако осознанное его выполнение требует некоторой подготовки.

### Необходимое введение

Термин "файловая система" - один из самых многозначных в компьютерной терминологии вообще, и в литературе, связанной с ОС Unix и ее клонами (а Linux именно к таковым и относится) - в особенности: больше значений имеет, пожалуй, только просто слово "система". Так сложилось исторически, и потому значение термина приходится определять по контексту. Мне таких значений известно как минимум пять, и нет уверенности, что это - все возможные значения.

Во-первых, файловой системой или, иногда, подсистемой, называют часть ядра, которая управляет организацией файлов и операциями над ними: в этом аспекте в одном ряду с файловой системой выступают система управления процессами и система ввода/вывода. Впрочем, именно это значение мы пока рассматривать не будем.

Во-вторых, и это специфично для Unix и Unix-подобных ОС, файловая система - универсальный интерфейс доступа ко всем ресурсам, как локальной машины, так и компьютеров, объединенных в сеть любого рода - от модемного подключения до спутникового канала. Собственно говоря, именно задействие такого интерфейса в некоей ОС и являет собой один из поводов для отнесения ее к семейству Unix-подобных.

Во-третьих, файловая система - это способ описания некоего физического устройства (обычно - дискового раздела). Именно это значение используется в термине "идентификатор типа файловой системы", о котором говорилось в статье [о дисковых разделах](#). Он обычно специфичен для конкретной ОС, и потому здесь уместны термины: DOS-раздел, расширенный (Extended) раздел DOS, раздел Linux native и так далее. Хотя большинство всамделишных операционных систем способны тем или иным образом опознавать "неродные" для них идентификаторы и обращаться к данным на них. А в Linux в качестве "родных" (native) рассматривает дисковые разделы, идентифицированные целым рядом файловых систем в третьем понимании термина - от истинно родного Linux native до Extended DOS, включая логические тома (LVM) о которых говорилось в [соответствующей статье](#), и программные RAID-массивы.

Здесь следует отметить, что помимо файловых систем физических (то есть



надстраивающих реальные дисковые устройства, почему в этом контексте нередко фигурирует выражение - disk based file system), существуют и виртуальные файловые системы различных типов. К ним относятся и файловая система устройств - devfs, и временная файловая система в оперативной памяти - tmpfs, и procfs - система, ответственная за представление в виде файловой системы (уж простите за тавтологию) процессов. Иногда используются и файловые системы промежуточного типа, например, виртуальные диски (RAM-диски). Подобно tmpfs, они существуют только в оперативной памяти, однако в остальном ведут себя также, как и файловые системы disk-based.

В четвертых, под файловыми системами понимается внутренняя управляющая структура, позволяющая хранить, идентифицировать и отыскивать данные, ну и, конечно, манипулировать ими. Такие структуры, с одной стороны, специфичны для операционных систем, как FAT16 (со всеми ее вариациями типа VFAT или FAT32) для DOS, UFS для FreeBSD или Ext2fs - для Linux. С другой же - структуры управления файлами в ряде операционных систем строятся по близким принципам, чему ярким примером служат файловые системы Unix- и Unix-подобных ОС. И потому они могут быть объединены в одно семейство, противопоставляемое FAT-семейству, например.

Кроме того, Linux в настоящее время способен работать с управляющими структурами различных типов - от Ext3fs, являющей собой надстройку над традиционной Ext2fs, до XFS и JFS, разработанных первоначально для версий Unix от SGI и IBM, соответственно, а также ReiserFS. Нет запрета и на размещение Linux'a на файловой системе типа FAT (хотя и резонов к тому - нет также).

Добавлю, что в списке из предыдущего абзаца перечислены только файловые системы, способные нести базовые компоненты Linux, отвечающие за ее запуск и минимальную функциональность. Что же касается обмена данными - таковой возможен из Linux'a практически со всеми известными файловыми системами, хотя с некоторыми из них (например, NTFS или HPFS) - только в режиме чтения.

Наконец, в пятых, файловая система в Unix - это и логическая структура каталогов и файлов, которая объединяет и физические, и виртуальные файловые системы самых различных типов (например, дисковые разделы с файловыми системами Ext2fs и FAT16, виртуальные procfs, devfs и tmpfs), причем не только на локальной машине, но и на любой удаленной. Структура эта - иерархическая, или древовидная, начинающаяся с корневого каталога, родительского по отношению ко всем прочим, от которого ответвляются отдельные файлы и дочерние каталоги, которые, в свою очередь, могут выступать как родительские по отношению к подкаталогам более глубоких уровней вложенности.

Положение дел в настоящий момент таково, что в Linux структура файловой системы обычно специфична для конкретного дистрибутива или их группы, связанной единством происхождения. Поэтому нередко можно столкнуться с такими выражениями, как файловая система Red Hat или Debian. Собственно, именно исторически сложившиеся различия в иерархии каталогов являются одним из критериев обособления нескольких линий дистрибутивов Linux. Как, впрочем, и потенциальной причиной их несовместимости. Однако можно надеяться, что усилиями стандартизирующих организаций, таких, как [Linux Standard Base](#) и [Filesystem Hierarchy Standard](#), русский перевод стандарта - на [сайте Виктора Костромина](#), увенчаются успехом, и можно будет говорить о единой логической файловой системе Linux, подобно тому, как это имеет место в ОС линии BSD.

В контексте настоящей статьи нас интересует только четвертый аспект файловых систем, то есть создание управляющих файлами структур, базируемых на дисках (вернее, их разделах).

## Устройство файловых систем Unix-семейства

В этом разделе будет говориться о предметах, общих для всех Unix. Все файлы в Unix физически состоят из двух частей, реально локализованных в различных блоках дискового накопителя, но обязательно находящихся в одном дисковом разделе, первичном или логическом.

Первая часть файла - его т.н. метаданные, которые содержат файловый дескриптор (это просто некое уникальное число), сведения о его атрибутах (принадлежности, правах доступа, времени модификации и т.д.), а также информацию о том, в каких блоках дискового раздела (которые так и называются - блоки данных) физически размещено содержимое файла - те самые последовательности байтов, которые образуют доступный пользователю ASCII-текст или исполняемый модуль программы.

Метаданные каждого файла записаны в специальной области диска, называемой суперблоком, где образуют т.н. inodes (от information nodes - информационные узлы). Каждому существующему файлу соответствует свой inode, и именно он однозначно идентифицируется файловым дескриптором. А сам по себе список inodes, соответствующих как существующим файлам, так и свободным блокам дискового раздела, и определяет границы файловой системы, то есть сколько файлов может быть в ней создано.

Так вот, суть процесса создания файловой системы на дисковом разделе (или, в понимании DOS/Windows, его форматирования) - в создании на нем суперблока (или, в некоторых файловых системах, нескольких его копий), списка inodes и отведении дискового пространства под блоки данных (а также загрузочного блока, о котором будет сказано ниже), а устройством этих дисковых областей определяются различия между файловыми системами различных типов. В результате на новом разделе образуется единственный файл - каталог корневого (для данной файловой системы) раздела (в некоторых случаях создается еще и каталог /lost+found, предназначенный для хранения нарушенных файлов).

Возникает вопрос, почему такой, казалось бы неотъемлемый, атрибут файла, как его имя, не обнаруживается ни в его метаданных, ни, тем более, среди его данных. Ответ прост: в Unix имя являет собой атрибут не файла, но файловой системы (в пятом, логическом, понимании термина). И для хранения имен файлов предназначены файлы особого типа - каталоги (в Unix есть и другие типы файлов, например, упомянутые выше файлы устройств). Они представляют собой просто списки файловых дескрипторов идентификаторов и соответствующих им имен файлов. Поэтому идущая от MacOS и активно используемая в Windows метафора каталога как папки с документами - в Unix только затемняет суть дела: здесь это скорее именно каталожный ящик в библиотеке.

Не смотря на столь простое устройство, роль каталогов в файловой системе Unix трудно переоценить: имена файлов, через которые они включаются в файловую систему (и через которые пользователь получает доступ к их содержимому), фигурируют только в составе каталога, к которому файл приписан - и больше нигде в системе. Так что удаление имени файла (или подкаталога) из списка, представляющего собой данные его родительского каталога (который, конечно, также имеет свой inode и файловый дескриптор, приписанный к каталогу, расположенному уровнем выше в иерархии файловой системы, и так далее) равносильно тому, что метаданные файла становятся недоступными, а приписанные к его inode блоки данных помечаются как свободные. Именно так осуществляется

удаление файла командой `rm` или файловым менеджером типа `Midnight Commander`.

Нас, однако, сейчас интересует прямо противоположное - сделать файловую систему доступной. Из сказанного понятно, что для этого она со всем ее содержимым (суперблоком, списком `inode`, блоками данных) должна быть включена в состав какого-либо из существующих каталогов, называемого точкой монтирования. Именно это и составляет суть процесса монтирования. Результат же для монтируемой файловой системы - в том, что ее корневой каталог (до сих пор безымянный) получает имя каталога - точки монтирования (`mount point`), содержимое которого отныне составляет список имен ее файлов и подкаталогов. Обратный процесс - размонтирование, следствием чего является отсоединение от точки монтирования дерева смонтированной файловой системы. Кроме того, в `inode` ее корневого каталога устанавливается т.н. бит чистого размонтирования (`clean bit`). Впрочем, вопросам монтирования и размонтирования файловых систем будет посвящена специальная статья. Пока же рассмотрим особенности файловых систем, используемых в Linux'e.

## **Файловые системы Linux**

### **Ext2fs**

До недавнего времени список истинно родных (`native`) файловых систем для Linux ограничивался единственной - `ext2fs` (правда, Linux способен загрузиться и работать с `FAT`-раздела, но об этом мне даже не хочется говорить). Название это расшифровывается как "вторая расширенная файловая система"; "расширенная" она - по сравнению с файловой системой ОС `minix`, послужившей прототипом Linux, "вторая" - потому что ранние версии Linux базировались на `Extfs` с более ограниченными возможностями.

О файловой системе `Ext2fs` написано немало (см. дополнительные источники). Поэтому замечу только, что по способу организации хранения данных она - типичная представительница файловых систем Unix. Отличительная ее особенность - наличие нескольких копий суперблока, что повышает надежность хранения данных. Кроме того, для характерен очень эффективный механизм кэширования дисковых операций, что обеспечивает замечательное их быстродействие - едва ли не рекордное среди известных мне файловых систем. Обратная сторона чего, однако, - относительно слабая устойчивость при аварийном завершении работы (вследствие мертвого зависания или отказа питания), поскольку отложенность записи изменений файлов делает весьма высокой вероятность нарушения связи между их `inodes` и блоками данных.

Конечно, времена, когда некорректный останов Linux-машины грозил полным разрушением файловой системы, остались в далеком прошлом. Однако в любом случае останов системы без штатного размонтирования файловых систем приводит к тому, что в них не устанавливается упомянутый выше "бит чистого размонтирования". А без этого утилиты обслуживания диска (такие, как программа проверки `fsck`) при перезагрузке не воспринимают их как целостные и начинают проверку, которая при современных объемах дисков может занять немалое время.

### **О журналируемых файловых системах**

Проблема нарушения целостности файловой системы при некорректном завершении работы в большей или меньшей мере характерна для всех ОС семейства Unix. И потому с давних пор в них разрабатываются т.н. журналируемые

файловые системы. Журнал -- это нечто вроде log-файла дисковых операций, в котором фиксируются не выполненные, а только предстоящие манипуляции с файлами, вследствие чего оказывается возможным самовосстановление целостности файловой системы после сбоя.

Во избежание недоразумений следует подчеркнуть, что журналирование направленно на обеспечение целостности файловой системы, но ни в коем случае не гарантирует сохранность пользовательских данных как таковых. Так, не следует ожидать, что журналирование волшебным образом восстановит не сохраненные перед сбоем изменения документа, загруженного в текстовый редактор.

Более того, в большинстве журналируемых файловых систем фиксируются грядущие операции только над метаданными изменяемых файлов. Обычно этого достаточно для сохранения целостности файловой системы и, уж во всяком случае, предотвращения долговременных их проверок, однако не предотвращает потери данных в аварийных ситуациях. В некоторых из файловых систем возможно распространение журналирования и на область данных файла. Однако, как всегда, повышение надежности за счет этого оплачивается снижением быстродействия.

Текущие версии ядра Linux поддерживают в качестве нативных четыре журналируемые файловые системы: ReiserFS и Ext3fs, специфичные для этой ОС, XFS и JFS - результаты портирования в Linux файловых систем, разработанных первоначально для рабочих станций под ОС Irix (SGI) и AIX (IBM), соответственно. Правда, широкое признание получили только три первых, так что о JFS я пока говорить не буду.

## ReiserFS

Файловая система ReiserFS оказалась для Linux исторически первой из журналируемых - она поддерживается каноническим ядром с <http://www.kernel.org>, начиная с первых версий ветви 2.4.x (в настоящее время существуют патчи, позволяющие использовать ее и с версиями ветви 2.2.xx). И была единственной, разработанной "с нуля" специально для этой ОС Хансом Райзером и его фирмой [Namesys](http://www.namesys.com). Как и в большинстве рассмотренных, в ReiserFS осуществляется журналирование только операций над метаданными файлов. Что, при определенном снижении надежности, обеспечивает высокую производительность: по моим наблюдениям, на большинстве типичных пользовательских задач она лишь незначительно уступает Ext2fs. А на такой, достаточно обычной, операции, как копировании большого количества мелких файлов, существенно ее опережает.

Кроме этого, ReiserFS обладает уникальной (и по умолчанию задействованной) возможностью оптимизации дискового пространства, занимаемого мелкими, менее одного блока, файлами (а следует помнить, что в любой Unix-системе такие файлы присутствуют в изобилии): они целиком хранятся в своих inode, без выделения блоков в области данных - вместе с экономией места это способствует и росту производительности, так как и данные, и метаданные (в терминах ReiserFS - stat-data) файла хранятся в непосредственной близости и могут быть считаны одной операцией ввода/вывода.

Вторая особенность ReiserFS - то, что т.н. хвосты файлов, то есть их конечные части, меньшие по размеру, чем один блок, могут быть подвергнуты упаковке. Этот режим (tailing) также включается по умолчанию при создании ReiserFS, обеспечивая около 5% экономии дискового пространства. Что, правда, несколько снижает быстродействие, и потому режим тайлинга можно отменить при монтировании файловой системы. Однако упаковка хвостов автоматически восстанавливается после перекомпиляции ядра - что, как будет сказано чуть ниже, требует внимательного отношения.

ReiserFS не совместима с Ext2fs на уровне утилит обслуживания файловой системы. Однако соответствующий инструментарий, объединенный в пакет reiserfsprogs, уже давно включается в штатный комплект современных дистрибутивов (или, в крайнем случае, может быть получен с сайта [Namesys](http://Namesys.com)).

Более серьезная проблема с совместимостью - в том, что распространенные загрузчики Linux (и Lilo, и GRUB - хотя и по разным причинам) часто не способны загрузить ядро Linux с раздела ReiserFS, оптимизированного в режиме тайлинга. А поскольку, будучи отключен, этот режим обладает свойством самовосстановления, пользователь может столкнуться с тем, что после пересборки ядра система просто откажется загружаться. Именно поэтому выше я упоминал, что создание раздела под каталог /boot может быть необходимым.

## Ext3fs

В отличие от ReiserFS, Ext3fs - не более чем журналируемая надстройка над классической Ext2fs, разработанная Стивеном Твиди в компании Red Hat и поддерживаемая ядром Linux, начиная с версии 2.4.16. Как следствие такого происхождения, она сохраняет со своей прародительницей полную совместимость, в том числе и на уровне утилит обслуживания (начиная с версии 1.21 объединяющего их пакета e2fsprogs). И переход от ext2fs к ext3fs может быть осуществлен простым добавлением файла журнала к первой, не только без переформатирования раздела, но даже и без рестарта машины.

Из этого вытекает первое преимущество ext3fs, особенно весомое в случае большого парка компьютеров. Второе же - чуть ли не максимальная надежность: ext3fs является единственной системой из рассматриваемых, в которой возможно журналирование операций не только с метаданными, но и с данными файлов.

В Ext3fs предусмотрено три режима работы - полное журналирование (full data journaling), журналирование с обратной записью (writeback), а также задействуемое по умолчанию последовательное (ordered).

Режим полного журналирования, как легко догадаться из названия, распространяется и на метаданные, и на данные файлов. Все их изменения сначала пишутся в файл журнала и только после этого фиксируются на диске. В случае аварийного отказа журнал можно повторно перечитать, приведя данные и метаданные в непротиворечивое состояние. Этот механизм практически гарантирует от потерь данных, однако является наиболее медленным.

В режиме отложенной записи, напротив, в файл журнала записываются только изменения метаданных файлов, подобно всем рассмотренным ниже файловым системам. То есть никакой гарантии сохранности данных он не предоставляет, однако обеспечивает наибольшее (в рамках Ext3fs) быстродействие.

В последовательном режиме также физически журналируются только метаданные файлов, однако связанные с ними блоки данных логически группируются в единый модуль, называемый транзакцией (transaction). И эти блоки записываются перед записью на диск новых метаданных на диск, что, хотя и не гарантирует сохранности данных, весьма последней способствует. Причем - при меньших накладных расходах по сравнению с полным журналированием, обеспечивая промежуточный уровень быстродействия.

Однако, по моим наблюдениям Ext3fs ни в одном из режимов не может похвастаться высокой производительностью при решении настоящих пользовательских задач. А в режиме полного журналирования возможны ситуации, вызывающие на некоторое время полный паралич машины.

## XFS

Файловая система XFS, в отличие от молодых ReiserFS и ext3fs, развивается для фирмой SGI на протяжении почти десяти лет - впервые она появилась в версии Irix 5.3, вышедшей в 1994 г. Но в Linux она была портирована лишь недавно (текущая ее версия - 1.1, свободно доступна с сайта SGI's XFS page - <http://oss.sgi.com/projects/xfs>) и по сию пору не поддерживается официальным ядром.

XFS - единственная 64-разрядная файловая система из рассмотренных. Однако уникальность ее - не только в этом. Особенности XFS являются:

- механизм allocation group, то есть деление единого дискового раздела на несколько равных областей, имеющих собственные списки inodes и свободных блоков, для распараллеливания дисковых операций;
- логическое журналирование только изменений метаданных, но - с частым сбросом их на диск для минимизации возможных потерь при сбоях;
- механизм delayed allocation - ассигнование дискового пространства при записи файлов не во время журналирования, а при фактическом сбросе их на диск, что, вместе с повышением производительности, предотвращает фрагментацию дискового раздела;
- списки контроля доступа (ACL, Access Control List) и расширенные атрибуты файлов (extended attributes), рассмотрение которых далеко выходит за рамки нынешней темы.

В результате XFS предстает как очень сбалансированная файловая система: она почти столь же надежна, как Ext3fs, и не очень уступает ReiserFS в быстродействии на большинстве файловых операций. А при манипуляциях с (очень) большими файлами XFS - просто вне конкуренции: как легко догадаться по имени фирмы-разработчика, она ориентирована на работу с мультимедийными приложениями с их огромными потоками данных. Не отмечалось для нее и проблем с совместимостью.

Все сказанное позволяет сделать вывод, что XFS - оптимальная файловая система для Linux. Однако следует учесть: в отличие от ReiserFS и ext2fs, поддержка которых является штатными опциями ядра Linux, XFS по сию пору (текущая версия - 2.4.19) не поддерживается каноническим ядром Линуса Торвальдса (тем, которое можно получить с <http://www.kernel.org>). Хотя недавнее включение такой поддержки в разрабатываемую ветвь ядра (версии 2.5.X) позволяют надеяться, что скоро эта функция станет штатной.

Возможность работы с XFS обеспечивает специальный патч (xfs-2.4.1X-all-i386.bz2), который можно получить с сайта [SGI](http://www.sgi.com) вместе с соответствующими утилитами поддержки: традиционные средства e2fsprogs, для XFS не пригодны. Утилиты поддержки для XFS объединены в несколько пакетов, из которых абсолютно необходимым является xfsprogs. Обо всем этом следует помнить при предварительной разметке диска.

## Критерии выбора

Таким образом, каждая из четырех рассмотренных файловых систем имеет свою уникальную положительную особенность (даже Ext2fs - как бы то ни было, лидером по суммарному быстродействию остается она), и как минимум один недостаток (который, тем не менее, не служит препятствием к ее использованию). Так что выбор файловой системы должен определяться задачами пользователя и характером преобладающих его данных.



Кроме этого, если, как рассказывалось в статье про [LVM](#), максимально разгрузить корневую файловую систему, выделив в логические тома отдельные ее ветви, возможно и комбинированное использование файловых систем. То есть подбор для каждого из дисковых разделов наиболее подходящей к его содержанию управляющей структуры.

Так, ранее неоднократно говорилось, что Ext2fs - наиболее подходящий выбор для загрузочного раздела (а при использовании в качестве загрузчика GRUB - это почти обязательное требование). Кроме того, Ext2fs вполне подходит для таких ветвей, как /tmp или /var. Для первого, по определению, устойчивость к сбоям не критична. Для второго же определяющим требованием является быстродействие (так, в Source Based дистрибутивах типа Gentoo /var задействуется для хранения временных продуктов компиляции, и быстродействие файловых операций в нем несколько способствует ускорению сборки пакетов). Наконец, на настольной машине Ext2fs можно применить и для корневой файловой системы - ведь при дробном разбиении диска в корне остается минимум редко изменяемых компонентов.

С другой стороны, корень - наиболее критичный в отношении устойчивости элемент файловой системы. И потому оптимальным для него представляется файловая система Ext3fs, как наиболее устоявшаяся. Кроме того, в экстремальных ситуациях она может быть без проблем смонтирована как Ext2fs. Для разделов типа /usr и /usr/local Ext3fs также видится вполне подходящим вариантом.

Наиболее важная часть файловой системы настольной машины с точки зрения пользователя - его, пользовательские, данные, то есть каталог /home (ибо систему можно переустановить, а вот потеря данных может быть невосполнимой). Однако это - и наиболее изменяемая ее часть, что предъявляет высокие требования к быстродействию файловых операций. И поэтому Ext3fs - не лучшее (ИМХО) решение для каталога /home, более целесообразно разместить здесь какую-либо из "быстрых" журналируемых файловых систем, ReiserFS или XFS. Выбор между ними определяется личными предпочтениями и характером данных (пользуясь случаем, замечу, что быстродействие JFS, по моим наблюдениям над типичными пользовательскими манипуляциями, оставляет желать лучшего).

Очевидно, что быстродействие XFS при работе с файлами (очень) большого размера делает ее предпочтительной, если речь идет об обработке изображений, мультимедийном контенте, картографической информации и т.д. В то же время преимущества ReiserFS сказываются в основном при работе с файлами (очень) маленького размера (менее блока файловой системы), каковых среди пользовательских данных обычно бывает мало. И потому мое личное мнение однозначно - и высказывается в пользу XFS. К тому же собственный опыт общения с ReiserFS был неблагоприятен, особенно в сочетании с технологией LVM. XFS же, напротив, по моим впечатлениям идеально с ней гармонирует.

Подведем итог - оптимальной мне видится следующее сочетание файловых систем:

- Ext3fs - для корневого каталога (/) и каталога /usr (а также /usr/local и /usr/X11R6, если таковые обособляются в отдельные ветви);
- Ext2fs - для загрузочного /boot, каталогов /tmp и /var;
- XFS - для раздела под домашние каталоги (/home).

Повторяю, это лишь мое мнение, основанное на опыте настольного применения Linux - для серверов различного рода оно силы не имеет.

## **Практические следствия**

Надеюсь, предыдущий раздел дал достаточно информации для выбора файловой системы, наиболее подходящей к конкретным условиям. И потому далее вкратце

рассмотрим последовательность действий после того, как выбор осуществлен.

Обычно при установке Linux создание файловых систем - компетенция инсталлятора, который осуществляет его с некоторыми опциями по умолчанию. В контексте рассмотренного в предыдущем разделе ясно, что опции эти не всегда будут лучшими с точки зрения пользователя. Изменить же характеристики, определенные для файловой системы при ее создании, невозможно без повторного выполнения этого процесса (и, соответственно, потери данных). Однако все SB-дистрибутивы допускают ручное вмешательство в процедуру создания файловой системы, а в Gentoo, например, это - просто единственный способ ее выполнения.

Файловая система Ext2fs может быть создана любой из следующих команд - /sbin/mke2fs, /sbin/mkfs, /sbin/mkfs.ext2 с указанием файла устройства в качестве аргумента, например:

```
$ /sbin/mke2fs /dev/hd?#
```

Каждая из этих команд (а /sbin/mkfs.ext2 - обычно символическая ссылка на /sbin/mke2fs) имеет ряд опций, как то: -b для определения размера блока файловой системы (возможные значения - 1024, 2048 или 4096 байт, по умолчанию принято последнее), -c для проверки испорченных участков диска, -N и -i для задания числа inodes и количества байт на один узел, соответственно; с деталями можно ознакомиться на соответствующей man-странице, например, man (8) mke2fs.

Для создания файловой системы ext3fs можно применить ту же команду mke2fs с опцией -j, при этом случае она получит некоторые "умолчальные" характеристики. Определить же их вручную позволяет следующая форма этой команды:

```
$ /sbin/mke2fs -J опции_журналирования /dev/hd?#
```

Возможные значения опций журналирования - size=размер, задающая объем журнального файла в мегабайтах, и device=внешний\_журнал подключения новой файловой системы к журналу, ранее созданному на другом дисковом разделе.

Можно использовать и специальную команду /sbin/mkfs.ext3 - возможности ее идентичны таковым /sbin/mke2fs (ибо она ни что иное, как символическая на нее ссылка). Но самое интересное - возможность преобразования существующей ext2fs в ext3fs простым добавлением журнала, не только без потери данных, но и без перезапуска системы (и даже без размонтирования). Делается это командой

```
$ tune2fs -j /dev/hd?#
```

Она просто добавляет файл журнала /journal в корневом каталоге модифицируемой файловой системы (если последняя не была размонтирована), или задействует для журнала скрытый inode (если перед модификацией файловая система была размонтирована). Добавлю, что обратное преобразование - еще проще, и осуществляется командой монтирования (о чем будет говориться в следующей статье).

Файловая система ReiserFS создается специально предназначенной для этого командой - /sbin/mkreiserfs из пакета reiserfsprogs. Для нее доступны многочисленные опции (-s для задания размера журнала, -f для принудительного переформатирования ранее существовавшей файловой системы иного типа, и т.д.), с которыми можно ознакомиться посредством man (8) mkreiserfs. И во избежание неожиданностей напомним: если корневой раздел форматируется как ReiserFS, не лишним будет предусмотреть небольшой раздел под каталог /boot для размещения на нем файловой системы ext2fs.

Для создания XFS также существует собственная команда mkfs.xfs (из пакета



xfsprogs). В ней предусмотрено несколько опций, каждая из которых имеет ряд субопций, принимающих численные значения. Важнейшие из них:

- -b, которая посредством субопции size=## позволяет задать размер блока данных в байтах, который должен быть кратен размеру страницы оперативной памяти (для платформы i386 - 4 Кбайт) и может варьировать в диапазоне от 512 до 65536 (по умолчанию - 4096);
- -d, определяющая параметры области данных файловой системы, такие, как количество самостоятельных областей раздела (Allocation groups, субопция agcount), или, напротив, их размер (субопция agsize);
- -l, специфицирующая параметры журнального файла, например, его размер (субопция size).

При использовании mkfs.xfs для достижения максимальной производительности рекомендуется в явном виде задать количество allocation groups - иначе оно будет определяться автоматически, что ведет к непроизводительным расходам ресурсов. Это делается из расчета - одна allocation group на 4 Гбайт дискового пространства. Далее можно установить размер файла журнала - здесь рекомендованное значение составляет 32 Мбайт. То есть для дискового раздела объемом в 20 Гбайт команда приобретет вид

```
$ mkfs.xfs -d agcount=5 -l size=32m /dev/hda1
```

Кроме всех перечисленных, команда mkfs.xfs имеет опцию -f (от force) - принудительное создание файловой системы XFS поверх любой существующей. Ее достаточно, если последняя была ext2fs (и, исходя из общих соображений, ext3fs, хотя я этого не проверял). Если же XFS создается поверх ReiserFS - после этого возможны ошибки при монтировании новой файловой системы. Впрочем, то же относится и к обратной процедуре (замене XFS на ReiserFS), а также, если любая из этих "продвинутых" файловых систем заменяется на разделе системой ext2fs. Они связаны с тем, что команда монтирования может распознать новосозданную XFS как дефектную ReiserFS, и наоборот.

Во избежание этого перед таким замещением приходится прибегать к несколько шаманскому приему - обнулению начальных областей раздела (хранящего метаданные файловой системы) командой

```
$ dd if=/dev/zero of=/dev/hd?#
```

Ждать заполнения нулями всего устройства не обязательно - достаточно дать этой команде поработать секунд 10-20, после чего прервать ее комбинацией клавиш Control+D и перейти к созданию новых файловых систем.

И последнее, о чем следует сказать - о swap-разделе, созданном на этапе разбиения диска. Хотя файловой системы как таковой он не несет, но нуждается в определении, что достигается командой

```
$ mkswap имя_устройства
```

к которой следует подходить со вниманием - применение ее к обычному разделу уничтожит на нем все данные.

## **Библиография вопроса**

Многие сложные вопросы устройства файловых систем в этой статье были затронуты лишь вскользь. За более подробной информацией по ним следует обратиться к дополнительным источникам. Так, общая организация файловой системы Unix рассматривается во многих руководствах по этой операционной

системе, например: С. [Д. Кузнецов. Операционная система UNIX](#).

Устройство файловой системы ext2fs подробно описано в статье Виктора Хименко "Файлы, файлы, файлы" (Мир ПК, 2000, [часть 1 - #2](#); [часть 2 - #3](#)).

Подробное описание современных журналируемых файловых систем, используемых в Linux, дано в цикле статей Дэниела Роббинса, русский перевод которого, выполненный Владимиром Холмановым, доступен на сайте [Ярославской группы пользователей Linux](#).

## Linux и файловые системы: еще раз о проблеме выбора

**Ссылка на оригинал:** <http://citkit.ru/articles/381/>

**Автор:** Алексей Федорчук ([alv@linux-online.ru](mailto:alv@linux-online.ru))

**Дата:** 15.03.2007 г.

**С версии:** 1.3

Весна наступила... Индо взопрели озимые, солнышко рассупонилось. Старик Ромуальдыч начал к своей портянке принюхиваться - оттаяла, видать. Податтаили и пользователи - на форумы потянулись, вопросы задавать, да ответы получать...

Одним из популярных вопросов в нынешнем сезоне оказался такой: как разбить диск и какую файловую систему выбрать.

Общий ответ на первую половину вопроса дать невозможно, а варианты решения этой проблемы обсуждались столько раз, что повторять их было бы скучно. А потому перейдем сразу ко второй половине - о дисковой разметке я вскользь скажу в самом конце.

В качестве "родных" для Linux (то есть тех, на которые он может быть установлен и с которых способен стартовать) рассматриваются следующие файловые системы: ext2fs, ext3fs, ReiserFS, XFS, JFS. Именно они обычно и предлагаются на выбор при установке подавляющего большинства дистрибутивов. Конечно, существуют и способы установки Linux на файловые системы FAT/VFAT/FAT32, но это - только для тех медам и мсье, которые понимают толк в извращениях, и о них я говорить не буду.

Исключу из рассмотрения также JFS - по следующим причинам:

- малой распространенности среди пользователей Linux;
- недостаточности источников информации по ней (именно в Linux - для родной ОС AIX эта файловая система прекрасно документирована);
- моим лично слабым с ней знакомством.

Основными критериями при выборе файловой системы являются обычно надежность и быстродействие. В некоторых случаях приходится учитывать также фактор совместимости - в данном случае под ней понимается способность других операционок обращаться к той или иной файловой системе.

Начну рассмотрение с ReiserFS - потому что поводом к сочинению этой заметки послужил вопрос: а что следует считать маленькими файлами? Ведь общеизвестно, что именно эффективность работы с мелкими файлами является сильной стороной этой файловой системы.

Так вот, под мелкими файлами понимаются файлы размером меньше логического

блока файловой системы, который в Linux в большинстве случаев равен четырем килобайтам, хотя и может задаваться при форматировании в некоторых пределах (зависящих от конкретной FS). Таких мелких файлов в любой Unix-подобной ОС - бесчисленное количество. Типичным примером являются файлы, составляющие дерево портов FreeBSD, портежей Gentoo и тому подобных портообразных систем.

В большинстве файловых систем для таких минифайлов существует как свой inode (информационный узел, содержащий метаинформацию о файле), так и блок данных, что приводит как к расходу дискового пространства, так и снижению быстродействия файловых операций. В частности, именно в этом причина катастрофической задумчивости файловой системы FreeBSD (как старой, UFS, так и новой, UFS2) при работе с собственной же системой портов.

В файловой системе ReiserFS в таких случаях отдельные блоки под данные не выделяются - она умудряется запихать данные файла непосредственно в область его же inode. За счет этого и дисковое пространство экономится, и быстродействие возрастает - буквально в несколько раз по сравнению со всеми прочими FS.

Такое обращение с мелкими файлами ReiserFS послужило причиной возникновения легенды о ее ненадежности. Действительно, при крахе файловой системы (то есть разрушении служебных областей) данные, размещенные совместно со своими inodes, вместе с ними же и пропадают - причем безвозвратно. Тогда как в тех файловых системах, где inodes и блоки данных всегда разобщены пространственно, последние теоретически можно восстановить. Так, для ext2/ext3 даже существуют средства, позволяющие это сделать.

Однако, как и всякая легенда, эта лишь производит впечатление достоверности. Во-первых, безвозвратная потеря данных относится лишь к очень маленьким файлам. Среди пользовательских таковых практически не бывает, а все прочие же легко восстанавливаются из дистрибутива.

Во-вторых, говоря о возможности восстановления данных из блоков, утративших привязку к своим inodes, я не случайно употребил слово "теоретическая". Потому что на практике это занятие чрезвычайно трудоемкое, не дающее гарантированного результата. Каждый, кому приходилось этим заниматься, согласится, что предаться ему можно только от полной безысходности. И это относится ко всем файловым системам Linux. Так что этим аспектом при выборе файловой системы можно пренебречь.

По суммарному быстродействию ReiserFS однозначно быстрее всех остальных журналируемых FS, а по некоторым показателям превосходит и ext2. С результатами сравнения скорости выполнения некоторых распространенных файловых операций можно ознакомиться [здесь](#).

А вот с совместимостью у ReiserFS дело обстоит несколько хуже. Доступ к ней из ОС семейства Windows, насколько мне известно, невозможен. В некоторых операционках семейства BSD (DragonFlyBSD, FreeBSD) реализована поддержка этой файловой системы, но в режиме только для чтения. Даже вероятность того, что произвольный Linux LiveCD прошлых лет не имеет поддержки ReiserFS, не нулевая.

И тут впору вспомнить об ext3fs. Преимущество ее вовсе не в большей надежности - это такая же легенда, как и неустойчивость ReiserFS. О случаях краха ext3fs я слышал не меньше, чем об аналогичных происшествиях с ReiserFS. Самому мне не удавалось порушить ни ту, ни другую. Разве что с ext2 получалось - но и то очень давно, во времена ядра 2.2 (или даже еще 2.0).

Нет, главное преимущество ext3fs в ее совместимости - она с гарантией будет прочитана любой Linux-системой. Например, при восстановлении с какого-нибудь древнего подручного LiveCD - ситуация, практически не столь уж невероятная, мне

приходилось в нее попадать. Опять же, большинство BSD-систем легко понимают ext3fs (хотя и без журналирования). Для Windows также имеются, насколько я знаю, всякого рода драйверы и plug-ins к распространенным файловым менеджерам (типа Total Commander), обеспечивающие доступ к разделам с ext2fs/ext3fs.

В отношении производительности ext3fs оставляет противоречивое впечатление. Во-первых, быстродействие ее очень зависит от режима журналирования, каковых предусмотрено три: с полным журналированием данных, частичным их журналированием и журналированием только метаданных. В каждом из режимов она показывает различную производительность на разных типах файловых операций. Впрочем, ни в одном случае быстродействие не является рекордным, в чем можно убедиться, пройдя по [указанной выше ссылке](#).

Впрочем, если требование быстродействия ставится на первое место, то тут вне конкуренции оказывается ext2fs - правда, в этом случае придется смириться с отсутствием журналирования вообще. И, следовательно, с длительными проверками файловой системы при любом некорректном завершении работы - а при объемах современных дисков это может затянуться ой как надолго...

Относительно XFS можно сказать следующее. В плане совместимости к ней относится все то же самое, что написано для ReiserFS - более того, до некоторого времени она не поддерживалась стандартным ядром Linux. С точки зрения быстродействия она XFS она также не блещет, выступая суммарно примерно на одном уровне с ext3fs. А на операции удаления файлов вообще демонстрирует удручающую медлительность.

По моим наблюдениям, использование XFS оправдывает себя при работе не просто с большими, а с очень большими файлами - каковыми являются фактически только образы DVD и видеофайлы.

Возвращаясь к вопросу о надежности. Банальное выключение питания в ходе обычной пользовательской работы, как правило, безболезненно переносят все журналируемые файловые системы (и ни одна из них не обеспечивает сохранности незаписанных на диск пользовательских операций - спасение утопающих и тут остается делом рук самих утопающих). Правда, для любой файловой системы можно смоделировать ситуацию, в ходе которой выключение питания приведет к более или менее серьезным ее повреждениям. Однако в реальной жизни возникновение таких ситуаций маловероятно. А полностью исключить их можно приобретением источника бесперебойного питания - он придаст больше уверенности в сохранности данных, чем тип файловой системы. Ну а гарантией восстановления разрушенных данных в любом случае может быть только их регулярное резервное копирование...

Думаю, изложенной выше информации достаточно для осознанного выбора. Мой личный выбор в течении последних нескольких лет - ReiserFS. Изредка, на системах, где оправданно вынесение за пределы корневого раздела всего, чего только можно, целесообразно использование ext3fs для корневой файловой системы и ReiserFS - для всех остальных.

Если предусматривается отдельный раздел под каталог /boot (а это рекомендуется при использовании загрузчика GRUB его разработчиками) - для него никакая другая файловая система, кроме ext2fs, не оправдана, какое-либо журналирование тут смысла не имеет. Наконец, если создается отдельный раздел под всякого рода мультимедийные материалы - тут можно подумать и о XFS.

В заключение добавлю, что более подробные сведения обо всех рассмотренных здесь материях можно получить из [цикла статей Дэниеля Роббинса](#).

# О Gentoo Linux

Ссылка на оригинал: <http://gentoo.ru/about>

С версии: 1.0

**Gentoo** — английское название вида пингвинов *Pygoscelis papua* (русские варианты названия: субантарктический пингвин, папуанский пингвин, ослиный пингвин). Согласно Книге рекордов Гиннеса 98 этому виду пингвинов принадлежит рекорд скорости плавания (27км/ч).

**Gentoo Linux** — дистрибутив, который может быть автоматически оптимизирован и приспособлен для применения практически в любой области в соответствии с вашими требованиями. **Наивысшая производительность, настраиваемость и высококлассное сообщество пользователей и разработчиков** — вот отличительные черты Gentoo.

Благодаря технологии портежей, Gentoo Linux станет идеальным защищенным сервером, рабочей станцией разработчика, профессиональной настольной системой, системой для игр, встроенной системой или чем-либо другим — всем, чем вы пожелаете. Из-за практически неограниченной способности дистрибутива подстраиваться под нужды пользователя, мы называем Gentoo Linux "метадистрибутивом".

## Что такое портежи?

**Portage** — сердце Gentoo Linux, выполняющее множество ключевых функций. Например, portage — это система распространения программного обеспечения Gentoo Linux. Для того, чтобы получить доступ к последним версиям программ, в Gentoo Linux достаточно набрать всего одну команду: **emerge --sync** (emerge; средство установки/удаления программ (дословно: **emerge** - *выяснять, всплывать, появляться*)). Эта команда заставляет Portage обновить локальное дерево портежей вашей системы через интернет. В локальном дереве портежей содержится полная коллекция сценариев, которые используются для сборки и установки последних версий пакетов Gentoo. Сейчас в нашем дереве Portage более 10000 пакетов, а обновления и новые пакеты добавляются каждый день.

Portage также является системой сборки и установки пакетов. Желая установить пакет, нужно просто набрать: **emerge имя\_пакета**, при этом Portage автоматически соберет вариант пакета «на заказ» в точности по вашим указаниям, оптимизируя его под ваше оборудование и гарантируя, что нужные вам дополнительные возможности включены, ненужные — нет.

Portage также следит за тем, чтобы ваша система не устаревала.

Команда: **emerge -u world** — всего одна команда — позаботится о том, чтобы все нужные именно вам пакеты в системе автоматически обновились.

## **Gentoo Linux**

С помощью портежей можно обновлять систему Gentoo Linux при малейшем желании. Поэтому опытные пользователи Gentoo не обращают особого внимания на «новые версии» Gentoo Linux: в конце концов, свежайшую версию Gentoo Linux всегда можно получить, набрав команду: **emerge --sync**. Нет необходимости ждать несколько месяцев выпуска «новой версии» дистрибутива, поскольку Gentoo Linux непрерывно обновляется и улучшается, а все изменения немедленно становятся доступны пользователям.

Безусловно, мы издаем официальные выпуски Gentoo Linux на компакт-дисках, чтобы вновь устанавливаемые системы с самого начала были как можно свежее. Вот краткий обзор того, что включено в самый свежий выпуск Gentoo Linux:

- *поддержка процессоров x86, AMD64, PowerPC, UltraSparc, Alpha и MIPS*
- *установка на x86, AMD64, PowerPC, UltraSparc и Alpha с диска LiveCD*
- *новейшие стабильные KDE и GNOME*
- *различные оптимизированные ядра Linux*
- *весьма современная среда разработки GNU*
- *отличная поддержка файловых систем: ReiserFS, XFS, ext3, EVMS, LVM*
- *превосходная поддержка оборудования: NVIDIA, Creative Labs Live! и Audigy*
- *модульная подсистема OpenGL и компилятора (с поддержкой сосуществования нескольких версий)*
- *понятные сценарии запуска системы, основанные на зависимостях*
- *новая инициатива системы безопасности «укрепленный Gentoo»*
- *более 10000 пакетов лучших новейших программ*
- *расширенные возможности системы портежей*

Если вас привлекли мощь, гибкость и скорость Gentoo Linux, обязательно испытайте его в деле. Мы уверены: вы не разочаруетесь!

## **История Gentoo**

Все началось с Лишнего Времени. Времени — на изучение, Времени — на исследования, Времени — на эксперименты... Именно так создатель Gentoo Дэниел Роббинс шагнул в мир Linux. Он начал с Debian Linux, настроил пару приложений, изучил Linux от «А» до «Я», а затем, как поступает большинство пользователей Linux, перепробовал еще парочку дистрибутивов, принявшись помогать дистрибутиву под названием Stampede Linux. И вскоре, погрузившись в разработку Stampede, он уже работал над их системой управления пакетами. Через некоторое время, по определенным причинам, он пошел дальше и решил создать свой собственный дистрибутив.



Так родился «Енох». Дэниел хотел, чтобы Енох стал чрезвычайно быстрым дистрибутивом, способным полностью автоматизировать процесс создания и обновления пакетов. Вскоре появился канал #enoch на irc.freenode.net и 10 разработчиков, участвующих в создании дистрибутива. По прошествии времени, когда Енох подрос, им показалось, что дистрибутиву нужно новое имя. Они нарекли его Gentoo Linux. И как раз в то время, когда выход Gentoo 1.0 был на носу, Дэниел купил новую быструю машину. В материнской плате оказался дефектный чип, из-за которого Linux при простое зависал, и из-за этого разработка Gentoo полностью встала.

С тех пор чего только не происходило с Gentoo... Дэниел переключился на FreeBSD. Увиденное ему понравилось. Особенно система портов. И он вернулся в мир Linux. При поддержке других разработчиков, таких как Аким Готтингер, Gentoo снова вышел в путь и стремительно рванулся вперед. Система управления пакетами подверглась полной переработке и получила название Portage. С тех пор год за годом Gentoo пребывает в активной разработке, с постоянным добавлением массы новых возможностей. Команды добровольцев помогают удерживать Gentoo на самом переднем крае, одновременно обеспечивая максимальную безопасность и стабильность.

Модель разработки Gentoo дополнилась полностью проектно-ориентированным подходом, где каждый проект разрабатывается независимо, но в сотрудничестве с другими проектами Gentoo. Регулярные совещания ведущих («менеджеров проектов») поддерживают быстрый темп разработки. Чтобы позаботиться о финансовых вопросах, обеспечить юридическую защиту и удержать разработку Gentoo в рамках общественного договора, создан фонд Gentoo.

В апреле 2004 Дэниел решил оставить пост разработчика Gentoo. Мы все весьма благодарны Дэниелу за его вклад в Gentoo, и желаем ему всего наилучшего.

Сейчас Gentoo продолжает расти, развиваться и улучшаться: появляются новые проекты, присоединяются новые разработчики, каждый день добавляются свежие пакеты. Но, безусловно, наивысшая ценность Gentoo — это сообщество разработчиков и пользователей Gentoo!

## Философия Gentoo

**Ссылка на оригинал:** <http://gentoo.ru/philosophy>

**С версии: 1.0**

**Автор:** Daniel Robbins - изначальный разработчик Gentoo

Перевод: Логинов Игорь

Я начал работу над Gentoo, потому как не мог найти дистрибутива, который бы меня устраивал. Во всех Linux-дистрибутивах я заметил одну неприятную деталь - инсталляционные утилиты, управляющие установкой ПО в системе, предназначались для упрощения работы, однако на практике лишь мешали администрированию системы.

Поэтому я и создал Gentoo с её собственной системой портэжей (т.н. Portages), более совершенной, чем все системы управления ПО, существовавшие до этого. Система портэжей отличается большой гибкостью и даёт возможность пользователю успешно использовать её для решения конкретных задач независимо от поставленных целей.

Например, если вы хотите посмотреть подробности сборки пакета, вы можете

посмотреть файл **ebuild**.

---

**ebuild** - сборочный файл (вероятно, авторское сокращение от emerge build file)

---

и получить всю необходимую информацию оттуда. Чтобы настроить параметры установки, можно воспользоваться переменной **USE**. Чтобы добавить новый пакет, можно просто создать новый файл **ebuild** для дерева портэжей. Вы также можете установить требуемый пакет командой **emerge** - система автоматически проследит за установкой всех необходимых пакетов, от которых зависит нужное вам приложение.

Linux-сообщество оценило новую систему портэжей, и Gentoo начал стремительно развиваться. Дистрибутив получил известность как "**дистрибутив исходников**", однако сутью Gentoo является вовсе не принцип "from source" (из исходников). Конечно, это важный, ключевой принцип дистрибутива, но не основной. Основным здесь является технология, позволяющая пользователю делать с системой всё что угодно, без каких либо ограничений.

Работая над этой технологией, я представлял себе произвольного пользователя ОС Linux. Что он (или она) хотел бы сделать с системой?

## **Философия Gentoo - дистрибутив, дающий возможность делать с системой всё что угодно.**

Камнем преткновения для появления Gentoo было отсутствие простого метода автоматической установки ПО из исходников, ориентированного на пользователей, с подробными инструкциями. На данный момент мы решили эту задачу практически полностью. Частично реализована поддержка прекомпилированных пакетов, хотя, с другой стороны, с момента своего появления система портэжей Gentoo **поддерживала** установку бинарников.

Реализация поддержки бинарных пакетов очень важна, т.к. большое количество ПО распространяется именно в бинарниках (например, в виде пакетов RPM - прим. переводчика). Если бы Gentoo не поддерживал бинарные пакеты, мы бы не смогли бы назвать его удобным дистрибутивом, позволяющим пользователю делать с системой всё, что ему придёт в голову. Впрочем, помимо "философских" причин для поддержки бинарников есть много причин практических.

Философию Gentoo можно сформулировать в нескольких предложениях:

- Каждый линуксоид использует компьютер для решения своих конкретных задач.
- Главной целью Gentoo является предоставление ПО, позволяющего пользователю выполнять свои задачи эффективно.
- Утилиты Gentoo должны быть удобны в использовании, должны позволить пользователю оценить по достоинству все преимущества открытого ПО вообще и Linux в частности. Такое возможно лишь при наличии огромной базы открытого ПО, отражающей все потребности современных пользователей, содержащей тысячи разнообразных программ, по возможности удобных.
- Если некая утилита вынуждает пользователя выполнять какие-то нестандартные действия для совершения стандартных операций - она скорее мешает работе, чем помогает. Каждый встречал подобные утилиты, неудобные в работе, лишённые ряда очевидных функций. Такое ПО противоречит философии Gentoo.



Другими словами, философия Gentoo также заключается в разработке корректного ПО. Когда утилита выполняет свои и только свои обязанности, но выполняет их в полной мере, вы можете забыть о её присутствии (что, впрочем, соответствует философии Unix вообще как таковой - прим. переводчика). Она не мешает, не попытается взаимодействовать с вами, пока вы сами не захотите этого.

В настоящее время в перспективы Gentoo входит поддержка и создание ПО, близкого к идеалу - мощных и простых в использовании утилит, способных обеспечить выполнение множества различных задач. Разве не прекрасно иметь возможности найти ПО, делающее именно то, что вы от него хотите? Разве это не вызывает чувство эстетического наслаждения? Наша задача - донести это чувство до как можно большего числа людей.

## Советы пользователям Gentoo

**Ссылка на оригинал:** <http://ylsoftware.com/?action=news&na=viewfull&news=130>

**С версии:** 1.0

**Автор:** MooSE 2006-11-22 16:35:01

**Gentoo** - один из самых трудных для изучения дистрибутивов. Хотя многие старые пользователи скажут что огромное сообщество и обширная документация могут здорово помочь новичкам. В этой статье приведён ряд подсказок, которые помогут новым пользователям быстрее освоить Gentoo.

Для начала приготовьтесь прочитать огромное количество документации. Вы не сможете установить Gentoo не изучив внимательно [Handbook](#) ([Главное руководство Gentoo- см. ниже: Настольная книга пользователя](#)).

Gentoo имеет графический инсталлятор, но новичкам не стоит его использовать. Установка Gentoo "традиционным" способом вынуждает читать документацию, и способствует лучшему пониманию системы. Так же инсталлятор Gentoo не так хорош при установке и настройке драйверов как инсталляторы других дистрибутивов.

Handbook содержит основную информацию о Gentoo. Дополнительная информация (например о настройке X и ALSA) может быть найдена на странице документации или на Gentoo Wiki.

---

**Вики (Wiki)** (от гавайского: быстрый) — гипертекстовая среда (обычно Веб-сайт) для сбора и структуризации письменных сведений (в программировании: документации, советов, часто задаваемых вопросов(FAQ), мини-руководств(HOWTO)).

Характеризуется следующими признаками:

- Множество авторов. Некоторые вики могут править все посетители.
- Возможность многократно править текст посредством самой вики-среды (вебсайта), без применения особых приспособлений на стороне редактора.
- Проявление изменений сразу после их внесения.
- Разделение информации на страницы, где у каждой своё название.
- Особый язык разметки, позволяющий легко и быстро размечать в тексте структурные элементы, форматирование, гиперссылки и т. п.
- Учёт изменений (учёт версий) текста и возможность отката к ранней версии.

Для создания вики-среды необходимо особое ПО — движок вики. Это частный вид системы управления сайтом, довольно простой в своём устройстве и функциональности, ибо почти все действия по структуризации и обработке сведений делаются пользователями вручную. Главная особенность —

посетители сайта могут сами редактировать содержимое этих страниц и создавать новые. Wiki сделаны так, чтобы редактировать страницы было просто; для редактирования ничего кроме программы веб-браузера не нужно.

**ALSA, или Advanced Linux Sound Architecture** — продвинутая Звуковая Архитектура Linux, обеспечивает аудио- и MIDI- (Musical Instrument Digital Interface — Цифровой Интерфейс Музыкальных Инструментов) функциональность в операционной системе Linux. ALSA является основной звуковой подсистемой в ядрах 2.6, заменившей собой OSS (Open Sound System - Открытая Звуковая Система), которая использовалась в ядрах 2.4

**Х-сервер (или просто Иксы)**- обычного пользователя могут испугать мысли о необходимости печатать в командной строке. Почему бы ему не прокладывать свой путь на территории свободы выбора Gentoo (и Linux вообще), просто «водя мышкой»? Ну, конечно, вы сможете это сделать :-). В Linux предлагается широкое разнообразие пользовательских интерфейсов и сред, которые можно установить поверх установленной у вас системы.

Это один из самых больших сюрпризов, с которым сталкиваются новички: графический интерфейс пользователя — это всего лишь прикладная программа, выполняемая в системе. Он не является частью ядра Linux или каких-либо других компонентов системы. Это — мощный инструмент, позволяющий полностью задействовать графические возможности вашего компьютера.

Так как стандартизация очень важна, для организации прорисовки и перемещения окон по экрану, взаимодействия пользователя с мышью и клавиатурой, а также других важных базовых аспектов, был создан стандарт под названием X Window System, сокращённо X11 или просто X. Он используется в Unix, Linux и других Unix-подобных операционных системах во всем мире.

Приложение, которое дает пользователям Linux возможность запускать графические интерфейсы пользователя и использует стандарт X11, называется Xorg-X11, ответвление проекта XFree86. Проектом XFree86 было решено использовать лицензию, возможно, несовместимую с лицензией GPL, поэтому рекомендуется использовать Xorg. Пакет XFree86 больше не поддерживается в официальном дереве портежей.

### *Проект X.org*

Проектом X.org создана и поддерживается бесплатно распространяемая реализация системы X11 с открытым исходным кодом. Это открытая инфраструктура рабочей станции, основанная на X11.

Xorg обеспечивает интерфейс между устройствами вашего компьютера и нужными вам графическими программами. Кроме того, Xorg является полностью сетевым, то есть вы можете запускать программу на одной системе, а отображать ее — на другой.

Если у вас возник специфический вопрос, не освещённый в документации - попробуйте спросить на форуме или в [IRC](#).

## **Основы Portage**

Одно из фундаментальных отличий Gentoo от других дистрибутивов - его система управления пакетами, названная **Portage**, ориентированная в первую очередь на установку программ из исходных текстов. Это позволяет иметь множество настроек и точек контроля для одного пакета там, где пользователи бинарных дистрибутивов имеют несколько пакетов с разными опциями и зависимостями (фактически это тоже самое, только управление различными опциями сборки не такое гибкое). С Portage вы можете установить так называемые **USE-флаги** для настройки опций одного пакета или даже всей системы.

Например если вы хотите включить поддержку KDE и выключить поддержку GNOME вам нужно добавить к строке USE в файле /etc/make.conf флаги "**kde -gnome**". Для получения более подробной информации о USE-флагах смотрите [Gentoo Handbook](#).

Для работы с Portage используется команда **emerge**. Например для установки пакета **moo** достаточно набрать команду: **emerge moo**. Portage проверит зависимости, скачает все необходимые архивы с исходниками, соберёт пакет и установит его. Вся необходимая информация сохранена в базе, состоящей из **ebuild**'ов, каждый из которых представляет собой специальный скрипт. Все **ebuild**'ы находятся в папке **/usr/portage**.

## Управление настройками Portage

Первое правило Gentoo - нужно внимательно следить за флагами **USE**. Почему при установке Midnight Commander устанавливается X.Org - потому что у вас разрешён флаг X. Почему нет KPDF в KDE? - Потому что KDE был собран без флага **pdf** (не спрашивайте почему всё же собрался KGhostScript).

Всегда устанавливайте новые пакеты командой **emerge --ask --verbose** (или **emerge -av**) для просмотра USE-флагов перед началом компиляции. Установите пакет **gentoolkit** (**emerge gentoolkit**) - среди множества полезных вещей он содержит утилиту **euse**, которая существенно упрощает управление флагами USE\*. Выполните **euse -i flag** для получения информации о флаге "flag". Так же просмотрите ман-страницу **euse** (не волнуйтесь. она короткая :-)).

Если вы хотите изменить USE-флаги только для одного пакета не пытайтесь собирать его командой **USE="some flags" emerge moo**. Это конечно сработает, но при обновлении системы пакет будет пересобран со старыми флагами. Лучше прописывайте USE-флаги в **/etc/portage/package.use**, для этого выполните **echo "category/moo some flags" >> /etc/portage/package.use** перед установкой пакета.

Тоже относится и к **ACCEPT\_KEYWORDS** - используйте **/etc/portage/package.keywords**. В Portage 2.1 вы можете использовать папки **package.use** и **package.keywords** вместо файлов для более простого управления.

## Когда нужно использовать оверлеи?

**Оверлей** - набор **ebuild**'ов, которые вы хотите добавить к официальной базе данных (дереву) Portage

---

Иногда возникает необходимость установить пакет, отсутствующий в Portage. Например вы можете захотеть установить пакет, удалённый из дерева Portage. Тогда вам нужно взять его **ebuild** из архива Portage CVS. Или например вы захотите установить новый пакет, взяв **ebuild** в bugzilla. Возможна ситуация, когда вы просто захотите немного подправить официальный **ebuild**.

В любом случае вам нужно в такой ситуации поместить ваш **ebuild** в локальный оверлей. Оверлей - набор **ebuild**'ов, которые вы хотите добавить к официальной базе данных (дереву) Portage.

**Не храните свои ebuild'ы внутри официального дерева (/usr/portage), поскольку при обновлении они будут удалены.**

---

\* Также имеется утилита: **ufed** (установка: **emerge ufed**) для более дружелюбного просмотра USE флагов

Создайте каталог для оверлея, и создайте в нём структуру подкаталогов, аналогичную **/usr/portage**. Поместите туда ваш **ebuild**. Добавьте **PORTDIR\_OVERLAY="/path/to/local/overlay"** в ваш файл **/etc/make.conf**.

Обычное место для оверлеев - **/usr/local/overlays**. Таким образом местом для вашего **ebuild**а будет **/usr/local/overlays/local**. Что же ещё может находиться в **/usr/local/overlays**.

Во многих случаях, когда вы ищите пакеты, не включённые в официальное дерево, вы находите не только **ebuild**ы, но и оверлеи. Например один из самых популярных оверлеев **xgl-coffee** содержит всё, что нужно для работы XGL. Вы можете установить его в **/usr/local/overlays/xgl-coffee** что не смешивать с другими оверлеями и обновлять его через SVN.

Вы можете найти подробную информацию относительно использования оверлеев на [Gentoo Wiki](#). Там же вы найдёте список неофициальных оверлеев и описание процесса синхронизации с ними.

### **Как часто нужно обновляться?**

Вам нет смысла обновляться чаще чем раз в сутки. Можно даже реже. Вы можете следить за вашими любимыми пакетами периодически посещая сайты **packages.gentoo.org** или **gentoo-portage.com** (или подписавшись на RSS).

### **Сохраните мир чистым !**

Каждый раз, когда вы выполняете **emerge some-package**, имя пакета записывается в файл **/var/lib/portage/world**. Когда вы обновляете систему командой **emerge --update --deep world** - используется список из этого файла. Поэтому этот файл должен быть по возможности короче. В нём должны быть только необходимые пакеты.

**Если какой-то пакет является всего лишь зависимостью другого - не стоит держать его в файле world.**

Например, если вы устанавливаете пакет, использующий библиотеку wxWidgets, то эта библиотека устанавливается как зависимость и не добавляется в файл world. Если вы деинсталлируете все пакеты использующие эту библиотеку - она перестанет обновляться при выполнении **emerge --update (...) world**, и будет удалены после вызова **emerge --depclean**. Единственный случай когда стоит добавить wxWidgets в world - если вы разрабатываете приложения на основе этой библиотеки.

Существует скрипт, помогающий очистить world от всего лишнего. Для установки пакета без занесения его в world используйте опцию **--oneshot**. Эта опция так же будет хорошим решением при временной установке какого-либо пакета.

### **Деинсталляция пакетов**

Все ваши действия с emerge записываются в файл `/var/log/emerge.log`. Если вы хотите знать какой пакет собирается в настоящее время - выполните **`tail /var/log/emerge.log`**. Утилита **`genlop (emerge genlop)`** позволяет извлечь множество различной информации из этого файла. Например `genlop -с` покажет вам какой пакет собирается в данный момент и попытается предсказать сколько времени осталось до конца сборки.

## **Удаление временных файлов**

Когда сборка пакета завершается с ошибкой - остаётся каталог с временными файлами в `/var/tmp/portage`. Если вы устраняете проблему и пакет нормально собирается - файлы удаляются. Если же нет - вы можете очистить каталог `/var/tmp/portage` вручную.

После обновления вы можете удалить устаревшие файлы из `/usr/portage/distfiles`. Для этого воспользуйтесь утилитой **`eclean`** из пакета **`gentoolkit`**. Она поможет удалить ненужные файлы. Эта утилита может работать как с исходниками, так и с бинарными пакетами. (просто вызовите её как **`eclean-dist`** или **`eclean-pkg`** соответственно). Прочитайте страницу man для получения полного списка опций.

Gentoo предоставляет вам огромные возможности по настройке вашей системы. Автор статьи надеется что после прочтения этих строк вы сможете воспользоваться этими возможностями и испытать чувство глубокого удовлетворения от возможности управления любыми аспектами системы. Вам понравится это ощущение.

## **Gentoo Linux: установка и настройка\***

Ссылка на оригинал: [http://posix.ru/distro/gentoo\\_old/](http://posix.ru/distro/gentoo_old/)

Автор: Алексей Федорчук ([alv@linux-online.ru](mailto:alv@linux-online.ru))

С версии: 1.0

### **О Gentoo Linux**

**Gentoo Linux** - один из типичных представителей т.н. **Source Based** дистрибутивов. То есть систем, основанных не на прекомпилированных бинарных пакетах, а собираемых непосредственно из исходных текстов. **Gentoo** относительно молод, но известностью сравнялся с такими ветеранами дистрибирования, как **Red Hat** или **Debian**.

**Gentoo** создан в 2001 г. Дэниелом Роббинсом. Официальный сайт проекта – <http://www.gentoo.org>. (Русский: <http://gentoo.ru>)

С самого начала Gentoo являл собой синтез наиболее передовых технологий Linux с традиционными достижениями BSD-систем (в первую очередь - **FreeBSD**). В

---

\* Лучшее для новичка (прим.: JohnBat26)

частности, в нем приняты сценарии начальной загрузки системы в BSD-стиле - большинство популярных дистрибутивов **Linux** наследуют схему инициализации системы, унаследованную от **Unix System V** (именно она была принята за основу Линусом Торвальдсом при создании своего ядра).

Главная особенность **Gentoo** также идейно унаследована от **FreeBSD**. В последней, как известно, только базовые компоненты (т.н. **Distribution**) устанавливаются из прекомпилированных пакетов. Для установки же пользовательских приложений предназначена т.н. система портов. Это - набор средств для автоматизированного получения из Сети (с мастер-сайтов разработчиков или их официальных зеркал) исходных текстов программ, их конфигурирования, компиляции, установки и регистрации в специальной базе данных. Конечно, и **FreeBSD** поставляется с наборами прекомпилированных пакетов (**packages**), однако роль их сугубо вспомогательная.

В **Gentoo** концепция портов (именуемых здесь портежами - **Portages**) проводится еще более последовательно, чем во **FreeBSD**. Поскольку здесь этим путем можно установить (или - обновить) не только прикладные программы, но и все базовые компоненты системы, вплоть до ядра, компилятора **gcc** и главной системной библиотеки **glibc**.

Система портежей **Gentoo** позволяет:

1. очень гибкое конфигурирование системы, с подбором только необходимых (и осознанно выбираемых) приложений;
2. открыть путь для оптимизации системы не только под различные архитектуры (а на данный момент дистрибутивом поддерживаются, кроме i86, также Sparc, PowerPC и Alpha), но и под конкретные типы процессоров;
3. средства генерации бинарных пакетов (аналог **packages FreeBSD**), которые могут быть установлены на целевую платформу, отличную от host-машины;
4. оптимизировать пакеты с учетом особенностей целевой (например, более слабой) машины.

Дополнительным плюсом дистрибутива **Gentoo** является его прекрасная документированность. На сайте проекта доступны [\*\*Gentoo Handbook\*\*](#) и множество более частных источников информации, ряд из которых переведен на русский язык.

Название дистрибутива (произносится как "джен-ту") может показаться несколько странным. Но объясняется просто - это один из видов пингуина (научное название *Rygoscelis parea*), обитающий на Циркум-Антарктических островах (Фолклендских, Кергелен, Южная Георгия).

В специальной литературе можно прочитать, что джен-ту отличается небольшим размером и стремительностью. Именно эти его качества и воплощены в дистрибутиве **Gentoo**.

## **Получение дистрибутива**

Все версии дистрибутива **Gentoo** доступны на основном сервере проекта и на ряде его зеркал. Скачивание - по протоколам http и ftp, разумеется, свободно и бесплатно. Конечно, за некоторую сумму можно заказать и CD-вариант, но в наших условиях это не актуально.



Процесс получения **Gentoo** имеет некоторые особенности, отличающие его от получения (приобретения) любого пакетного дистрибутива. Если для последнего достаточно скачать iso-образы CD-дисков и записать их на соответствующие носители, продуктивное использование Gentoo требует выхода в Интернет. Правда, не обязательно с той же машины, на которую он устанавливается - вполне допустима ситуация, когда необходимые компоненты получаются, например, по рабочему каналу, а сама система устанавливается дома.

Итак, первое, что необходимо для установки - скачать образ инсталляционного диска. Объемы образов - достаточно скромны, обычно около 200 Мбайт (в зависимости от версии). Образы эти являют собой **Live CD** - то есть фактически полноценную Linux-систему, запускаемую с компакт-диска. И включающую набор не только системных утилит, но и пользовательских приложений.

Образ инсталляционного диска обычно содержит все необходимое для установки. Однако в некоторых версиях его оказывается недостаточно - потребуется еще три архива (\*.tar.bz2), содержащие (в прекомпилированном виде) компоненты базовой системы. Первый из них (файл **stage1-\***, объемом около 10 Мбайт) специфичен для архитектуры (i86, PowerPC, Sparc и т.д.). Два других тарбалла (**stage2-\*** и **stage3-\***, около 50 и 80 Мбайт, соответственно), кроме этого, для PC-архитектуры представлены в нескольких вариантах, оптимизированных под конкретные процессоры: от i486 до Pentium-4 (включая и Athlon). Ясно, что скачиванию подлежат только тарбаллы, соответствующие реалиям пользователя.

Далее, для установки пре-релиза потребуется скачать архив системы портежей - один из файлов вида **portage-200XXXXX.tar.bz2**, расположенных в каталоге **~/gentoo/snapshots** (в стабильной версии текущий архив портежей размещен на установочном диске). Как легко понять, цифры в имени файла соответствуют дате создания архива (а обновляется система портежей чуть ли не ежедневно), и лучше выбрать последний по времени.

И наконец, при установке на машину без выхода в сеть неплохо заблаговременно скачать архивы исходных текстов базовых компонентов системы - об их составе я скажу несколько позже. И не обязательно с сайта разработчика: исходники всех портированных в **Gentoo** программ можно найти на сайте проекта в каталоге **~/gentoo/distfiles** (или **~/gentoo/gentoo-sources**, содержание их идентично). Наличие их очень упростит жизнь, однако, как будет показано ниже, обойтись без них можно.

А вот без чего никак не обойтись - это без одного из дополнительных носителей, на котором будут размещены скачанные компоненты. При установке на "чистую" машину ими могут быть: второй CD (при наличии двух приводов, например, пишущего и читающего) или Zip (как нетрудно подсчитать - желательно 250-мегабайтного объема, иначе их придется менять в процессе установки). Можно также временно подцепить дополнительный винчестер с уже существующим разделом и какой-либо файловой системой (из числа поддерживаемых) на нем. Ну и заблаговременно переписать на эти носители все требуемые файлы - тарбаллы **stage2-\***, **stage3-\*** и **portage-\*** - обязательно, прочие - по возможности и желанию.

К слову сказать, на тот же носитель хорошо поместить и инструкцию по установке с сайта **Gentoo** (добавлю, и все прочие файлы документации - тоже, отнюдь не повредят). Или, по крайней мере, держать ее распечатку под рукой - зело способствует. И можно приступать к установке.

Помещать тарбаллы базовой системы и системы портежей, а также документацию,

можно где угодно - хоть в корне файловой системы дополнительного накопителя. А вот под архивы исходников лучше создать на нем отдельный каталог, например, `src` - ниже будет ясно, почему.

Загрузка системы и первые шаги

Редкий инсталляционный диск Linux-дистрибутива ныне не является загрузочным. И **Gentoo** тут - не исключение: чтобы загрузиться с любого из вышепоименованных образов, достаточно поместить диск в привод, нажать на `Reset`, выставить соответствующие опции в `Setup BIOS` и подождать некоторое время, по возможности вникая в появляющиеся сообщения об имеющемся оборудовании (впрочем, все эти сведения можно будет получить и потом - командой **`dmesg`**).

Ожидание завершается выводом приглашения на ввод пользовательского имени (**`login`**), ответом на что служит волшебное слово **`root`**, а затем и пароля - на это отвечаем просто нажатием клавиши **`Enter`**. За этим появляется приглашение командной строки - и можно приступить к дальнейшим действиям.

Они не вполне тривиальны. Еще одна отличительная особенность дистрибутива **Gentoo**, о которой я не упомянул ранее - полное отсутствие какого-либо инсталлятора, все манипуляции по установке осуществляются прямыми директивами из командной строки и прямым редактированием конфигурационных файлов в текстовом редакторе. Что начинающего (да и не очень начинающего) пользователя может просто обескуражить.

Однако все не так страшно. Во-первых, в нашем распоряжении - вышеупомянутая подробная, буквально пошаговая, инструкция по установке. А во-вторых (и об этом также говорилось ранее), установочный диск **Gentoo** - не просто загрузочный, а так называемый Live CD. То есть просто-напросто загружающий полноценную рабочую Linux-среду с корневой файловой системой в оперативной памяти (способной монтировать в себя файловые системы любых устройств). И потому в процессе установки в руках пользователя - вся мощь ОС Linux, включая командную оболочку **`bash`** и 6 виртуальных консолей (авторизоваться в любой из них можно точно так же, как и в первой - вводом имени `root` без пароля). Ну а в третьих - загружаемое с CD Linux-ядро собрано с поддержкой абсолютно всех опций, которые могут потребоваться при установке (встроенной в ядро или модульной).

И потому первое действие после старта системы - загрузка всех необходимых модулей. Это будут модули поддержки:

- устройств хранения данных - SCSI-адаптеров, контроллеров IDE-RAID, Zip-приводов (`atari-floppy`, в терминологии Linux), короче, всего, на что мы собираемся устанавливать систему или считывать устанавливаемые компоненты;
- файловых систем: это потребуется, только если устанавливаемые тарбаллы и архивы исходников размещены на разделе с файловой системой FATxx - поддержка всех родных (`native`) файловых систем Linux, включая XFS, встроена в установочное ядро);
- сетевого адаптера - если таковой, вместе с постоянным подключением к Сети, имеется;
- соединения `ppp` - это даст возможность использовать при установке модемное подключение к Сети.

При этом возможно, что многие из перечисленных устройств (в частности, сетевые карты) будут корректно определены при загрузке системы и соответствующие модули загрузятся автоматически, что проверяется командой



**\$ modprobe** имя\_модуля

В остальных случаях это придется сделать вручную, командой

**\$ insmod** имя\_модуля

Имена поддерживаемых модулей можно посмотреть (авторизовавшись в следующей виртуальной консоли) в соответствующих подкаталогах каталога **/lib/modules/2.6.XX.X/kernel/** - ведь наша корневая система пока находится в оперативной памяти.

Следующий шаг - настройка сетевого соединения (при постоянном подключении). Если машина использует динамически выделяемый IP-адрес, назначаемый через DHCP-сервер (а в большинстве случаев это так и есть), делается это элементарно просто, командой

**\$ dhcpcd eth0**

Все остальное происходит почти автоматически. При статическом IP-адресе потребуются ручная настройка, но на этом я останавливаться не буду (все вопросы конфигурирования сети очень подробно рассмотрены в руководстве).

При отсутствии постоянного подключения необходимо смонтировать носитель со скачанными ранее архивными файлами. Для этого в каталоге **/mnt** создаем соответствующую ситуации точку монтирования, например,

**\$ mkdir /mnt/zip**

для Zip-диска, или

**\$ mkdir /mnt/cd2**

для второго CD (каталог для установочного CD - **/mnt/cdrom**, - там уже имеется, и диск этот в него уже смонтирован). А теперь даем команду монтирования. Для zip-диска "из коробки" она будет выглядеть так:

**\$ mount /dev/hdd4 -t msdos /mnt/zip**

где, кроме явного указания файловой системы, обращаю внимание, что файл устройства - именно **/dev/hdd4**: при фабричной разметке на zip-дисках создается один, но четвертый, первичный раздел. И еще одна тонкость: вследствие особенностей реализации файловой системы **devfs** в дистрибутиве **Gentoo**, zip-диск должен находиться в приводе в момент старта системы - иначе соответствующий файл устройства просто не будет создан.

Аналогичны и действия по монтированию второго CD или дискового раздела. Единственно, что в этом случае может не потребоваться явного указания на тип файловой системы (хотя и вреда от него, конечно, не будет).

Завершив монтирование носителя установочных файлов, следует перво-наперво активизировать еще одну виртуальную консоль и вывести на нее [инструкцию по инсталляции](#) - дабы сверять с ней все дальнейшие шаги, как со Священным писанием:

**\$ lynx /mnt/mount\_point/build.html**

При наличии модемного подключения к Сети можно настроить и его - даже если все требуемые файлы, включая исходники базовой системы, размещены на смонтированном носителе, не исключено, что в процессе их синхронизации может потребоваться какой-либо специфичный для **Gentoo** патч, скачать который по

модему - дело пары минут, а то и секунд. Хотя должен предупредить, что установить базовую систему чисто по модему (что вполне реально, скажем, для **FreeBSD**) - не то что невозможно, но, скажем так, весьма долго и накладно для кармана.

Для настройки модемного соединения проще всего использовать программу **wvdial**, находящуюся на Live CD (в каталоге **/usr/bin**). Подгрузив модули поддержки протокола **ppp** (общий, **ppp\_generic**, и асинхронного соединения, **ppp\_async**), запускаем программу

```
$ wvdialconf /etc/wvdial.conf
```

Она автоматически определит порт, к которому подключен модем, определит его скорость и создаст строку инициализации, занеся все эти сведения в файл, указанный в качестве аргумента. В результате содержание его примет вид, подобный следующему:

```
[Dialer Defaults]  
Modem = /dev/ttyS0  
Baud = 115200  
Init1 = ATZ  
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0  
ISDN = 0  
Modem Type = Analog Modem
```

Пользователю остается только добавить туда строку, определяющую способ дозвона. По умолчанию он - тоновый, тогда как в наших условиях, скорее всего, потребуется импульсный дозвон:

```
Dial Command = ATDP
```

Остальное же - обычная информация: телефон провайдера

```
Phone =
```

имя пользователя и его пароль

```
Username =  
Password =
```

Теперь, проследив, чтобы в файле **/etc/hostname** наличествовала единственная строка

```
localhost
```

подкорректируем содержимое файла **/etc/resolv.conf**: значение строки **nameserver** должно соответствовать реальному IP-адресу DNS-сервера провайдера. И теперь при необходимости для дозвона можно использовать команду

```
$ wvdial
```

А больше на этапе установки нам ничего и не требуется - все остальное сделает система портежей.

Возникает вопрос: а каким образом редактировать все вышепоименованные конфигурационные файлы (и те файлы, которые потребуется редактировать в

дальнейшем)? Этой цели в дистрибутиве Gentoo штатно служит текстовый редактор **nano** - очень простой в освоении и обращении, снабженный встроенной справкой и обеспечивающий достаточные базовые функции редактирования. Нужно только не забывать запускать его в форме  
**\$ nano -w** имя\_файла

где опция -w отключает режим переноса строк - конфигурационные файлы на это, как правило, реагируют болезненно. Поскольку редактор понадобится нам за время установки неоднократно, резонно было бы сразу определить псевдоним:

**\$ alias nano='nano -w'**

благо, как уже говорилось, в нашем распоряжении полнофункциональная оболочка **bash**.

### **Установка - первый этап**

Вот теперь подготовительные мероприятия закончены - можно начинать собственно установку, начальный этап которой происходит внутри корневой файловой системы Live CD. Первый шаг на этом этапе - создание дисковых разделов и файловых систем на них. Эти вопросы были предметом детального рассмотрения в моей статье, подготовленной к печати (Открытые системы, #11, 2002) и потому здесь я на них останавлиюсь лишь вкратце.

Для первой операции можно использовать одну из программ - **fdisk**, **cfdisk**, **parted**, все они имеются на Live CD. А созданию файловых систем служат специфичные для них утилиты: **mkfs** для **ext2fs**, **mkfs.xfs** - для XFS, **mkswap** - для раздела подкачки, и так далее. Рекомендуемая в руководстве схема разбиения для одного диска (первого на первом IDE-канале) приведена в табл. 1.

**Таблица 1.** Рекомендуемая схема разбиения диска для установки Gentoo

Раздел	Размер	Тип	Устройство
Загрузочный	100 Мбайт	ext2fs или ext3fs	/dev/hda1
Раздел подкачки	128 Мбайт=<2*RAM=<2 Гбайт	Linux swap	/dev/hda2
Корневой раздел	1,5 Гбайт минимум	XFS, ext3	/dev/hda3

Разумеется, эта рекомендация - приближительна. Так, 1,5 Гбайт для корневого раздела - лишь абсолютно необходимый минимум. При интенсивном использовании системы портежей цифру эту нужно увеличить в два-три раза, а то и более. Или выделить файловую систему **/usr** в отдельный раздел. Кроме того, на мой взгляд, абсолютно необходимо создать отдельный раздел под файловую систему **/home** - для домашних каталогов пользователей. А в некоторых случаях целесообразно выделение и каталогов **/var** и **/tmp**.

Если под установку **Gentoo** предполагается задействовать более одного физического диска, можно прибегнуть к таким современным технологиям, как программные RAID-массивы или система управления логическими томами (LVM - Logical Volumes Manager). Рассмотрение их далеко выходит за рамки настоящей статьи. Скажу только, что первая технология способствует повышению быстродействия дисковых операций или (иногда - и) сохранности данных. А система LVM позволяет достичь непревзойденной гибкости в управлении уже созданными и

смонтированными файловыми системами (в частности, динамического перераспределения занимаемого ими дискового пространства). И, разумеется, на установочном Live CD содержится весь необходимый инструментарий для работы с программными RAID-массивами и системой LVM. Впрочем, нет препятствий и к использованию дисков, подключенных к аппаратному контроллеру IDE-RAID, хотя это потребует некоторых дополнительных действий (детали - в руководстве по установке).

Следующий шаг - монтирование созданных файловых систем в структуру корневой системы (напомню, она - в оперативной памяти). Но сначала - активизируем раздел подкачки:

```
$ swapon /dev/hda2
```

что будет способствовать быстродействию при всех дальнейших манипуляциях, особенно при не очень большом объеме ОЗУ.

Теперь необходимо создать точку монтирования для той файловой системы, которая в дальнейшем станет корневой:

```
$ mkdir /mnt/gentoo
```

и смонтировать в нее соответствующий раздел:

```
$ mount /dev/hda3 /mnt/gentoo
```

Если создавались отдельные разделы под каталоги /boot, /usr, /home - они тоже потребуют своих точек монтирования, теперь уже относительно каталога /mnt/gentoo:

```
$ mkdir /mnt/gentoo{/boot,/usr,/home}
```

и выполнить для них команды монтирования:

```
$ mount /dev/hda1 /mnt/gentoo/boot
```

и так далее. Теперь командой

```
$ cd /mnt/gentoo/
```

переходим в будущий корневой каталог Gentoo и вспоминаем наконец о наших архивных файлах. Разворачиваем тарбалл первой очереди, находящийся на инсталляционном CD:

```
$ tar -xvjp /mnt/cdrom/nocompress/stage1-x86-*.tar.bz2
```

Здесь и далее следует обратить внимание на опцию -p в командной строке: она необходима для сохранения оригинальных атрибутов доступа к файлам, извлекаемым из архива. По самой же команде разворачивается иерархия каталогов в дереве /mnt/gentoo (/mnt/gentoo/etc, /mnt/gentoo/proc и т.д.) и устанавливается некий минимум прекомпилированных программ, необходимых для дальнейшей инсталляции, в том числе и ftp-клиент wget. И действительно, при постоянном подключении к Сети более ничего не потребуется - можно переходить к подготовке смены корневого каталога. Однако тем, кто не может им похвастаться, предварительно придется распаковать два оставшихся тарбалла, stage2\* и stage3\*.

Делается это точно так же, как и развертывание первого, только пути к соответствующим файлам нужно указать те, что соответствуют их реальному расположению, например:

```
$ tar -xvjp /mnt/zip/stage2*.tar.bz2
```

и так далее. После этого мы имеем в каталоге **/mnt/gentoo** тот самый набор прекомпилированных библиотек, утилит и приложений, который объединяется понятием base Linux. И, соответственно, морально готовы к тому, чтобы сделать каталог **/mnt/gentoo** корневым.

## **Этап второй - подготовка к смене корня**

Однако этот ответственный шаг требует некоторых подготовительных мероприятий - ведь после него все остальные ветви нынешнего корневого каталога станут недоступными. Правда, только с текущей, первой, консоли - во всех прочих сеансах корневым останется каталог в оперативной памяти.

Для начала, чтобы не делать двойную работу, скопируем настроенные нами конфигурационные файлы - ведь и они пока существуют только виртуально:

```
$ cp /etc/resolv.conf /mnt/gentoo/etc/resolv.conf
```

```
$ cp /etc/hostname /mnt/gentoo/etc/hostname
```

```
$ cp /etc/wvdial.conf /mnt/gentoo/etc/wvdial.conf
```

Теперь воспользуемся замечательным свойством современных ядер Linux - возможностью доступа к одной и той же файловой системе через разные точки монтирования. И смонтируем таким образом файловую систему процессов (она, как ясно из названия, представляет протекающие в системе процессы в виде файлов):

```
$ mount -o bind /proc /mnt/gentoo/proc
```

К слову сказать, таким же образом можно поступить и с тем носителем, на котором у нас размещены архивы системы портежей и исходников (если последние мы туда помещали). А можно, просто размонтировав его (доступ к ним из виртуального корня Live CD больше не обязателен), смонтировать заново - но уже относительно будущего нового корня:

```
$ mount /dev/hdd4 /mnt/gentoo/mnt
```

Ведь в нем развернута собственная корневая система, имеющая все положенные компоненты. Правда, в этом случае мы потеряем доступ к инструкции по установке - а она нам еще потребуется. Вообще говоря, обойтись с этим носителем можно самыми разными способами - например, не перемонтируя его, просто обращаться с других виртуальных консолей. Я опишу самый, на мой взгляд, удобный.

Не предпринимая никаких размонтировочных действий, переходим в каталог **/mnt/gentoo/usr** (таковой ведь у нас тоже имеется). И из него разворачиваем архив портежей, точно так же, как мы это делали для основных тарбаллов:

```
$ tar -xvjp /mnt/zip/portage*.tar.bz2
```

Если исходников на вспомогательном носителе нет - можно оставить его в покое, нужды в нем больше не возникнет. Если же исходники имеются - создаем в новообразованном каталоге **/mnt/gentoo/usr/portage** подкаталог **distfiles** и копируем в него все наши архивы исходных текстов:

```
$ cp /mnt/zip/src/* /mnt/gentoo/usr/portage/distfiles
```

Теперь они находятся именно там, где их ожидает встретить система портежей, прежде чем обращаться к Сети (и выдавать сообщение об ошибке при отсутствии подключения). А дополнительный носитель оставляем смонтированным - чтобы подсматривать в документацию через другой виртуальный терминал.

Теперь все готово к волнующему моменту - смене корневого каталога. Итак, командой

```
$ chroot /mnt/gentoo /bin/bash
```

объявляем каталог **/mnt/gentoo** корнем, одновременно запуская новый экземпляр командной оболочки (уже из нового источника). Теперь пересчитываем пути к системным библиотекам

```
$ env-update
```

ответом на что будет сообщение

```
Regenerating /etc/ld.so.cache...
```

И устанавливаем для нового экземпляра оболочки переменные окружения:

```
$ source /etc/profile
```

Все, мы - внутри системы Gentoo Linux, причем уже очень даже реальной, не имеющей никакого отношения к виртуальной системе, загруженной с Live CD. И далее все пути к файлам в абсолютной форме (типа **/usr/portage**) будут указываться с учетом того, что отныне наш / - это не что иное, как **/mnt/gentoo**.

### ***Третий этап - внутри Gentoo***

Итак, мы уже как бы установили операционную систему **Gentoo** Linux (без перезагрузки, прошу обратить внимание). И вот тут-то и начинается самый охмурень - рядом планомерных и последовательных действий мы можем после перезагрузки (ну куда же без нее) получить не просто полнофункциональную и аккуратную настроенную, но, не побоюсь громких слов, - идеально оптимизированную систему.

Требуемые к тому действия описываются в предположении: а) установки всех трех тарбаллов и системы портежей, б) помещения в каталог **/usr/portage/distfiles** (уже относительно нового корня - **/mnt/gentoo**) некоторого набора минимально необходимых исходников, и в) хоть какого-то соединения с Интернетом (на крайний случай). Без последнего можно, как я неоднократно говорил, и обойтись - но тут уж потребуются ювелирная точность в выборе предварительно скачанных (и ныне помещенных в **/usr/portage/distfiles**) архивов исходников.

В чем смысл этих действий? Да ни много, ни мало, как в полной пересборке базовой системы (те, кто имел дело с FreeBSD, неизбежно вспомнят тут **make world**) и доустановки необходимых компонентов. Поскольку это - не **apt-get**, и не установка прекомпилированных пакетов, все требуемое извлекается непосредственно из исходников. И потому первое, что нам потребуется - настройка условий компиляции.

Глобальные параметры сборки посредством системы портежей описываются в файле **/etc/make.conf**. Вызываем его в редакторе **nano** (не забыв про опцию **-w**, если не решили этот вопрос разом - определением псевдонима) и начинаем действовать.

Первое, что бросается в глаза после открытия вышеупомянутого файла - переменная **USE**. Смысл ее - в определении условий предкомпиляционного конфигурирования исходников. То есть, она эквивалента опциям **--enable/disable** или **--with/without** скрипта **./configure**, с исполнения которого начинается ручная сборка любой программы. И, задав значения этой переменной (несложно догадаться, что указав

```
USE="gpm"
```

мы включаем поддержку службы консольной мыши во всех собираемых через портежи программах (разумеется, тех, где это имеет физический смысл). А задав значение

**USE="-gpm"**

такую отключаем раз и навсегда (на самом деле - не совсем навсегда, но это тема отдельной беседы).

Список включенных по умолчанию значений переменной **USE** можно подсмотреть в файле **/usr/portage/profiles/default-x86-XX/make.defaults**. А полный (и логично классифицированный) список доступных значений можно найти в файле документации - **use-howto.html**. Мы ведь списали его на дополнительный носитель? Да? - тогда можем вызвать с очередной виртуальной консоли...

Следующая переменная файла **/etc/make.conf** -

**CHOST='i686-pc-linux-gnu'**

Она указывает на архитектуру машины, на которой будут собираться программы. И ее, при необходимости, нужно будет привести в соответствие с реалиями оной. Правда, ныне такая необходимость возникнет редко - разве что поменять первую шестерку на пятерку в случае **Pentium**-просто...

А вот следующая строка,

**CFLAGS=""**

очень важна. Ею задаются именно те флаги оптимизации, передаваемые компилятору для оптимизации под конкретный процессор. Вдаваться в детали опять же не буду - и это тема особого, и очень подробного, разговора. Достаточно помнить, что флаг **-O3** (а именно он указан здесь по умолчанию) устанавливает самый высокий уровень оптимизации, а оба флага **-mcpu=\*** и **-march=\*** указывают, под какой именно процессор оптимизировать следует. Различие между ними - в том, что программа, собранная с флагом **-mcpu=i686**, будучи оптимизирована под класс процессоров **Pentium Pro** и выше (а с формальной точки зрения, и **P-II**, и **P-III**, и даже **P-4** принадлежат к нему), теоретически способна запуститься и на **Pentium-100**, тогда как флаг **-march=athlon-xp** потребует именно того процессора, который указан - даже на просто **Athlon'e** скомпилированная с ним программа работать не будет.

Смысл переменной

**CXXFLAGS=**

аналогичен - только для программ, написанных не на Си, а на Си++. И ее имеет смысл определить просто, через переменную **CFLAGS=**

**CXXFLAGS="{CFLAGS}"**

как это и сделано в **example**-файле. Остальные строки которого определяют условия скачивания (в частности, **ftp**-клиента по умолчанию). Менять их на данном этапе - оснований не вижу, кроме разве что идиосинкразии к **wget**, используемому по умолчанию.

Определившись с условиями, можно начинать перестройку. Для этого меняем каталог

**\$ cd /usr/portage**

и запускаем сценарий  
**\$ scripts/bootstrap.sh**

Он производит сборку пакетов **binutils**, **gcc**, **gettext** и **glibc** (замещая установленные из тарбаллов прекомпилированные их варианты), а после сборки **glibc** - пересборку **binutils**, **gcc** и **gettext**. Достигая, таким образом, их идеальной оптимизации... Хотя и ценой временных затрат - процесс даже на мощной машине вполне может занять часок-другой.

Теперь можно полюбоваться на список пакетов, установленных в нашей системе - ибо это и есть тот самый base Linux с очень небольшими коррективами. Для этого даем команду

**\$ emerge -p system**

Если есть желание, время, а главное - полный набор исходников в **/usr/portage/distfiles**, можно завершить пересборку базовой системы командой

**\$ emerge system**

Здесь без постоянного коннекта с очень большой вероятностью рано или поздно последует сообщение об ошибке - о невозможности получить пакет имя\_рек с такого-то адреса. Если имеется модем, к подбору исходников мы подошли ответственно, - не беда: активизируем еще одну виртуальную консоль (а на ней будет жить наша первозданная Live-система с CD) и дозваниваемся до провайдера через программу **wvdial** (а она у нас пока существует только в той самой виртуальной Live-системе), а потом повторяем последнюю команду. И, скорее всего, дело было в каком-нибудь исконно Gentoo'ском патчике объемом в пару килобайт, так что процедура эта нас не разорит. Если же обуюет приступ жадности, можно просто продолжить, а к **emerge system** вернуться в другой раз - после полного зеркалирования каталога **distfiles** с сайта **Gentoo**, и за казенный счет, разумеется :-)

### **Этап четвертый: в недра ядра**

На предыдущем этапе были перестроены все компоненты базовой системы, кроме одного - зато самого главного: ядра системы. Перекомпиляция ядра - процедура, рано или поздно неизбежная для пользователя любого дистрибутива Linux, даже самого user-ориентированного. Перед пользователем же Source Based дистрибутива она встает с неотвратимостью рока уже при установке системы.

**Gentoo** - опять не составляет исключения. Хотя в заключительных разделах я расскажу, как при крайней необходимости можно обойти этот этап, однако по хорошему к тому нет никаких причин. И потому, хотя тема пересборки ядра - совершенно отдельная, и нам придется еще не раз к ней возвращаться, применительно случаю постараюсь осветить ее здесь в минимально возможном объеме.

Чтобы пересобрать ядро, для начала нужно иметь его исходники. Ни на инсталляционном диске, ни в составе тарбаллов **stage2-3** мы их не найдем. Что делать?

При подключении к Сети - все просто: к нашим услугам система портежей. Отправляемся в каталог **/usr/portage**, просматриваем содержимое подкаталога **sys-kernel** - и остается только дивиться свалившемуся на голову богатству: даже отметая классово чуждые портежи ядер для Sparc, PPC, Alpha, на выбор



предоставляется чуть не дюжина вариантов. Подчеркну, речь идет не о прекомпилированных ядрах, а именно о вариантах исходников, снабженных патчами, обеспечивающими дополнительные опции относительно ядра перевозданного.

Разумеется, и последнее, каноническое ядро от Линуса Торвальдса - тоже доступно (под известным именем **vanilla-sources**) для тех, кто хочет быть святее папы римского. Далее - фирменная коронка, **gentoo-sources**, это - лучший выбор с точки зрения функциональности). Ну и as-sources - ядро от Алана Кокса, в патчах которого обычно воплощены новейшие достижения ядреной науки и техники.

Есть и более специфичные варианты: **usermode-sources** - ядро для запуска Linux в Linux, в т.н. пользовательской моде (предназначается обычно для тестирования); **openmosix-sources** - ядро с поддержкой одноименной (**openMosix**) кластерной технологии, и т.д..

Так что в этих условиях главное - определиться с выбором. А потом - простая команда

**\$ emerge sys-kernel/name-sources**

некоторый период ожидания, пока скачиваются сами исходники ядра и все необходимые патчи, еще чуть-чуть времени на развертывание архивов - и вперед, в каталог **/usr/src/linux** для конфигурирования.

При наличии предусмотрительно скачанных исходников ядра, наиболее явно необходимых патчей и модемного подключения - сложнее, но не намного. Выполняем ту же команду. Если повезет - все по прежней схеме, но предупреждаю, шансов на это мало: почти наверняка при предварительном скачивании какой-то Gentoo-специфичный патч будет пропущен (учтем также наше пока еще очень ограниченное знакомство с системой портежей вообще - в дальнейшем такой ситуации можно и избежать). Но не беда: получив сообщение об ошибке, соединяемся с Сетью и либо скачиваем недостающий патч вручную (хоть через браузер **lynx** - полный адрес к недостающему компоненту будет выведен), либо (и это, товарищи, правильнее) запускаем процедуру по новой. Разорвав соединение по завершении установки исходников ядра.

И не нужно бояться, что это больно ударит по карману: при соединении по линии, способной вынести протокол V90, скачивание большинства патчей занимает минуты. И **wget** при обрыве коннекта способен к докачке - так что непроизводительные расходы сведены к минимуму.

Хуже, если нет ни полного комплекта исходников, ни модема. И это не смертельно, но придется кое-чем поступиться (надеюсь, не принципами). То есть: просто тупо распаковываем архив исходников ядра в каталог **/usr/src**, распаковываем и накладываем минимально необходимые патчи, далее - обычным порядком, как описано ниже. Можно и просто ограничиться каноническим ядром Линуса. Недостаток - кроме возможных потерь некоторых функций, эти действия не будут зафиксированы в базе данных портежей (имеющей место пребывания в каталоге **/var/db/pkg**), то есть система портежей о них как-бы и не узнает (на самом деле - не совсем, но все равно - этот способ может рассматриваться как временное решение).

И еще одно временное решение - вообще отказаться от пересборки ядра. А как при этом получить возможность загрузить систему - я расскажу под занавес (если вы к тому времени сами не догадаетесь).

В любом случае, кроме последнего, исходники ядра мы получили, так что можно отправляться в вышеупомянутый каталог **/usr/src/linux** (или **/usr/src/linux-номер\_версии-имя\_варианта**, без разницы, - первый лишь символическая на него ссылка). И давать команду

```
$ make menuconfig
```

которая сгенерирует меню настройки опций ядра. Любители отвечать черному экрану могут выбрать и **make config** - но я к ним не принадлежу.

Сам процесс конфигурирования описывать не буду - это совсем другая история. Отмечу только обязательные опции. Первая из них - поддержка экспериментальных и разрабатываемых опций: большинство продвинутых возможностей ядра Linux описываются именно этими терминами (что не мешает им функционировать вполне справно). В секции файловых систем необходимо отметить поддержку файловых систем процессов (**procfs**), файловой системы виртуальной памяти (Virtual Memory Filesystem) и файловой системы устройств (**devfs**). И, конечно же, тех файловых систем, которые были размещены на созданных разделах (например, XFS или **ext3fs** - про поддержку **ext2fs**, надеюсь, не забудет ни один линуксоид). Ну и учесть поддержку требуемых устройств - сами знаете, что у вас вкручено в машине.

Закончив с конфигурированием, выходим из системы меню (не забыв сохранить изменения) и даем последовательность команд:

```
$ make dep
```

```
$ make clean
```

```
$ make bzImage
```

```
$ make modules
```

```
$ make modules_install
```

Разумеется, не обязательно последовательно, можно и одной строкой, разделяя символом ; (или, правильнее, &&). А то и вообще в форме, рекомендованной руководством:

```
$ make dep && make clean bzImage modules modules_install
```

Теперь остается только скопировать образовавшийся образ ядра в загрузочный каталог

```
$ cp /usr/src/linux/arch/i386/boot/bzImage /boot
```

И ядерный вопрос можно считать закрытым. Хотя остается еще вопрос, косвенным образом с ним связанный - программы поддержки файловых систем. Их, в соответствие с выбором и предпочтениями, устанавливаем через систему портежей, например, команда

```
$ emerge sys-apps/xfsprogs
```

установит все необходимое для работы с файловой системой **XFS**. Напомню, что архив исходников пакета **xfsprogs** должен иметься в каталоге **distfiles**.

Попутно через систему портежей можно установить и другие программы, однако я пока ограничился бы только тем, что облегчит завершение конфигурирования, а именно - службой консольной мыши:

```
$ emerge sys-libs/gpm
```

А теперь активизируем ее. Если мы имеем наиболее типичного ныне грызуна с колесиком и под разъем PS/2, это будет выглядеть так:

```
$ gpm -t imps2 -m /dev/psaux
```

И на следующих этапах гораздо проще будет выделять нужные фрагменты и вставлять их щелчком средней клавиши, нежели набрать вручную.

Финал: постинсталляционное конфигурирование

Собственно установка на этом закончена. Осталось довести систему до идеала не только изнутри, но и снаружи. И первый шаг здесь - настройка часового пояса: для Source Based дистрибутивов это не последнее дело (утилита управления компиляцией **make** обращает внимание на даты создания файлов). Реализуем просто: созданием символической ссылки

```
$ ln -sf /usr/share/zoneinfo/path/timezonefile /etc/localtime
```

Рискну предположить, что компонент пути path/timezonefile для большинства моих читателей окажется в диапазоне от Europe/Kaliningrad до Asia/Kamchatka (наличие Анадыря в нашей стране почему-то упорно игнорируется - но, слава богу, хоть о Петропавловске-Камчатском знают). Если системное время в BIOS установлено на не время по Гринвичу, дополнительно потребуется внести коррективы в файл /etc/rc.conf.

Теперь увековечим монтирование новых файловых систем. Как? - очень просто, лобовым редактированием ответственного за сие файла **/etc/fstab**. Вызываем его:

```
$ nano -w /etc/fstab
```

И видим, если исключить комментарии, следующую картину:

/dev/BOOT	/boot	ext2	noauto,noatime	1 1
/dev/ROOT	/	xfs	noatime	0 0
/dev/SWAP	none	swap	sw	0 0
/dev/cdroms/cdrom0	/mnt/cdrom	iso9660	noauto,ro	0 0
proc	/proc	proc	defaults	0 0
#tmpfs	/dev/shm	tmpfs	defaults	0 0

Если мы следовали рекомендациям инструкции, остается только заменить символы верхнего регистра конкретными реалиями, как то: **/dev/hda1**, **/dev/hda3**, **/dev/hda2**. Или, что будет более корректно, **/dev/discs/disc0/part1** и т.д., сняв при этом комментарий со строки **#tmpfs**. Если же использовалась какая-либо иная схема разбиения - можно в одной из ранее активизированных консолей с помощью команды **mount** без аргументов подсмотреть, какой раздел с какой файловой системой куда монтировался. И просто перетащить мышью (не зря же мы ее запускали) соответствующие значения в наш **/etc/fstab**. Ну и всякого рода коррективы (типа монтирования FAT-разделов, флоппи-дисков, zip-накопителей и еще чего требуется) также никто не запрещает.

Внимательный читатель обратил внимание - в строке **/dev/BOOT** по умолчанию стоит значение **noauto**, предписывающее не монтировать соответствующий раздел автоматически при старте системы. Именно так рекомендуется делать при использовании **GRUB** - он принят в **Gentoo** в качестве загрузчика по умолчанию. Так вот, настало время обеспечить загрузку системы. Для этого даем команду **grub** и после появления одноименного приглашения последовательно вводим:

```
grub> root (hd0,0)
grub> setup (hd0)
```

вслед за чем выходим из программы:

**grub> quit**

Начальный загрузчик установлен в первый раздел первого диска, тот самый, который в текущий момент смонтирован у нас как **/boot**. И который в дальнейшем не будет монтироваться автоматически - дополнительная защита, в первую очередь от человеческого фактора. Так что в дальнейшем не следует удивляться, если, собрав новое ядро системы, вы не сможете его сразу скопировать в каталог **/boot** - предварительно он придется смонтировать вручную.

Теперь остается **GRUB** настроить. Для этого посредством того же редактора создаем файл его меню:

**\$ nano -w /boot/grub/menu.lst**

и записываем в него две секции. Строки первой относятся к **GRUB** в целом:

**default 0**

**timeout [значение\_в\_секундах]**

**splashimage=(hd0,0)/boot/grub/splash.xpm.gz**

Она указывает на ОС, загружаемую по умолчанию, время ожидания выбора в секундах и описывает путь к файлу графической заставки. Последняя штатно - нечто вроде аметиста с морионовым оттенком. Не нравится - заставку можно убрать (стерев или закомментировав третью строку). Или - изготовить собственную соответствующем формате.

Вторая секция описывает непосредственно загрузку **Gentoo**.

**title=My Love Gentoo Linux**

**root (hd0,0)**

**kernel /boot/имя\_образа root=/dev/hda3**

Первая строка - произвольное название пункта меню. Вторая определяет имя устройства, с которого будет загружаться система. Имя это дается в нотации **GRUB**, отличной от принятой в Linux: дело в том, что **GRUB** способен загрузить ядро с не смонтированной файловой системы, относящейся к незагруженной операционке. И это распространяется не только на Linux, но на практически любую ОС из клана свободных Unix'ов (например, на FreeBSD или OpenBSD).

В третьей строке определяется имя образа ядра, подлежащего загрузке, и устройство, на котором после этого будет располагаться корневая файловая система. Если мы собирали собственное ядро, в качестве образа будет выступать тот самый файл **/boot/bzImage**, который был скопирован в каталог **/boot** после компиляции. Если же ядро не собиралось, значение **kernel**, как будет показано ниже, проще всего определить как **/boot/linux**. А имя устройства с корневым каталогом на этот раз дается в нотации Linux - ведь соответствующее ядро уже загружено, и оно знает, что такое **/dev/hda?**.

Напомню еще раз, что все действия по редактированию меню GRUB можно выполнить, просто перетаскивая мышью соответствующие фрагменты руководства (там приведены все необходимые примеры) в наш **/boot/grub/menu.lst** и редактируя их в соответствии с собственными реалиями.

И все же - супер финал

Настройкой загрузчика можно считать завершенной официальную часть инсталляции. Однако для полного счастья не хватает еще некоторых штрихов. Так, руководство рекомендует (и это - более чем резонно) оптимизировать режим работы

жесткого диска на предмет использования DMA - что весьма способствует его быстродействию. Для такого рода штук предназначен файл **/etc/conf.d/local.start**. Открываем его в редакторе и после комментария (по умолчанию там больше ничего нет) приписываем строку  
**hdparm -d 1 /dev/hda**

Если физических дисков больше одного - повторяем ее для каждого устройства. А затем, воспользовавшись случаем, обеспечим здесь же и загрузку консольной мыши при старте системы:

**gpm -t imps2 -m /dev/psaux**

Теперь обратимся к локализации. Не смотря на свое исходно американское происхождение, **Gentoo** - дистрибутив интернациональный (это ясно из состава команды его разработчиков). И потому в нем есть все базовые средства для работы с языками, отличными от американского. Нужно только немного потрудиться для их активизации.

Достигается это можно редактированием файла **/etc/rc.conf**. Что может показаться необычным для пользователей Linux. Но мы помним, что в Gentoo принят порядок начальной загрузки в стиле BSD (а не в стиле System V, как в большинстве распространенных дистрибутивов Linux, за исключением Slackware и его сородичей). Впрочем, вдаваться в подробности пока не буду - это тема для отдельной беседы. Пока же - голая практика: все, что мы хотим видеть загруженным при старте машины, должно быть явным образом указано в файле **/etc/rc.conf**. И именно в том порядке, в каком требуется.

Так что открываем этот универсальный файл

**\$ nano -w /etc/rc.conf**

отыскиваем в начале его строку

**KEYMAP="us"**

и заменяем чуждую нам Америку на что-нибудь родное, от братчиков Кирилла и Мефодия. Что именно - определяем в какой-либо из иных виртуальных консолей командой

**\$ ls /usr/share/keymaps/i386/qwerty/ru\***

В ответ она выведет на экран такой список:

**/usr/share/keymaps/i386/qwerty/ru1.map.gz**  
**/usr/share/keymaps/i386/qwerty/ru2.map.gz**  
**/usr/share/keymaps/i386/qwerty/ru3.map.gz**  
**/usr/share/keymaps/i386/qwerty/ru4.map.gz**  
**/usr/share/keymaps/i386/qwerty/ru-cp1251.map.gz**  
**/usr/share/keymaps/i386/qwerty/ru.map.gz**  
**/usr/share/keymaps/i386/qwerty/ru-ms.map.gz**  
**/usr/share/keymaps/i386/qwerty/ru\_win.map.gz**  
**/usr/share/keymaps/i386/qwerty/ru-yawerty.map.gz**

Можно видеть, что в нем присутствуют кириллические раскладки на все случаи жизни (а пути к ним, хотя и взяты с Live CD, соответствуют тем, что будут в инсталлированной системе). Дело, конечно, сугубо личное (я бы даже сказал, интимное), но по мне предпочтительна ru4: это раскладка для кодировки **KOI8-R** с Windows-маркировкой клавиш (не путать с собственно раскладкой для Windows-кодировки - ее, как видно из списка, тоже можно использовать) и переключением с

латиницы на кириллицу по нажатию CapsLock. Так что вписываем  
**KEYMAP="ru4"**

и переходим к следующей строке, определяющей экранный шрифт консоли - **CONSOLEFONT**. Очевидно, что умолчальное ее значение описывает шрифт без символов кириллицы. Но - их у нас есть, в чем легко убедиться, просмотрев каталог **/usr/share/consolefonts**. Шрифтов там немерянно, и среди них легко отыскиваются семейства **alt-\*.psf.gz** и **cp866-\*.psf.gz**. И то, и другое нам (почти) подходит - это экранные шрифты с символами кириллицы в кодировке **CP866** (она же - альтернативная, кодировка DOS или IBM). Почему так - ведь мы только что установили раскладку **KOI8-R** для клавиатуры, да и шрифты (семейства **koi8r-\*.gz**), имеются? Распространяться не буду, ограничившись словами: это лучше, Ленья.

А слово "почти" я добавил потому, что все имеющиеся стандартно кириллические шрифты далеки от совершенства как эстетически, так и медицински. Так что в дальнейшем лучше будет воспользоваться шрифтами из пакета Виктора Вагнера **console-tools-cyrillic** - там есть очень симпатичные экземпляры, но об этом - как-нибудь в другой раз. Пока же просто вписываем, например,  
**CONSOLEFONT="cp866-8x16"**

И начинаем рассуждать логически: как же так, раскладка клавиатуры **KOI8**, а кодировка экранного шрифта - **CP866**. Ясно, что без загрузки карты трансформации одного в другое не обойтись. И такая карта у нас есть тоже - в каталоге **/usr/share/consoletrans** обнаруживаем файл **koi2alt**. Имя которого и вписываем в строку  
**CONSOLETRASLATION="koi2alt"**

сняв с нее предварительно стоящий по умолчанию комментарий. Откладываем в уме, что таким образом карта соответствия будет загружена только на одной, первой, из виртуальных консолей (она же - консоль системная) - к этому мы еще вернемся. А пока, дабы покончить с файлом **/etc/rc.conf**, вспоминаем, что системные часы нашей машины установлены по местному часовому поясу (скорее всего - хотя в последнее время я осознал преимущества времени гринвичского) - тому самому, который мы определили через файл-ссылку **/etc/localtime**. И обращаем внимание, что строка  
**CLOCK="UTC"**

указывает у нас, напротив, именно на время по Гринвичу (UTC=GMT). Заменяем UTC на localtime - и файл **/etc/rc.conf** можно закрывать (прочие его строки можно оставить без изменений, если вам лично не известны веские причины их сделать).

А вот теперь пора довести до конца дело с картами соответствия. Открываем файл **/etc/init.d/local**: как явствует из названия, он предназначен для всякой всячины местного значения (и исполняется после всех остальных конфигурационных файлов). И вставляем в его конец с помощью редактора (или перетаскиваем мышью из данного руководства) простенький скрипт (оператор цикла):

```
for i in 1 2 3 4 5 6
do
    echo -ne ">" > /dev/vc/$i
done
```

В первой его строке определяется значение переменной **i** (ее можно обозвать любым другим образом) как диапазона цифр от 1 до 6 (по умолчанию у нас именно шесть виртуальных консолей), вторая - указывает, что следующей строкой будет

тело цикла. А тело цикла - вывод "магической последовательности" ESC-(K (именно она и обеспечивает активизацию карты соответствия) и ее перенаправление на виртуальные консоли, номера которых заданы переменной *i*, то есть `/dev/vc/1` - `/dev/vc6`. Это - если используется файловая система устройств (*devfs*), в противном случае (если не был снят комментарий со строки `tmpfs` в файле `/etc/fstab`) имена консолей будут обычными - `/dev/tty0` - `/dev/tty5`.

Отступление для любителей (или невольников) кодировки **CP1251**. Все сказанное о русификации применимо и к ней. Нужно только определить в файле `/etc/fstab` загрузку соответствующих раскладки клавиатуры (они, как уже говорилось, имеются) и карты соответствия **CP1251->CP866**. А вот ее придется поискать (где - за ненадобностью не помню) или просто слямзить из любого отечественного дистрибутива - Altlinux или ASPLinux.

Все же открою секрет - почему в консоли в любом случае предпочтительны экранные шрифты в кодировке DOS. Так уж исторически сложилось: вследствие аппаратных особенностей VGA-адаптеров (а в консоли мы имеем дело именно с режимом VGA, сколь бы крутой GeForce не стоял в машине) только они обеспечивают нормальный вывод на экран символов псевдографики. Кроме того, их просто больше, и некоторые среди них, как я говорил, лучше. Так что есть резон загружать именно их, даже если программы с псевдографикой не используются совсем. В том числе и **Midnight Commander** - но какой же русский не любит детей командира Нортон!

Мы же пока завершим русификацию. Для чего установим собственно локаль последней строкой в файле `/etc/init.d/local` (опять прекрасный каламбур, господа!):  
**export LANG="ru\_RU.KOI8-R"**

Она обеспечит установку всех локально-зависимых переменных (не только страны, языка, набора символов, но и формата даты, времени, денежной единицы, и так далее) в значения, принятые для нашей многострадальной Отчизны. Тем не менее, для представления чисел я сделал бы исключение, добавив строку  
**export LC\_NUMERIC="POSIX"**

Ведь в любых программах, связанных со счетом, удобнее оперировать с десятичной точкой, а не запятой (а говорят, что некоторые такие программы этого просто требуют).

В итоге, выполнив несложную последовательность действий, мы получили корректно русифицированную систему - русифицированную ничуть не хуже, чем те дистрибутивы, кириллической поддержкой в которых гордятся (и - справедливо гордятся) именитые отечественные разработчики - Altlinux и ASPLinux. А поскольку мы сделали все это своими руками и с полным пониманием сути действий, у нас тоже есть основание для гордости. Правда, "лобовым" способом, без учета специфики дистрибутива - но зато работающим всегда и везде. А к "идеологически правильному" (для **Gentoo**) методу локализации можно будет вернуться позднее.

Можно, конечно, активизировав внесенные в конфигурацию изменения командой `etc-update`, продолжать устанавливать программы (Иксы, например). Или даже начинать обычную работу - установленная система, внутри которой мы находимся, это вполне уже позволяет. Однако для полной проверки лучше все-таки сначала перезагрузиться.

Итак, временно расстаемся с нашим новым Gentoo командой `exit`, после чего вновь

оказываемся в системе Live CD, той же командой зарываем все виртуальные консоли, кроме первой (а у нас их могло расплодиться немало) и готовимся к выходу.

Хотя - стоп. А собирали ли мы при установке ядро системы. Если да - все в порядке, тотальный **umount**:

```
$ umount /mnt/gentoo/boot
```

```
$ umount /mnt/gentoo
```

и что было смонтировано руками, а затем - **reboot** (или **halt**, если есть желание прерваться - при первой загрузке хорошо бы внимательно следить за ее ходом по сообщениям на экране).

Если же ядро не собиралось (за отсутствием каких-то исходников - другого оправдания такому разгильдяйству нет), то совершаем последнее установочное действие: просто тупо копируем ядро с Live CD (оно, как мы уже убедились, не только работоспособно, но и поддерживает большинство жизненно необходимых функций):

```
$ cp /cdroot/boot/linux /mnt/gentoo/boot/
```

Правда, у нас не собиралось не только ядро, но и модули. И это не беда, их тоже копируем с CD:

```
cp -R /cdroot/lib/modules /mnt/gentoo/lib/modules
```

А далее - все то же, размонтирование и перезагрузка. Конечно, в ходе ее мы рискуем получить кое-какие сообщения об ошибках, но, по моему опыту - ни одна из них не критична. И загрузившаяся система будет вполне пригодной к использованию. По крайней мере, до тех пор, пока не разживемся всеми необходимыми архивами и все-таки ядро не пересоберем - от этого все равно не уйти.

Заключительный аккорд

У терпеливого читателя, добравшегося до этих строк, может возникнуть три совершенно законных вопроса: не слишком ли все это сложно, на кой все это нужно, и кому все это нужно. Постараюсь ответить последовательно.

Относительно сложности - могу заверить, что это из той самой оперы, когда глаза боятся, а руки делают. Проверено на собственном опыте - причем первый из них проводился в подмосковной деревне, где не то что постоянного подключения к Сети - и телефона-то с модемом не было. И все исходники скачивались на службе (в полутора часах езды) и таскались домой на CD и съемном винчестере. И если праведно выполнять все заветы инструкции по установке, то сложностей возникнет не много (надеюсь, что и настоящий мемуар поспособствует их преодолению).

Второй вопрос - а зачем их было преодолевать, когда можно легко и быстро развернуть любой пакетный дистрибутив, разработчики которого уже решили все мои проблемы. Отвечаю: решить-то они решили, но - не скажу плохо, просто в меру своих представлений, как это нужно народу (и конкретно мне, как его представителю). **Gentoo** же позволяет не просто поставить систему, идеально оптимизированную под железо, но и настроить ее тоже в соответствии именно со своим представлением об идеале. Причем - с полным осознанием, что и зачем делается, немало способствующим общему образованию.

Ну а третий ответ очевиден из второго. Дистрибутив Gentoo нужен пользователю, которому требуется система, настроенная на максимальную производительность



для конкретной тяжелой задачи - счетной, графической, GIS или CAD. Разработчику, которому надоело проводить время компиляции тысяч и тысяч строк кода за игрой в тетрис. Сисадмину, у которого в системе не должно быть ничего лишнего, но все необходимое должно быть. Изучающему Linux, которого не устраивают готовые рецепты, предлагаемые руководствами пакетных дистрибутивов. И наконец, тому, кто просто стремится к идеалу - пусть идеал этот и недостижим...

## Краткое руководство по установке Gentoo Linux x86

Ссылка на оригинал: <http://www.gentoo.org/doc/ru/gentoo-x86-quickinstall.xml>

С версии: 1.0

### Установочный диск

Скачайте компакт-диск с одного из наших зеркал. ISO-образы находятся в каталогах: **releases**/**<архитектура>**/**<выпуск>**/установочный-диск. Минимальный установочный диск подходит только для установки из интернета; с универсального диска также возможна установка без обращения к сети.

Запишите компакт-диск, затем загрузитесь с него. В загрузочном окне нажмите F2, чтобы выяснить, какие варианты загрузки предлагаются. Если вам нужна поддержка РСМСІА, сразу после загрузки потребуется запустить сценарий инициализации `rcmciа`.

Установочные диски позволяют запускать сервер `sshd`, добавлять новых пользователей, запускать `irssi` (консольный чат-клиент) и просматривать веб-страницы с помощью `lynx` или `links2`.

### Настройка сети

Если ваша сеть еще не работает, для ее настройки можно использовать **net-setup**. Перед настройкой вам может потребоваться загрузить поддержку своей сетевой платы с помощью `modprobe`. Если у вас ADSL, используйте **adsl-setup** и **adsl-start**. Для поддержки PPTP, сначала отредактируйте `/etc/ppp/chap-secrets` и `/etc/ppp/options.pptp`, а затем запустите `pptp <ip сервера>`.

При беспроводном доступе для установки параметров соединения используйте **iwconfig**, а затем либо опять **net-setup**, либо запустите вручную `ifconfig`, `dhcpcd` и/или `route`.

Если вы находитесь за прокси-сервером, не забудьте инициализировать систему с помощью `export http_proxy, ftp_proxy` и `RSYNC_PROXY`.

### Подготовка дисков

Для организации разделов запустите **fdisk** или **cfdisk**. Вам нужен хотя бы раздел подкачки (тип 82) и один раздел Linux (тип 83).

Для создания файловых систем на своих Linux-разделах используйте **mke2fs**, **mke2fs -j**, **mkreiserfs**, **mkfs.xfs** и **mkfs.jfs**. Подготовьте свой раздел подкачки командами **mkswap** и **swapon**.

Смонтируйте свежесозданные файловые системы в `/mnt/gentoo`. Создайте каталоги

для других точек монтирования (например, **/mnt/gentoo/boot**), если они вам нужны.

## Установка стадии

Прежде всего, установите верную дату командой **date MMDDhhmmYYYY**. Далее скачайте файл стадии с одного из наших зеркал или используйте находящийся на установочном диске (в **/mnt/cdrom/stages**). Перейдите в **/mnt/gentoo** и распакуйте стадию командой **tar -xvjp <архив стадии>**.

В случае установки без сети установите снимок портежей: перейдите в **/mnt/gentoo/usr** и запустите **tar -xvjp /mnt/cdrom/snapshots/<файл снимка>**. Другие пользователи таким же образом могут скачать снимок портежей и установить его.

При установке без сети, скопируйте файлы исходников из **/mnt/cdrom/distfiles/** в **/mnt/gentoo/usr/portage/distfiles/**.

Отредактируйте **/mnt/gentoo/etc/make.conf** под ваши нужды (флаги **USE**, **CFLAGS** и **CXXFLAGS**). Для этого можно использовать редактор **nano**.

## Установка базовой системы Gentoo

Сначала смонтируйте файловую систему **/proc**, скопируйте файл **/etc/resolv.conf** поверх существующего, а затем смените корневой каталог (**chroot**) на свою среду **Gentoo**.

**Листинг 1.1:** Подготовка и смена корневого каталога

```
# mount -t proc none /mnt/gentoo/proc
# cp /etc/resolv.conf /mnt/gentoo/etc/
# chroot /mnt/gentoo /bin/bash
# env-update && source /etc/profile
```

Если вы не выполняете бессетевую установку, командой **emerge --sync** обновите свое дерево портежей.

Далее убедитесь, что **/etc/make.profile** указывает на верный профиль. Профиль по умолчанию должен подойти большинству пользователей; имеются подпрофили для различных ядер (например, 2.4/ для профилей, основанных на ядре 2.4). Изменяйте профиль командой **ln -sf**.

Самогенерация системы (не выполняется при бессетевой установке) запускается сценарием **scripts/bootstrap.sh** в каталоге **/usr/portage**.

Установка системы (не выполняется при бессетевой установке) запускается командой **emerge -e system** (или **-N**, если вы не меняли значения **CFLAGS/CXXFLAGS** по умолчанию).

## Настройка ядра

Установите сведения о своем часовом поясе, сменив символическую ссылку **/etc/localtime** на указывающую на верное место в **/usr/share/zoneinfo** командой **ln -sf**.

Установите исходный код ядра (при установке без сети доступны **gentoo-sources** и **vanilla-sources**) и настройте его с помощью **make menuconfig**, затем **make && make modules\_install** внутри каталога **/usr/src/linux**. Скопируйте файл **arch/i386/boot/bzImage** в **/boot**, заменяя существующий. Также можно выполнить **emerge genkernel** и **genkernel all**.

Пользователям, использующим **Genkernel**, потребуется выполнить **emerge coldplug** и **rc-update add coldplug default**.

## Настройка системы

Отредактируйте свой **/etc/fstab**; пример показан ниже.

**Листинг 1.2:** Пример файла **fstab**

<b>/dev/hda1</b>	<b>/boot</b>	<b>ext2</b>	<b>defaults,noatime</b>	<b>1 2</b>
<b>/dev/hda2</b>	<b>none</b>	<b>swap</b>	<b>sw</b>	<b>0 0</b>
<b>/dev/hda3</b>	<b>/</b>	<b>ext3</b>	<b>noatime</b>	<b>0 1</b>
<b>none</b>	<b>/proc</b>	<b>proc</b>	<b>defaults</b>	<b>0 0</b>
<b>none</b>	<b>/dev/shm</b>	<b>tmpfs</b>	<b>nodev,nosuid,noexec</b>	<b>0 0</b>
<b>/dev/cdroms/cdrom0</b>	<b>/mnt/cdrom</b>	<b>auto</b>	<b>noauto,user</b>	<b>0 0</b>

Для настройки своей сети отредактируйте **/etc/conf.d/hostname** и **/etc/conf.d/domainname**, запустите **rc-update add domainname default**, затем отредактируйте **/etc/conf.d/net**. Добавьте сценарий инициализации **net.eth0** к уровню запуска по умолчанию (**default run level**). Если у вас несколько сетевых карт (NIC), поместите символические ссылки на них в сценарий инициализации **net.eth0**, и также добавьте их в уровень запуска по умолчанию.

Отредактируйте **/etc/hosts**; примеры приведены ниже:

**Листинг 1.3:** Примеры файла **/etc/hosts**

(Для статических IP-адресов)

**127.0.0.1 localhost**

**192.168.0.5 jenny.homenetwork jenny**

**192.168.0.6 benny.homenetwork benny**

**192.168.0.7 tux.homenetwork tux**

(Для динамических IP-адресов)

**127.0.0.1 localhost.homenetwork tux localhost**

При необходимости установите (**emerge**) **pcmcia-cs** и добавьте его в уровень запуска по умолчанию.

Установите пароль **root** с помощью **passwd**.

Выполните необходимую настройку в файлах: **/etc/rc.conf**, **/etc/conf.d/rc**, **/etc/conf.d/keymaps**, **/etc/conf.d/clock**.

## **Установка системных утилит**

Пользователям ядра 2.4 нужно запустить **emerge --unmerge udev** и **emerge devfsd**.

Установите системный журнал, например **syslog-ng**, и добавьте его в уровень запуска по умолчанию. Сделайте то же самое для демона **cron**, например, **vixie-cron** (необязательный шаг).

Установите необходимые средства для файловых систем (**xfsprogs**, **reiserfsprogs** или **jfsutils**), а также для сети (**dhcpcd** или **rp-pppoe**).

## **Настройка загрузчика**

Установите (**emerge**) **grub** или **lilo**. Отредактируйте **/boot/grub/grub.conf** или **/etc/lilo.conf**, соответственно. Ниже приведены примеры обоих.

### **grub**

*Листинг 1.4:* Пример **grub.conf**

**default 0**

**timeout 30**

**splashimage=(hd0,0)/grub/splash.xpm.gz**

**# для пользователей genkernel**

**title=Gentoo Linux 2.6.11-r3**

**root (hd0,0)**

**kernel /kernel-genkernel-x86-2.6.11-gentoo-r3 root=/dev/ram0 init=/linuxrc ramdisk=8192**

**real\_root=/dev/hda3 udev**

**initrd /initramfs-genkernel-x86-2.6.11-gentoo-r3**

**# для остальных пользователей (без initrd)**

**title=Gentoo Linux 2.6.11 r3**

**root (hd0,0)**

**kernel /kernel-2.6.11-gentoo-r3 root=/dev/hda3**

**# только если вам нужна двойная загрузка**

**title=Windows XP**

**root (hd0,5)**

**makeactive**  
**chainloader +1**

**lilo**

**Листинг 1.5:** Пример файла **lilo.conf**

**boot=/dev/hda**  
**prompt**  
**timeout=50**  
**default=gentoo**

**# для пользователей, не использующих genkernel**

**image=/boot/kernel-2.6.11-gentoo-r3**  
**label=gentoo**  
**read-only**  
**root=/dev/hda3**

**# для пользователей genkernel**

**image=/boot/kernel-genkernel-x86-2.6.11-gentoo-r3**  
**label=gentoo**  
**read-only**  
**root=/dev/ram0**  
**append="init=/linuxrc ramdisk=8192 real\_root=/dev/hda3 udev"**  
**initrd=/boot/initramfs-genkernel-x86-2.6.11-gentoo-r3**

**# Для двойной загрузки**

**other=/dev/hda6**  
**label=windows**

Пользователям **GRUB** необходимо установить **GRUB** в главную загрузочную запись (**MBR**) командой **grub-install /dev/hda** после копирования **/proc/mounts** в **/etc/mtab**. Пользователям **LILO** нужно запустить **/sbin/lilo**.

Выйдите из изолированной среды, размонтируйте все файловые системы и перезагрузитесь.

## **Окончание установки**

Зайдите как **root**, затем добавьте одного или нескольких пользователей для повседневного использования командой **useradd -m -G <группа> <имя пользователя>**.

Если вы выполняли установку без сети, смонтируйте компакт-диск с пакетами в **/mnt/cdrom** и выполните **export PKGDIR="/mnt/cdrom"**, после чего вы сможете

устанавливать дополнительное ПО, например, **kde**, командой **emerge -k <имя пакета>**. для .

Спасибо за установку Gentoo!

## Описание альтернативных способов установки Gentoo Linux

Ссылка на оригинал: <http://www.gentoo.org/doc/ru/altinstall.xml>

С версии: 1.0

### 1. Об этом документе

Если у вас не работает стандартный метод установки **boot-from-CD** (или он вас не устраивает), вам сюда. Этот документ описывает альтернативные приемы установки Gentoo Linux для тех, кто в них нуждается. Или, если хотите, это место сбора всех "оригинальных" способов установки. Если у вас один такой завалялся, или вы откопали новую прикольную методу, не стесняйтесь, написать запрос в Bugzilla.

### 2. Загрузка с установочного диска при помощи Smart BootManager

Скачайте **Smart BootManager** с <http://btmgr.sourceforge.net/download.html>. Доступны как исходники, так и бинарники для Linux, не говоря уже про виндовский .exe и множество языковых пакетов. Все-таки, в настоящее время, бинарный формат предпочтительнее, поскольку исходники могут не откомпилироваться современными версиями NASM.

Скомпилируйте пакет или возьмите уже готовый. Ниже я расскажу о некоторых полезных опциях, доступных при создании загрузочной дискеты.

**Листинг 2.1:** Опции Smart BootManager

**sbminst [-t theme] [-d drv] [-b backup\_file] [-u backup\_file]**

-t тема     выбор используемой темы, где тема должна быть:  
us = English тема     de = German тема  
hu = Hungarian тема     zh = Chinese тема  
ru = Russian тема     cz = Czech тема  
es = Spanish тема     fr = French тема  
pt = Portuguese тема

-d диск     выбор диска, куда вы собираетесь инсталлировать Smart BootManager;  
для Linux:  
   /dev/fd0 первый floppy-диск,  
   /dev/hda первый IDE-диск.  
   /dev/sda первый SCSI-диск.  
для DOS:

0 первый floppy-диск  
128 первый жесткий диск;

- c отключить загрузку с CD-ROM;
- b backup\_file сохранить (для будущей деинсталляции) данные, которые будут перезаписаны;
- u backup\_file деинсталлировать Smart BootManager, нельзя использовать с другими ключами;
- y никаких вопросов и предупреждений.

**Листинг 2.2:** Использование sbminst для создания загрузочной дискеты  
# **sbminst -t us -d /dev/fd0**

Примечание: Замените fd0 на имя вашего floppy-дисковода (если у вас он называется по-другому).

Теперь просто вставьте дискету в дисковод компьютера, на котором вы хотите запустить установочный диск, и загрузитесь с нее.

Вас поприветствует Smart BootManager. Выберите CD-ROM и нажмите ENTER для загрузки установочного диска. Если все прошло удачно, следуйте стандартной процедуре установки.

Больше информации по Smart BootManager вы найдете на <http://btmgr.sourceforge.net/>

### 3. Установка при помощи Knoppix

Примечание: Knoppix доступен только для пользователей x86.

Загрузившись с Knoppix LiveCD, вы получаете полностью работоспособную систему для компиляции Gentoo. Во время сборки Tux Racer скрасит ваше ожидание, а для работы вы можете пользоваться OpenOffice.

Предупреждение: Помните, что если вы что-либо сохраните в домашний каталог Knoppix, ожидая установки своей системы Gentoo, вы потеряете это после перезагрузки в Gentoo. Убедитесь, что сохраняете важные файлы на жесткий диск или на другой компьютер!

Загрузитесь с Knoppix CD. Если у вас Knoppix 3.6-3.8.2, потребуется указать knoppix26 в качестве параметра ядра, чтобы загрузить ядро 2.6. Если вы пропустите этот шаг, то при chroot вы получите сообщение об ошибке, что ваше ядро слишком старое. Однако, если у вас Knoppix 3.9+, этот шаг не потребуется, так как ядро 2.6 выбрано по умолчанию.

По умолчанию вы попадаете на рабочий стол KDE. Откройте консоль и введите su -, чтобы была возможность изменить пароль. Это позволяет задать пароль для root в Knoppix. Теперь, при желании, можете настроить sshd для удаленного входа в систему.

**Листинг 3.1:** Создание точки монтирования /mnt/gentoo



```
# mkdir /mnt/gentoo
```

Далее вы можете придерживаться стандартного руководства по установке, начиная с части 4. Кроме монтирования /proc. Тут вам надо поступить по-другому:

**Листинг 3.2:** Монтирование с привязкой псевдо-ФС proc  

```
# mount -o bind /proc /mnt/gentoo/proc
```

Также знайте, что некоторые возможности (FEATURES) портежей не будут работать в knoppix. Будьте особенно осторожны с userpriv и usersandbox. Если вы обнаружили, что начали получать ошибки, будет разумным отключить все или некоторые необязательные возможности.

## 4. Установка со Stage 1 без доступа в сеть

Создайте установочный компакт-диск из iso-образа.

Возьмите последний snapshot портежей из <http://distro.ibiblio.org/pub/linux/distributions/gentoo/snapshots/> (или с любимого зеркала). Запишите этот tarball на существующий раздел в компьютере, где будет проходить установка, или прожгите его на CD.

Следуйте инструкции из Gentoo Install Doc вплоть до chroot /mnt/gentoo в главе 6. Если у вас в наличии только один привод CD-ROM, не забудьте при загрузке ввести опцию dosache чтобы иметь возможность размонтировать установочный компакт-диск и смонтировать CD со snapshot-ом портежей.

Открыв новую консоль (Alt-F2), а в ней руководство по установке, мы можем следовать ему вплоть до запуска скрипта bootstrap.sh.

Предупреждение: В старых выпусках установочного компакт-диска требуется изменить пароль командой passwd прежде чем логиниться с другой консоли.

Возвращаемся к первой консоли (Alt-F1, without chroot) и монтируем второй CD в /mnt/gentoo/mnt/cdrom2. Копируем portage tarball с cdrom2 и распаковываем в /mnt/gentoo/usr.

**Листинг 4.1:** Монтирование snapshot cd  

```
# umount /mnt/cdrom
# mkdir /mnt/gentoo/mnt/cdrom2
# mount /dev/cdroms/cdrom0 /mnt/gentoo/mnt/cdrom2
# cp /mnt/gentoo/mnt/cdrom2/portage-$date.tar.bz2 /mnt/gentoo/usr
# cd /mnt/gentoo/usr
# tar -xvjp portage-$date.tar.bz2
```

Снова переключаемся на вторую консоль. Если сейчас попытаться запустить bootstrap.sh, ничего не выйдет, так как он не сможет закачать ни одного файла. Мы получим эти файлы из другого места и разместим их в /usr/portage/distfiles (в консоли F2).

Нам необходим список пакетов Stage1: glibc, baselayout, texinfo, gettext, zlib, binutils, gcc, ncurses плюс их зависимости.

Примечание: Не забывайте, что вам нужны версии пакетов, синхронизированные с вашим деревом портежей.

**Листинг 4.2:** Получение списка для скачивания

(Не забывайте 2 перед >)

```
# emerge -fp glibc baselayout texinfo gettext zlib binutils gcc ncurses 2> stage1.list
# mount -t vfat /dev/fd0 /mnt/floppy
# cp /mnt/gentoo/stage1.list /mnt/floppy
# umount /mnt/floppy
```

Вставьте дискету в компьютер с хорошим каналом. Если заглянуть в файл stage1.list, можно увидеть, что в нем перечисляются несколько ссылок на каждую загрузку. К сожалению, это не совсем то, что нам надо. Сначала список надо почистить от лишнего:

**Листинг 4.3:** Stripping URLs

(Этот скрипт привязан к формату вывода emerge, который может измениться без предварительного предупреждения - используйте с осторожностью!)

```
# cut -f 1 -d ' ' stage1.list > stage1.download
```

Теперь используйте wget для загрузки списка исходников:

**Листинг 4.4:** Загружаем пакеты с помощью wget

```
# wget -N -i stage1.download
```

Получив все файлы, переносим их в наш компьютер в /mnt/gentoo/usr/portage/distfiles. Вот теперь можно запускать bootstrap.sh. Повторяем ту же операцию с wget для stage2 и 3.

## 5. Бездисковая установка с загрузкой через PXE

### Что необходимо

Нужна сетевая карта на бездисковом клиенте, которая способна загружаться по протоколу PXE (многие карточки 3com). BIOS тоже должен поддерживать такой способ загрузки.

### Настройки на стороне сервера

Создание папок: Во-первых, следует создать папки, в которых будет размещена ваша бездисковая система. Начнем с /diskless, в ней будет находиться по отдельной папке для каждого бездискового клиента. Дальше мы рассмотрим работу с клиентом 'eta'.

**Листинг 5.1:** создание папок

```
# mkdir /diskless
# mkdir /diskless/eta
# mkdir /diskless/eta/boot
```

Настройка DHCP и TFTP: Клиент получает загрузочную информацию, используя

DHCP, затем загружает необходимые файлы с помощью TFTP. Собираем DHCP командой `emerge` и настраиваем его для наших нужд. Добавляем следующее в `/etc/dhcp/dhcpd.conf`:

Примечание: Мы получим статический IP для клиента и путь к загрузочному образу PXE, здесь `pxegrub`. Вам необходимо подставить соответствующий MAC-адрес сетевой карты вашего клиента и определиться с директорией, где будут храниться клиентские файлы.

Для DHCPd, запустите `emerge dhcp` (или любой другой сервер DHCP на свое усмотрение). Убедитесь, что в `/etc/conf.d/dhcp` выбран верный интерфейс.

**Листинг 5.2:** `dhcp.conf`

```
option option-150 code 150 = text ;
ddns-update-style none ;
host eta {
    hardware ethernet 00:00:00:00:00:00;
    fixed-address ip.add.re.ss;
    option option-150 "/eta/boot/grub.lst";
    filename "/eta/boot/pxegrub";
}
```

Для установки TFTP, соберите `app-admin/tftp-hpa`. Впишите следующее в `/etc/conf.d/in.tftpd`:

**Листинг 5.3:** `in.tftpd`

```
INTFTPD_PATH="/diskless"
INTFTPD_USER="nobody"
INTFTPD_OPTS="-u ${INTFTPD_USER} -l -vvvvvv -p -c -s ${INTFTPD_PATH}"
```

Настройка GRUB: Я использую GRUB для PXE-загрузки. Чтобы можно было использовать PXE, вы должны самостоятельно скомпилировать GRUB ... ведь это просто. Во-первых, скачайте самую свежую версию исходников GRUB (`emerge -f grub` поместит tarball в `/usr/portage/distfiles`). Скопируйте tarball в `/diskless` и скомпилируйте GRUB, чтобы получить доступный для рхе бинарник. Готовый бинарник копируем в boot-папку бездискового клиента. И редактируем файл `grub.lst`:

**Листинг 5.4:** установка grub

```
# tar zxvf grub-0.92.tar.gz
# cd grub-0.92
# ./configure --help
(Среди опций вы увидите список поддерживаемых драйверов сетевых интерфейсов.)
(Выберите драйвер, совместимый с вашей картой. Здесь он фигурирует как $nic)
# ./configure --enable-diskless --enable-$nic
# make
# cd stage2
# cp pxegrub /diskless/eta/boot/pxegrub
# nano -w /diskless/eta/boot/grub.lst
```

**Листинг 5.5:** `grub.lst`

```
default 0
timeout 30
```

```
title=Diskless Gentoo
```

```
root (nd)
kernel /eta/bzImage ip=dhcp root=/dev/nfs nfsroot=ip.add.re.ss:/diskless/eta
```

(В опции nfsroot указывается IP адрес сервера и путь к директории,)  
(в которой на сервере лежат файлы нашего бездискового клиента.)

Настройка NFS: NFS легко сконфигурировать. Все что нам надо - это добавить одну строку в конфигурационный файл /etc/exports:

**Листинг 5.6:** /etc/exports  
# nano -w /etc/exports  
# /etc/exports: NFS file systems being exported. See exports(5).  
/diskless/eta eta(rw, sync, no\_root\_squash)

Обновление hosts: Еще одно важное дело - изменение файла /etc/hosts.

**Листинг 5.7:** /etc/hosts  
127.0.0.1 localhost  
  
192.168.1.10 eta.example.com eta  
192.168.1.20 sigma.example.com sigma

## Создание системы на сервере

Вы можете захотеть перезагрузить сервер с установочного диска Gentoo, хотя точно так же можете немедленно продолжить, если знать, как выполнять инструкции по установке Gentoo на установленной системе. Следуйте стандартной процедуре установки, как описано в Gentoo Install Howto, НО с учетом следующих отличий: Для монтирования файловой системы сделайте следующее (здесь hdaX - раздел, в котором создана папка /diskless). Вам не надо монтировать другие разделы, так как все файлы располагаются в /diskless/eta.

**Листинг 5.8:** монтирование файловой системы  
# mount /dev/hdaX /mnt/gentoo

Stage tarballs и chroot: Этот пример использует tarball stage3. Смонтируйте /proc в вашу бездисковую директорию и выполните chroot в нее(директорию) для продолжения установки. Дальше - по инструкции вплоть до конфигурации ядра. Предупреждение: Будьте очень внимательны при разархивировании stage tarball. Не распакуйте его в существующую систему.

**Листинг 5.9:** извлечение stage tarball  
# cd /mnt/gentoo/diskless/eta/  
# tar -xvjp /mnt/cdrom/gentoo/stage3-\*.tar.bz2  
# mount -t proc /proc /mnt/gentoo/diskless/eta/proc  
# cp /etc/resolv.conf /mnt/gentoo/diskless/eta/etc/resolv.conf  
# chroot /mnt/gentoo/diskless/eta/ /bin/bash  
# env-update  
# source /etc/profile

Конфигурация ядра: При выполнении **make menuconfig** во время конфигурации

ядра, не забудьте сделать доступными следующие опции (помимо рекомендуемых в руководстве по установке):

**Листинг 5.10:** опции menuconfig

- Your network card device support  
(в ядре, \*не\* как модуль!)

- Under "Networking options" :

[\*] TCP/IP networking  
[\*] IP: kernel level autoconfiguration  
[\*] IP: DHCP support  
[\*] IP: BOOTP support

- Under "File systems --> Network File Systems" :

<\*> NFS file system support  
[\*] Provide NFSv3 client support  
[\*] Root file system on NFS

Сохраните ядро в / своей изолированной среды (не в /boot), в соответствии с настройкой rhexgrub, определенной ранее. Затем настройте /etc/fstab своего бездискового клиента.

**Листинг 5.11:** /etc/fstab

**# nano -w /etc/fstab**

```
/dev/cdroms/cdrom0 /mnt/cdrom iso9660 noauto,ro 0 0
proc /proc proc defaults 0 0
tmpfs /dev/shm tmpfs nodev,nosuid,noexec 0 0
```

Вам также потребуется предотвратить запуск на клиенте проверки файловой системы:

**Листинг 5.12:** Исключение запуска проверки файловой системы на клиенте

**# touch /fastboot**

**# echo "touch /fastboot" >> /etc/conf.d/local.start**

Установите nfs-utils, так как ваш клиент будет сильно зависеть от них:

**Листинг 5.13:** Установка nfs-utils

**# emerge nfs-utils**

Загрузчик. Не устанавливайте других загрузчиков, так как один у нас уже есть - rhexgrub. Просто завершаете установку и перезагружаете сервер. Запустите службы, необходимые для загрузки новоиспеченного клиента: DHCP, TFTPД и NFS.

**Листинг 5.14:** Запуск служб

```
# /etc/init.d/dhcp start
# /etc/init.d/in.tftpd start
# /etc/init.d/nfs start
```

## Загрузка нового клиента

Чтобы клиент загрузился, надо настроить bios и сетевую карту для использования PXE как первого способа загрузки - до CD-ROM или floppy. Помощником вам будет документация к оборудованию или website производителя. Сетевая карта должна получить IP адрес от DHCP и загрузить PXE-образ GRUB при помощи TFTP. Дальше вы увидите симпатичное черно-белое меню GRUB, в котором можно выбрать ядро для загрузки, а нажав Enter, и загрузить его. Если все в порядке, ядро примонтирует корневую файловую систему, используя NFS, и выведет на терминал приглашение для входа в систему. Вот и все.

## 6. Установка Gentoo из существующей Linux-системы

### Что необходимо

Для установки Gentoo из существующей системы Linux вам должна быть доступна команда chroot, а также копия установочного tarball-а Gentoo, либо его ISO-образ. Не мешает доступ в сеть, если вы не хотите ограничиваться тем, что есть в вашем tarball-е. (к слову, tarball - это просто файл, заканчивающийся на .tbz или .tar.gz). Автор (ориг. текста) использовал RedHat Linux 7.3 как "базовую" ОС, хотя это не так уж важно. Итак, приступим!

### Обзор

Сначала надо отвести раздел для Gentoo, изменив размер существующего раздела Linux, смонтировать его, разархивировать tarball, chroot внутрь нашей псевдо-системы, и начинаем строительство. После настройки загрузчика немного поработаем напильником, просто для того, чтобы система была способна загрузиться самостоятельно, перезагружаемся и пользуемся Gentoo.

### Как отвести место для Gentoo?

Корневой раздел - это файловая система, примонтированная к /. Запущенная команда mount показывает, о чем я веду речь. Еще мы можем выполнить df (disk free), чтобы посмотреть, сколько свободного места у нас есть. Прошу заметить - совсем не обязательно кромсать ваш root раздел! Вы можете отрезать в любом другом месте, доступном вашей программе редактирования разделов жесткого диска, но мы поговорим об этом чуть позже.

**Листинг 6.1:** Информация о файловых системах

```
# mount
/dev/hdb2 on / type ext3 (rw)
none on /proc type proc (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
none on /dev/shm type tmpfs (rw,nodev,nosuid,noexec)
# df -h
Filesystem      Size Used Avail Use% Mounted on
/dev/hdb2       4.0G 1.9G 2.4G 82% /
none            38M  0 38M  0% /dev/shm
```

Как видите, раздел, смонтированный как /, носит имя /dev/hdb2 и на нем еще свободно 2.4 гигабайта. В моем случае, думаю, 400 МБ свободного пространства хватит с головой, следовательно, под Gentoo отводим 2 ГБ. Совсем неплохо, учитывая, что большинству пользователей хватит и одного гигабайта. Приступим!

## Сборка parted для переразбивки разделов

**Parted** - утилита, поддерживаемая GNU foundation, старым, уважаемым, безразмерным проектом, софтом которого мы в данный момент и собираемся воспользоваться. Всего одним инструментом, который вполне нам подходит. Называется он parted, partition editor, загрузить его можно из <http://www.gnu.org/software/parted/>

Примечание: Конечно, существуют и другие инструменты для работы с разделами, но автор не уверен/не интересуется, сможет ли PartitionMagic(tm) или любая другая программа сделать то, что нам нужно. Проверить - это забота читателя.

Взгляните на ту страницу руководства parted, где речь идет о типе файловой системы, размер которой вы собираетесь изменять, и узнайте, сможет ли parted это сделать. Если нет, что ж, вам не повезло. Придется убить какой-нибудь раздел и создать по-новому, чтобы было место для Gentoo. Вперед, качаем программу, устанавливаем. Возникает проблема. Мы хотим изменить размер корневого раздела, следовательно нам нужно загрузить с дискеты минимальную linux-систему и использовать предварительно скомпилированный parted для редактирования /. Если в вашем случае можно отмонтировать выбранный раздел, оставаясь в Linux, вам везет и нет необходимости делать то, что описано ниже. Просто соберите parted и запустите его на разделе, который хотите урезать. А вот то, что я сделал в своей системе.

**Важно:** Убедитесь, что действия, которые вы собираетесь проделать с разделом, поддерживаются parted!

Скачайте загрузочный/корневой диск mininux (бесплатный мини-дистрибутив Linux с ядром 2.4 на дискете) с <http://mininux.free.fr/uk/>, создайте дискету, как описано в прилагающейся документации, вставьте еще одну чистую дискету в дисковод и перейдите к следующему шагу.

Примечание: Опять замечу, что Linux - синоним понятия "Этот путь не единственный". Наша цель - запустить parted на отмонтированном разделе жесткого диска и это можно сделать не только описанным способом. Вы можете использовать любой другой набор boot/root дискет, не только mininux. Вы можете вообще не выполнять данный шаг, главное, что нужно - отмонтировать файловую систему, которую вы собираетесь использовать под Gentoo, и переключить ее, используя parted.

**Листинг 6.2:** Создание вспомогательной дискеты

```
# mkfs.minix /dev/fd0
```

```
480 inodes
```

```
1440 blocks
```

```
Firstdatazone=19 (19)
```

```
Zonesize=1024
```

```
Maxsize=268966912
```

Продолжим со сборкой parted. Если он еще не закачан и не распакован, сделайте это. А затем перейдите в каталог, куда вы его распаковали. И здесь выполните следующий набор команд для сборки утилиты и записи ее на дискету.

**Листинг 6.3:** Подготовка вспомогательной дискеты

```
# mkdir /floppy; mount -t minix /dev/fd0 /floppy &&
```

```
export CFLAGS="-O3 -pipe -fomit-frame-pointer -static" && ./configure
```

```
&& make && cp parted/parted /floppy && umount /floppy
```

Поздравляю, вы готовы перезагрузиться и изменять выбранный раздел. Только делайте это после хотя бы беглого просмотра документации к parted на GNU website. Наберитесь терпения - изменение размера может занять больше получаса на больших дисках. Загрузитесь с дискеты mininux, а после этого замените загрузочную дискету на вспомогательную, которую мы только что создали. Введите `mount /dev/fd0 /floppy` и parted будет доступен в папке /floppy. Нам туда. Запускаете parted и перекраиваете нужный раздел жесткого диска. Когда этот бесконечный процесс наконец завершится, все будет готово для того, чтобы получить истинное наслаждение, устанавливая Gentoo. Диск, с которым мы работаем - это тот, на котором находится раздел, размер которого мы меняем. К примеру, если мы хотим изменить /dev/hda3, то диск, соответственно, будет называться /dev/hda.

**Листинг 6.4:** Команды, которые следует выполнить, загрузившись с mininux

```
# mount /dev/fd0 /floppy
# cd /floppy; ./parted [диск, с которым мы работаем]
(parted) print
Disk geometry for /dev/hdb: 0.000-9787.148 megabytes
Disk label type: msdos
Minor  Start   End     Type    Filesystem  Flags
1       0.031 2953.125 primary ntfs
3     2953.125 3133.265 primary linux-swap
2     3133.266 5633.085 primary ext3
4     5633.086 9787.148 extended
5     5633.117 6633.210 logical
6     6633.242 9787.148 logical ext3
(parted) help resize
resize MINOR START END      resize filesystem on partition MINOR

MINOR is the partition number used by Linux. On msdos disk labels, the
primary partitions number from 1-4, and logical partitions are 5
onwards.
START and END are in megabytes
(parted) resize 2 3133.266 4000.000
```

Важно: Наберитесь терпения! Компьютер работает! Посмотрите на индикатор обращения к жесткому диску (находится на корпусе), чтобы убедиться, что это так. Процесс может занять от 2 до 30 минут.

Когда процесс завершится, перезагрузитесь в вашу старую linux-систему. После этого откройте документ Настольная книга Gentoo: подготовка дисков и следуйте инструкциям. Дойдя до chroot, используйте следующую команду чтобы очистить переменные окружения:

```
Листинг 6.5: Обновление среды при chroot
# env -i HOME=$HOME TERM=$TERM chroot /mnt/gentoo /bin/bash
# /usr/sbin/env-update
# source /etc/profile
```

Удачи!



## Полезные советы по установке Gentoo/x86

Ссылка на оригинал: <http://www.gentoo.org/doc/ru/gentoo-x86-tipsntricks.xml>

С версии: 1.0

### 1. Вступление

Для начала

Этот документ содержит различные полезные советы по установке Gentoo/x86. Большинство из них описаны кратко — подразумевается, что они послужат дополнением к руководству по установке, а не заменой ему.

### 2. Расширенная установка

Программный RAID

Примечание: Если вы не знакомы с программным RAID, пожалуйста, прочтите Software-RAID-HOWTO (англ.).

Примечание: Более подробное описание установки приведено в руководстве по быстрой установке программного RAID и LVM2 для x86 (англ.).

После загрузки с установочного CD, загрузите соответствующие модули RAID. Например, если вы собираетесь использовать RAID-1:

**Листинг 2.1:** Загрузка модуля RAID-1  
`# modprobe raid1`

Разбивая свои диски, убедитесь, что используете тип раздела fd (Linux raid autodetect), а не 83 (Linux native). Тип раздела можно изменить, используя команду `t` программы fdisk.

Теперь, до начала создания массивов RAID, нам потребуется создать узлы метаустройств:

**Листинг 2.2:** Создание узлов метаустройств  
`# mkno`  
`# mknod /dev/md2 b 9 2`  
`# mknod /dev/md3 b 9 3d /dev/md1 b 9 1`

После разбивки на разделы, создайте файл `/etc/mdadm.conf` (да, именно так, в среде установочного CD), с помощью mdadm, расширенного средства управления RAID. Например, чтобы зеркалировать (RAID-1) разделы `boot`, `swap` и `root`, охватывая `/dev/sda` и `/dev/sdb`, можете использовать:

**Листинг 2.3:** Создание устройств raid командой mdadm

```
# mdadm --create --verbose /dev/md1 --level=1 --raid-devices=2 /dev/sda1 /dev/sdb1
# mdadm --create --verbose /dev/md2 --level=1 --raid-devices=2 /dev/sda2 /dev/sdb2
# mdadm --create --verbose /dev/md3 --level=1 --raid-devices=2 /dev/sda3 /dev/sdb3
```

**Важно:** На загрузочном разделе не следует использовать никаких разновидностей чередования (striping), таких как RAID-0 or RAID-5.

Драйвер Linux Software RAID начнет создавать метаустройства. Вы можете проследить за этим в /proc/mdstat. Перед продолжением дождитесь, пока создание метаустройств окончательно завершится..

**Листинг 2.4:** Сохранение сведений о созданных устройствах

```
# mdadm --detail --scan > /etc/mdadm.conf
```

Теперь и далее используйте /dev/md1 для загрузочного раздела, /dev/md2 для раздела подкачки и /dev/md3 для корневого раздела.

Прямо перед изменением корня (chroot), не забудьте скопировать /etc/mdadm.conf в /mnt/gentoo/etc.

При конфигурации ядра, обязательно включите соответствующую поддержку RAID в состав ядра, а не модулем.

При установке дополнительных утилит, также установите mdadm. Заметьте, что она есть не на всех установочных CD, поэтому у вас может не получиться бессетевая установка Gentoo на программный raid.

При настройке загрузчика не забудьте установить его в MBR обоих дисков, если используется зеркалирование.

## ATA RAID с ядрами 2.4

Удостоверьтесь, что вы загрузились с установочного CD с параметром doataraid. После загрузки, проверьте содержимое /dev/ataraid. Там должны находиться различные каталоги disc\* для каждого жесткого диска, доступного в ATA RAID. Целый диск показывается как disc, а разделы — как part\*.

Выпишите различные файлы устройств /dev/ataraid/disc\*/\*, на которые будете устанавливать Gentoo. При установке вам потребуется указывать этот путь вместо /dev/hda, указанного в примерах.

Перед изменением корня, свяжите структуру /dev с новой средой:

**Листинг 2.5:** Связывание /dev

```
# mount -o bind /dev /mnt/gentoo/dev
```

При настройке ядра не забудьте включить поддержку вашего ATA RAID чипсета с нужными параметрами. Например, для популярной системы ATA RAID Promise FastTrack built-in RAID требуется включение в ядро Promise FastTrack Options.

При настройке GRUB сначала потребуется создать загрузочный диск GRUB. Это не так сложно, как кажется. Сначала установите GRUB как обычно, а дойдя до пункта, в котором GRUB устанавливается в MBR, следуйте этим инструкциям:

**Листинг 2.6:** Создание загрузочного диска GRUB

```
# cd /boot/grub
# dd if=stage1 of=/dev/fd0 bs=512 count=1
# dd if=stage2 of=/dev/fd0 bs=512 seek=1
```

Еще вам потребуется записать файл grub.conf. Здесь нет никаких отличий от установочных инструкций, просто убедитесь, что запись root= указывает на ваше устройство ATA RAID.

После окончания установки, загрузитесь со своего загрузочного диска GRUB. Вы должны увидеть приглашение командной строки GRUB. Теперь настройте GRUB для загрузки с устройства ATA RAID:

**Листинг 2.7:** Установка GRUB на ATA RAID

```
grub> root (hd0,x)
grub> setup (hd0)
grub> quit
```

Теперь перезагрузитесь (вытащив загрузочную дискету GRUB из дисковода).

Пользователи LILO могут просто следовать указаниям руководства по установке.

## Использование ядра с установочного CD

Если вы не хотите компилировать ядро сами, можно взять ядро с установочного компакт-диска и скопировать его в свою систему. Дойдя в процессе инсталляции Gentoo до стадии компиляции ядра, перейдите на другую виртуальную консоль (Alt-F2) и войдите в систему как ROOT, используя пароль, установленный вами в начале установки (passwd root).

Скопируйте ядро и модули в свою систему:

**Листинг 2.8:** Копирование ядра с установочного CD

(\${KN} это название ядра, обычно это что-то вроде 'gentoo' или 'smp')

```
cdimage ~# cp /mnt/cdrom/isolinux/${KN} /mnt/cdrom/isolinux/${KN}.gz /mnt/gentoo/boot
cdimage ~# mkdir -p /mnt/gentoo/lib/modules
cdimage ~# cp -Rp /lib/modules/`uname -r` /mnt/gentoo/lib/modules
```

Удостоверьтесь в том, что вы установили hotplug (emerge hotplug) и уже добавили его в загрузку. Чтобы все запущенные сейчас модули (с установочного CD) загружались на вашей машине, запустите следующие команды из среды с измененным корнем (chroot):

**Листинг 2.9:** Добавление всех запущенных модулей в файл modules.conf

```
# cat /proc/modules | cut -d ' ' -f 1 >> \
/etc/modules.autoload.d/kernel-`uname -r` | cut -d . -f -2`
# modules-update
```

### 3. Упрощение установки

#### Как оставить терминал без присмотра

Многим хочется отойти от своей системы, пока она компилируется. Иногда это довольно сложно, поскольку установка производится в месте, где много народу и нельзя доверять всем подряд. На этот случай пригодится возможность проводить компиляцию в фоновом режиме, выйдя из всех терминалов.

Есть несколько возможных путей. Первый — использовать screen. После загрузки с LiveCD, установите пароль для root и запустите сеанс screen:

Примечание: screen есть не на всех LiveCD. Если у вас его нет, придется использовать один из других способов, описанных в этом разделе.

**Листинг 3.1:** Запуск сеанса screen  
**# screen -S gentoo**

Из сеанса screen можно хоть провести полную установку. Захотев уйти от терминала, нажмите Ctrl-a, d (то есть control и a одновременно, затем d), чтобы открепить свой сеанс screen. Теперь можно с уверенностью выйти из системы.

Чтобы восстановить доступ к терминалу, опять войдите как root и прикрепитесь к запущенному сеансу screen:

**Листинг 3.2:** Прикрепление к сеансу screen  
**# screen -x gentoo**

Если вы не можете использовать screen, есть другой путь отойти от терминала. Следуйте инструкциям по установке, а дойдя до пункта, в котором запускается длительная компиляция (например, шаг с запуском ./scripts/bootstrap.sh), используйте команду nohup, которая позволит процессу продолжиться, даже если вы выйдете. Не забудьте добавить в конце "&", иначе процесс не будет выполняться в фоновом режиме! Запомните, в каком каталоге вы находитесь (команда pwd покажет ее), так как это вам позже понадобится.

**Листинг 3.3:** Использование nohup  
**# pwd**  
**/usr/portage**  
**# nohup ./scripts/bootstrap.sh &**

Теперь выйдите из среды измененного корня (exit) и из сеанса загрузочного CD. Компиляция продолжится в фоновом режиме.

Захотев проверить компиляцию, войдите как root (на установочный CD) и сделайте chroot обратно в свою среду, затем перейдите в оставленный каталог:

**Листинг 3.4:** Chroot обратно  
`# chroot /mnt/gentoo /bin/bash`  
`# env-update && source /etc/profile`  
`# cd /usr/portage`

Теперь запустите команду `less` на файле `pothp.out`, расположенном внутри каталога. Компиляция добавляет свои сообщения в этот файл, так что при желании следить за ней запустите `less pothp.out` и нажмите `F`, чтобы наблюдать за ее ходом. Когда компиляция закончится, можно приступить к следующему пункту указаний по установке.

Если вам надоело следить за изменениями, нажмите `Ctrl-C`, затем `q`. Это остановит только процесс `less`, не затрагивая процесс компиляции.

## 4. Решение ошибок/проблем

### *Тщательное тестирование дисков*

Если вы считаете, что необходимо тщательно проверить ваш диск на предмет целостности (неисправные секторы и т.д.), можете включить параметр `-c` при создании на нем файловой системы `ext2` или `ext3` (используя `mke2fs`). Это запустит проверку на чтение, которая пометит все неисправные блоки. Если вы настоящий параноик, можете включить `-c -c`, чтобы провести детальный тест на чтение/запись.

**Листинг 4.1:** Проверка целостности диска  
`# mke2fs -j -c /dev/hda3`

## Восстановление сбойной установки

Если по какой-то причине ваша установка Gentoo дает сбой, вам не придется повторять ее раз за разом с самого начала. Вместо этого можно спокойно вернуться к моменту, в который вы, как вам кажется, ошиблись (или где, как вы считаете, есть ошибка в инструкции), и попробовать другой подход.

Прежде всего, вам потребуется перейти обратно в свою среду Gentoo Linux командой `chroot`. Снова следуйте указаниям, пропуская шаги по разбивке диска, так как ваши разделы уже созданы и даже заполнены. Таким образом, вы можете сразу монтировать эти разделы в `/mnt/gentoo`. Следует также пропустить шаги, связанные с извлечением файла стадии и изменением `make.conf` — вы же не хотите перезаписывать существующие файлы, не так ли?

Изменив корень на свою среду Gentoo Linux, сразу переходите к шагу, где, как вам кажется, следует попробовать действовать по-другому. Не повторяйте все шаги, такие как самогенерация, если не считаете, что именно там что-то пошло не так.

Например, если вы считаете, что неверно настроили `grub.conf`, можно сразу запустить свой редактор, чтобы изменить `/boot/grub/grub.conf`.

Попробовав другой подход в своей ситуации, вы, скорее всего, сможете представить, сколько последующих шагов потребуется выполнить снова. Если

последующие действия зависели от вашего изменения, их потребуется повторить.

Например:

изменив переменную в `make.conf`, вам потребуется выполнить всю последующую компиляцию, поскольку ее результаты зависят от настройки `make.conf`

изменив `/boot/grub/grub.conf`, можно сразу выходить из среды измененного корня и перезагружаться, так как никакие последующие шаги не зависят от `grub.conf`

перекомпилировав свое ядро, вам нужно лишь убедиться, что конфигурация вашего начального загрузчика указывает на верный образ ядра (убедитесь, что вы смонтировали свой `/boot!`), затем можно выйти из среды измененного корня и перезагрузиться

изменив `/etc/fstab`, можно выходить из среды измененного корня и перезагружаться

Как видите, после большинства действий по восстановлению можно сразу перезагружаться. Лишь изредка вам потребуется повторять последующие шаги установки.

# Настольная книга Gentoo Linux x86

Ссылка на оригинал: <http://www.gentoo.org/doc/ru/handbook/handbook-x86.xml>

С версии: 1.0

Содержание:

## ***Установка Gentoo***

Из этой части вы узнаете, как устанавливать Gentoo на свою систему.

### ***Об установке Gentoo Linux***

В этой главе дается общее представление о предлагаемом порядке установки.

### ***Выбор подходящего источника для установки***

Gentoo можно устанавливать по-разному. Здесь описывается, как установить Gentoo с минимального установочного диска. Способ подходит и для установки с универсального установочного диска.

### ***Настройка сети***

Чтобы загрузить новейший исходный код, потребуется настройка сети. Здесь описывается порядок ее выполнения.

### ***Подготовка дисков***

Чтобы установить Gentoo, нужно создать подходящие дисковые разделы. В этой главе описывается, как разбить диск для будущего использования.

### ***Установка установочных файлов Gentoo***

Установка Gentoo выполняется с помощью архива третьей стадии (stage3). В этой главе мы расскажем, как извлечь файл третьей стадии и настроить Portage.

### ***Установка базовой системы Gentoo***

После установки и настройки третьей стадии в вашем распоряжении оказывается базовая система Gentoo. Здесь описывается, как этого достичь.

### ***Настройка ядра***

Ядро Linux — сердце каждого дистрибутива. В этой главе описывается, как его настроить.

### ***Настройка параметров системы***

Вам потребуется отредактировать несколько важных конфигурационных файлов. Из этой главы вы получите представление об этих файлах и поймете, как с ними обращаться.

### ***Установка нужных системных средств***

Как отмечено ранее, Gentoo — это богатство выбора. В этой главе мы поможем вам выбрать и установить некоторые важные инструменты.

### ***Настройка начального загрузчика***

Для архитектуры x86 существует несколько загрузчиков. Каждый из них

настраивается по-своему. В этой главе мы шаг за шагом объясним вам порядок настройки начального загрузчика для ваших нужд.

### ***Завершение установки Gentoo***

Установка почти закончена. Осталось создать одну (или несколько) учетных записей для пользователей вашей системы. Как это делается, описано в данной главе.

### ***Чем заняться дальше?***

Теперь у вас появилась собственная система Gentoo. Но чем же заняться дальше?

### ***Работа с Gentoo***

Здесь вы научитесь работать с Gentoo: устанавливать программное обеспечение, изменять значения переменных, управлять поведением Portage и т.д.

### ***Введение в Portage***

В этой главе описываются «простые» шаги, которые вам точно потребуются знать для поддержания в порядке и обслуживания программного обеспечения.

### ***USE-флаги***

USE-флаги (признаки использования) — очень важный аспект Gentoo. Прочитав эту главу, вы научитесь обращаться с ними и разберетесь, как USE-флаги влияют на систему.

### ***Возможности Portage***

Откройте для себя дополнительные возможности Portage: поддержку распределенной компиляции, кэш компилятора и др.

### ***Сценарии инициализации***

В Gentoo используется специальный формат сценариев инициализации (initscript), в котором, например, предусмотрены решения, управляемые зависимостями, и виртуальные сценарии. Здесь описываются эти аспекты, и объясняется, как обращаться со сценариями.

### ***Переменные среды***

В Gentoo можно довольно легко управлять системными переменными среды. В этой главе объясняется, как это делать, и описываются часто используемые переменные.

### ***Работа с Portage***

В этой части подробно описывается Portage, средство управления программным обеспечением Gentoo.

### ***Файлы и каталоги***

Чтобы поближе познакомиться с Portage, вам потребуется узнать, где же хранятся его файлы и данные.

### ***Настройка с помощью переменных***

Portage полностью настраивается с помощью различных переменных, которые устанавливаются в конфигурационном файле или как переменные среды.

### ***Смешение ветвей программного обеспечения***

Программное обеспечение в составе Gentoo подразделяется на ветви в зависимости от стабильности и поддержки различных архитектур. В этой главе рассказывается, как настраивать использование ветвей, а также как при необходимости



преодолевать такое разделение.

### ***Дополнительные средства Portage***

В состав Portage входит несколько дополнительных инструментов, которые могут значительно улучшить ваше впечатление от Gentoo. В этой главе раскрываются секреты использования dispatch-conf и других инструментов.

### ***Отступление от официального дерева***

Здесь даются советы и показываются приемы «выращивания» собственного дерева Portage, рассказывается о том, как синхронизировать только нужные категории, добавлять свои пакеты и т.д.

### ***Использование ebuild***

В этой главе рассказывается о шагах, выполняемых Portage при установке программного обеспечения, и об их самостоятельном выполнении с помощью утилиты ebuild.

## ***Настройка сети в Gentoo***

Полное руководство по сетям в Gentoo.

### ***Начальная настройка***

Руководство по быстрому запуску и подключению сетевого интерфейса в наиболее распространенных случаях.

### ***Расширенная настройка***

Здесь вы изучите, как работает конфигурация — это нужно сделать перед изучением модульного построения сети.

### ***Модульное построение сети***

В Gentoo предусмотрены гибкие сетевые средства: здесь рассказывается о выборе различных клиентов DHCP, настройке объединения, образования мостов, виртуальных сетей (VLAN) и т.п.

### ***Беспроводная сеть***

Настроить беспроводную сеть не совсем просто. Надеемся, мы поможем вам заставить ее работать.

### ***Дополнительные возможности***

Если вы ищете приключений, можете подключить к сети свои собственные функции.

Управление сетью

Для пользователей ноутбуков и тех, кто перемещает свои компьютеры из сети в сеть.

## ***А. Установка Gentoo***

## **А. Установка Gentoo**

### **1. Об установке Gentoo Linux**

#### **1.а. Введение**

##### ***Добро пожаловать!***

Прежде всего, добро пожаловать в Gentoo! Вы — на пороге мира больших возможностей и высокой производительности! Gentoo весь пропитан идеей свободы выбора. При установке Gentoo вы не раз убедитесь в этом: вам предстоит выбрать объем самостоятельной компиляции, способ установки Gentoo, службу журналирования системы и т.д.

**Gentoo** — быстрый современный метадистрибутив, обладающий большой чистотой и гибкостью. Gentoo основан на свободном программном обеспечении, и не скрывает от пользователя, «что под капотом». Portage, система управления пакетами Gentoo, написана на языке Python, что позволяет легко просматривать и изменять ее исходный код. Сборка Gentoo также выполняется из исходных текстов (хотя есть и поддержка бинарных пакетов), а настройка Gentoo выполняется с помощью обычных текстовых файлов. Другими словами — сплошная открытость и свобода!

Важно понимать, что Gentoo развивается именно благодаря свободе выбора. Мы стараемся ничего вам не навязывать. А если вам покажется обратное — пожалуйста, сообщите нам об этой оплошности.

##### ***Как организована установка?***

Установка Gentoo рассматривается как последовательность из 10 шагов, которым соответствуют главы 2—11. Каждый шаг приводит к определенному состоянию:

- **результат шага 1:** вы — в рабочей среде, готовой к установке Gentoo;
- **результат шага 2:** ваше подключение к интернету готово для установки Gentoo;
- **результат шага 3:** ваши жесткие диски готовы стать родным домом для Gentoo;
- **результат шага 4:** подготовлена установочная среда, и вы готовы переключиться (chroot) в новую среду;
- **результат шага 5:** развернуты пакеты ядра, общие для всех систем Gentoo;
- **результат шага 6:** вы скомпилировали собственное ядро Linux;
- **результат шага 7:** вы написали большую часть конфигурационных файлов системы Gentoo;
- **результат шага 8:** установлены необходимые системные средства (которые вы сами выбрали из славного списка);
- **результат шага 9:** установлен и настроен избранный загрузчик ОС, и вы вошли в новую систему Gentoo;
- **результат шага 10:** вы можете начинать изучение своей собственной среды Gentoo Linux!

Мы приложили все усилия, чтобы объяснить вам все «за» и «против» каждого варианта, когда вам предоставляется возможность выбора. При этом один из

вариантов помечен как «По умолчанию: ». Другие варианты помечены как «Альтернатива: ». Не подумайте, что вариант по умолчанию — наша рекомендация. Нам просто кажется, что именно его выбирает большинство пользователей.

Иногда есть возможность выполнить необязательный шаг. Такие шаги помечены как «Дополнительно: », и не требуются для установки Gentoo. Однако, некоторые из них будут обусловлены вашими предшествующими решениями. Мы будем сообщать об этом, как в момент выбора, так и непосредственно перед описанием необязательных шагов.

### ***Какие варианты установки существуют?***

Gentoo можно установить разными способами. Можно скачать и запустить один из установочных компакт-дисков, установить с имеющегося дистрибутива, с загрузочного CD (например, Knoppix), из сетевой загрузочной среды, с дискеты аварийного восстановления и т.д.

В этом руководстве описывается установка с установочных дисков Gentoo, и, в некоторых случаях, с помощью сетевой загрузки (netboot). Предполагается, что вы собираетесь устанавливать самые свежие версии пакетов. Если вам нужна установка, при которой не требуется использование сети, обратитесь к настольным книгам Gentoo 2006.1 (англ.), где даются указания по установке в бессетевой среде.

Если вы планируете использовать GRP (Gentoo Reference Platform — эталонная платформа Gentoo, набор бинарных пакетов, предназначенных для немедленного использования сразу после установки Gentoo), вам необходимо следовать инструкциям, приведенных в настольной книге Gentoo 2006.1 (англ.).

Чтобы получить сведения о других способах установки, прочитайте описание альтернативных способов установки. Также рекомендуется прочитать полезные советы по установке Gentoo. Если вы почувствуете, что приведенные указания по установке слишком подробны, обратитесь к краткому руководству по установке (см. перечень документации), если, конечно, такое существует для вашей архитектуры.

Кроме того, вы можете выбрать между компиляцией своей системы полностью «с нуля» или установкой заранее собранной среды, позволяющей запустить Gentoo практически моментально. Естественно, есть и промежуточные варианты, в которых вы не компилируете все подряд, а начинаете с полу готовой системы.

### ***Появились затруднения?***

Если при установке вы столкнулись с проблемой (или с ошибкой в документации по установке), войдите в нашу систему распределения запросов (англ.) и убедитесь, что такая ошибка еще не заявлена. В этом случае, создайте отчет об ошибке, чтобы мы о ней позаботились. Не бойтесь разработчиков, которым выпадает работа над (вашими) ошибками — людей они обычно не едят.

Обратите внимание, что, хотя руководство, которое вы сейчас читаете, посвящено определенной архитектуре, в нем упоминаются и другие архитектуры. Это связано с тем, что значительная часть текста настольной книги Gentoo является общей для всех архитектур (чтобы не дублировать работу, а также из-за острой нехватки разработчиков). Во избежание путаницы, мы стараемся сократить такие упоминания до минимума.

Если вы не уверены, пользовательская ли у вас ошибка (какую-то ошибку совершили вы, хотя внимательно прочитали документацию), или программная (какую-то ошибку совершили мы, хотя тщательно тестировали установку/документацию), то не стеснясь, заходите на канал **#gentoo** (а также **#gentoo-amd64**, **#gentoo-ru**, **#rugentoo**) сервера **irc.freenode.net**. Разумеется, мы в любом случае будем вам рады :)

Если у вас есть вопрос, касающийся Gentoo, сначала загляните в список распространенных вопросов (FAQ), входящий в состав документации Gentoo. Можно также обратиться к FAQ (англ.) в наших форумах. Если вы не найдете там ответа, задайте вопрос на **#gentoo**, нашем IRC-канале на **irc.freenode.net**. Да-да, кое-кто из нас — маньяки, висящие на IRC :-)

## **2. Выбор подходящего источника для установки**

### **2.a. Аппаратные требования**

#### **Введение**

Прежде чем начать, перечислим аппаратные требования, необходимые для успешной установки Gentoo на вашем компьютере.

Аппаратные требования:

- Центральный процессор i486 или новее;
- Память 64 МБ;
- Дисковое пространство 1.5 ГБ (исключая пространство подкачки);
- Пространство подкачки не менее 256 МБ.

### **2.b. Установочные компакт-диски Gentoo**

#### **Введение**

Установочные компакт-диски Gentoo — это загрузочные диски, содержащие самодостаточную среду Gentoo. Они позволяют загружать Linux прямо с компакт-диска. При запуске определяются ваши устройства и загружаются соответствующие драйверы. Эти диски сопровождаются разработчиками Gentoo.

Все установочные компакт-диски позволяют загрузиться, настроить сеть, разметить разделы жесткого диска и начать установку Gentoo из интернета. В настоящее время мы выпускаем два установочных компакт-диска, одинаково подходящих для установки Gentoo с загрузкой последних версий существующих пакетов из интернета.

Если вы хотите установить Gentoo без работоспособного соединения с интернетом, или желаете использовать одну из существующих программ установки, пожалуйста, следуйте указаниям по установке, приведенным в настольной книге Gentoo 2006.0 (англ.).

В настоящее время мы выпускаем два установочных компакт-диска:

Минимальный установочный диск Gentoo, маленький загрузочный компакт-диск без излишеств, единственное назначение которого — загрузить систему, подготовить подключение к сети и перейти к установке Gentoo.

Живой установочный диск Gentoo, содержащий все необходимое для установки Gentoo. На нем есть графическая среда, графическая и консольная программы автоматической установки, и, конечно, указания по установке для вашей архитектуры.

Чтобы помочь вам решить, какой из них нужен, мы привели главные преимущества и недостатки каждого диска.

### **Минимальный установочный диск Gentoo**

Минимальный установочный диск называется `install-x86-minimal-2006.0.iso` и занимает всего 49 МБ. Gentoo можно устанавливать с этого диска только при постоянном наличии работоспособного подключения к интернету. Минимальный установочный диск:

#### **За и против**

- + наименьший объем загрузки из сети
- не содержит ни архива `stage3`, ни снимков Portage, ни готовых двоичных пакетов, и поэтому не подходит для установки без сети

### **Живой (live) установочный диск Gentoo**

Живой установочный диск называется `install-x86-universal-2006.0.iso` и занимает 697 МБ. С этого диска вы сможете установить Gentoo даже без исправного подключения к интернету, на случай, если в процессе установки Gentoo на свой компьютер у вас появится неодолимое желание занести его еще и на соседний :) Живой установочный диск Gentoo:

#### **За и против**

- + содержит все, что требуется; установку можно выполнять даже без сетевого подключения;
- огромный объем загрузки.

### **Файл Stage3**

**Архив третьей стадии — `stage3` — это файл, содержащий минимальную среду Gentoo, пригодную для продолжения установки Gentoo в соответствии с инструкциями, данными в этом руководстве (другими словами: образ (снимок) минимальной системы.**

Когда-то в настольной книге Gentoo описывались варианты установки с использованием файла любой из трех существующих стадий создания системы. Несмотря на то, что в Gentoo до сих пор представлены файлы `stage1` и `stage2`, в официальном способе установки используется архив `stage3`. Если вас интересует установка Gentoo из файлов `stage1` или `stage2`, пожалуйста, обратитесь к сборнику распространенных вопросов по Gentoo, раздел как установить Gentoo, используя файлы Stage1 или Stage2?

Архивы **stage3** можно загрузить из `releases/x86/2007.0/stages/` или с любого из официальных зеркал Gentoo; на «живом диске» они не поставляются.

## 2.с. Скачивание, запись и загрузка с установочного диска Gentoo

### Скачивание образа и запись на диск

Итак, вы выбрали нужный установочный диск Gentoo. Начнем с загрузки его образа из сети, и его записи на компакт-диск. Ранее обсуждались варианты доступных дисков, но где же их взять?

Любой из установочных дисков (а при желании — заодно и диск пакетов, если есть) можно загрузить с одного из наших зеркал. Установочные диски расположены в каталоге `releases/x86/2007.0/installed`.

В этом каталоге находятся ISO-файлы. Это полные образы компакт-дисков, которые можно записать на CD-R.

Если вас волнует, не искажен ли загруженный файл, можно сверить его контрольную сумму MD5 с рассчитанной нами (например, `install-x86-minimal-2007.0.iso.DIGESTS`). Вычислить контрольную сумму MD5 можно утилитой `md5sum` в Linux/Unix, или программой `md5sum` (англ.) для Windows.

Другой способ убедиться в пригодности загруженного файла — с помощью GnuPG сверить криптографическую подпись с той, что хранится у нас (в файле с расширением `.asc`). Скачайте файл подписи, затем получите открытый ключ:

**Листинг 1:** Получение открытого ключа

```
$ gpg --keyserver subkeys.pgp.net --recv-keys 17072058
```

Теперь сверьте подпись:

**Листинг 2:** Сверка криптографической подписи

```
$ gpg --verify <файл подписи> <скачанный образ диска>
```

Записывать загруженный ISO-образ на компакт-диск нужно в «сыром» режиме. Как это сделать, сильно зависит от используемой программы. Здесь мы обсудим запись с помощью `cdrecord` и K3B. Дополнительные сведения находятся в нашем сборнике распространенных вопросов.

При использовании `cdrecord`, просто введите `cdrecord dev=/dev/hdc <скачанный iso-файл>` (/dev/hdc замените на путь к вашему устройству CD-RW).

При использовании K3B, выберите Tools > CD > Burn Image (Инструменты > Компакт-диск > Прожечь образ). Затем в области 'Image to Burn' ('Образ для записи') укажите свой ISO-файл. В завершение нажмите Start (Запуск).

### Загрузка с установочного компакт-диска

**Важно:** Перед тем, как приступить, полностью прочитайте этот подраздел, так как перед последующими действиями вам, скорее всего, больше не представится такая возможность.

Записав установочный компакт-диск, пора его загрузить. Уберите все компакт-диски из своих приводов CD, перезагрузите компьютер и войдите в BIOS. В зависимости от BIOS, для этого обычно нужно нажать DEL, F1 или ESC. В BIOS измените порядок загрузки так, чтобы обращение к CD-ROM выполнялось до обращения к жесткому диску. Этот параметр часто задается в разделе «CMOS Setup». Если порядок загрузки не изменить, система просто перезагрузится с жесткого диска, игнорируя CD-ROM.

Теперь поместите установочный диск в привод CD-ROM (наконец-то!) и перезагрузитесь. Должно появиться загрузочное приглашение. Здесь можно просто нажать ENTER, чтобы запустить процесс загрузки с параметрами по умолчанию, или загрузить установочный диск по-своему: указать ядро, потом загрузочные параметры, а затем нажать ENTER.

Указать ядро? Да, на нашем установочном компакт-диске есть несколько вариантов ядра. По умолчанию используется **gentoo**. Другие ядра предназначены для специфических аппаратных задач; вариант **-nofb** отключает кадровый буфер.

Ниже перечислены доступные варианты ядра:

Ядро	Описание
<b>gentoo</b>	ядро 2.6 с многопроцессорной поддержкой (используется по умолчанию)
<b>gentoo-nofb</b>	аналогично gentoo, но без поддержки кадрового буфера
<b>memtest86</b>	для выявления ошибок локальной оперативной памяти

Также можно указать параметры ядра, явно включающие или отключающие определенные режимы. Приведенный список соответствует тому, что выводится при нажатия F2 в экране загрузки.

**Листинг 3:** Параметры, которые можно сообщить выбранному ядру

- <b>agpgart</b>	загрузка agpgart (используйте при сбоях графики, зависаниях)
- <b>acpi=on</b>	включение поддержки ACPI
- <b>ide=nodma</b>	принудительное отключение DMA для сбойных IDE-устройств
- <b>doscsi</b>	поиск scsi-устройств (нарушает работу некоторых ethernet-плат)
- <b>dopcmcia</b>	запуск службы pcmcia для PCMCIA-приводов компакт-дисков
- <b>nofirewire</b>	отключение в initrd модулей firewire (предназначенных для приводов компакт-дисков с подключением firewire и т.п.)
- <b>nokeymap</b>	отключение выбора неамериканских раскладок клавиатуры
- <b>docache</b>	кэширование исполняемой части компакт-диска в памяти, позволяет размонтировать /mnt/cdrom для установки другого диска
- <b>nodetect</b>	запрет запуска hwsetup/kudzu и hotplug
- <b>nousb</b>	отключение загрузки модулей usb в initrd, отключает hotplug
- <b>nodhcp</b>	отмена автоматического запуска dhcp при обнаружении сетевой платы
- <b>nohotplug</b>	отключение загрузки службы hotplug
- <b>noapic</b>	отключение apic (попробуйте, если есть аппаратные проблемы с устройствами scsi, сетевыми платами и т.д.)
- <b>noevms</b>	отключение поддержки модулей EVMS2
- <b>nolvm2</b>	отключение поддержки модулей LVM2
- <b>hdx=stroke</b>	позволяет использовать жесткий диск целиком, даже если BIOS не поддерживает диски большого размера

- `noload=module1,[module2,...]`  
запрет загрузки определенных модулей ядра

Теперь загрузите систему с компакт-диска, выбрав ядро (если вас не устраивает ядро по умолчанию `gentoo`) и указав дополнительные параметры. В качестве примера мы покажем, как загрузить ядро `gentoo` с параметром `dopcmcia`:

**Листинг 4:** Загрузка установочного диска  
`boot: gentoo dopcmcia`

Появится заставка с полосой индикатора загрузки. Если вы устанавливаете Gentoo на машину с неамериканской раскладкой клавиатуры, нужно немедленно нажать ALT+F1, чтобы переключиться в подробный режим, и следовать появившимся указаниям. Если ничего не выбрано в течение 10 секунд, устанавливается раскладка по умолчанию (клавиатура США), а загрузка продолжается. По окончании загрузки для вас будет выполнен автоматический вход в «живой» Gentoo Linux в качестве «root» (администратора). В текущей консоли должно появиться приглашение root («#»). Также можно переключаться в другие консоли, нажимая ALT-F2, ALT-F3 и ALT-F4. Вернуться в первоначальную консоль можно нажатием ALT-F1.

Приступим к дополнительной настройке оборудования.

### Дополнительная настройка оборудования

При загрузке с установочного компакт-диска система пытается определить все устройства и загрузить соответствующие модули для их поддержки. В подавляющем большинстве случаев она очень хорошо справляется с этим. Однако, в редких случаях некоторые из нужных модулей не загружаются автоматически. Если какие-либо устройства пропущены при автоматическом опросе шины PCI, модули ядра для их поддержки вам придется загрузить вручную.

В следующем примере мы попробуем загрузить модуль **8139too** (для поддержки некоторых типов сетевого интерфейса):

**Листинг 5:** Загрузка модуля ядра  
`# modprobe 8139too`

Если вам нужна поддержка PCMCIA, требуется запустить сценарий инициализации `pcmcia`:

**Листинг 6:** Запуск сценария инициализации PCMCIA  
`# /etc/init.d/pcmcia start`

### *Дополнительно: оптимизация скорости жесткого диска*

Если вы опытный пользователь, возможно, вам захочется настроить скоростные параметры жесткого диска IDE с помощью программы **hdparm**. Указав параметр `-tT`, можно измерить скорость вашего диска (для уточнения значений запускайте программу несколько раз подряд):

**Листинг 7:** Тестирование производительности диска  
`# hdparm -tT /dev/hda`



Для настройки вы можете использовать любой из приведенных примеров (или поэкспериментировать самостоятельно), при необходимости указав название своего диска вместо `/dev/hda`:

**Листинг 8:** Оптимизация скорости жесткого диска

Включение DMA:

```
# hdparm -d 1 /dev/hda
```

Установка безопасных значений:

```
# hdparm -d 1 -A 1 -m 16 -u 1 -a 64 /dev/hda
```

### Дополнительно: учетные записи пользователей

Если вы собираетесь дать другим людям доступ к своей системе, или намерены входить в чат `irssi` без прав администратора (из соображений безопасности), потребуется создать учетные записи пользователей и изменить пароль администратора.

Для изменения пароля администратора используйте `passwd`:

**Листинг 9:** Изменение пароля администратора

```
# passwd
```

New password: (введите новый пароль)

Re-enter password: (введите новый пароль еще раз)

Для создания учетной записи пользователя сначала задаются его реквизиты, а затем вводится пароль. Для этого служат программы `useradd` и `passwd`. В следующем примере мы создадим пользователя «john»:

**Листинг 10:** Создание учетной записи пользователя

```
# useradd -m -G users john
```

```
# passwd john
```

New password: (введите новый пароль для john)

Re-enter password: (введите новый пароль Джона еще раз)

Вы также можете изменить свой идентификатор пользователя с администратора (`root`) на вновь созданного пользователя командой `su`:

**Листинг 11:** Изменение идентификатора пользователя

```
# su - john
```

### Дополнительно: просмотр документации во время установки

Если вам нужна возможность просмотра настольной книги Gentoo в процессе установки (не важно, из сети или с компакт-диска), убедитесь, что вы создали учетную запись обычного пользователя (см. дополнительно: учетные записи пользователей). Затем нажмите `ALT+F2` для перехода в другой терминал, и войдите в систему.

Для чтения документации с компакт-диска, вы можете сразу запустить **links**:

**Листинг 12:** Просмотр документации с компакт-диска  
**# links /mnt/cdrom/docs/html/index.html**

Но лучше открывать настольную книгу Gentoo из сети (там она новее, чем на компакт-диске). Для этого тоже можно пользоваться **links**, но только после настройки сети (иначе вы не сможете выйти в интернет, чтобы прочесть документ):

**Листинг 13:** Просмотр документации, находящейся в сети  
**# links http://www.gentoo.org/doc/ru/handbook/handbook-x86.xml**

На исходный терминал можно переключаться нажатием **ALT+F1**.

### **Дополнительно: запуск демона SSH**

Если вы хотите разрешить другим пользователям доступ к своей системе в процессе установки Gentoo (возможно затем, чтобы они помогли вам с установкой или даже провели ее за вас), для них потребуется создать учетные записи, а возможно, даже дать им пароль администратора (делайте это только в том случае, если вы им полностью доверяете).

Чтобы запустить демон SSH, выполните следующую команду:

**Листинг 14:** Запуск демона SSH  
**# /etc/init.d/sshd start**

Для использования sshd, сначала требуется настроить сеть. Переходите к главе настройка сети.

## **3. Настройка сети**

### **3.а. Автоматическое подключение к сети**

#### ***Может, она уже настроена?***

Если ваша система подсоединена к сети Ethernet, в которой есть сервер DHCP, очень вероятно, что сетевое подключение на вашей машине уже автоматически настроено. Если так, вы сможете воспользоваться многими сетевыми командами, находящимися на установочном компакт-диске, например, **ssh**, **scp**, **ping**, **irssi**, **wget** и **links**.

Если сеть уже настроена, команда **/sbin/ifconfig** должна показывать сетевые интерфейсы кроме **lo**, например, **eth0**:

**Листинг 1:** /sbin/ifconfig для рабочей сетевой конфигурации  
**# /sbin/ifconfig**  
(...)  
eth0 Link encap:Ethernet HWaddr 00:50:BA:8F:61:7A  
inet addr:192.168.0.2 Bcast:192.168.0.255 Mask:255.255.255.0  
inet6 addr: fe80::50:ba8f:617a/10 Scope:Link  
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
RX packets:1498792 errors:0 dropped:0 overruns:0 frame:0

TX packets:1284980 errors:0 dropped:0 overruns:0 carrier:0  
collisions:1984 txqueuelen:100  
RX bytes:485691215 (463.1 Mb) TX bytes:123951388 (118.2 Mb)  
Interrupt:11 Base address:0xe800

## Дополнительно: указание прокси-серверов

Если вы подключены к интернету через прокси-сервер, при установке вам может потребоваться ввести сведения о нем. Задать прокси-сервер очень просто: нужно определить переменные, содержащие необходимые сведения.

В большинстве случаев в определении переменных достаточно указать имя прокси-сервера. Предположим, например, что прокси-сервер называется `proxy.gentoo.org`, а его порт — 8080:

**Листинг 2:** Указание прокси-сервера

```
(если прокси-сервер фильтрует трафик HTTP)
# export http_proxy="http://proxy.gentoo.org:8080"
(если прокси-сервер фильтрует трафик FTP)
# export ftp_proxy="ftp://proxy.gentoo.org:8080"
(если прокси-сервер фильтрует трафик RSYNC)
# export RSYNC_PROXY="proxy.gentoo.org:8080"
```

Если прокси-сервер запрашивает имя пользователя и пароль, для значения переменной следует использовать такой формат:

**Листинг 3:** Указание имени/пароля в адресе прокси-сервера  
`http://имя_пользователя:пароль@proxy.gentoo.org:8080`

## Проверка сети

Может оказаться полезным проверить отклик DNS-сервера вашего поставщика услуг интернета (адрес сервера находится в `/etc/resolv.conf`), а также произвольного веб-сайта, чтобы убедиться, что ваши пакеты выходят в интернет, разрешение имен DNS работает и т.д.

**Листинг 4:** Проверка доступности сети  
`# ping -c 3 google.com`

Сеть доступна? Тогда пропустите остаток этого раздела, и переходите к разделу подготовка дисков. Если сеть все же недоступна, то, к сожалению, вам придется еще поработать над ее настройкой.

## 3.b. Автоматизированная настройка сети

Если сеть не заработает сразу, при установке с некоторых носителей можно использовать **net-setup** (для обычных или беспроводных сетей), **pppoe-setup** (для пользователей ADSL) или **pptp** (для пользователей PPTP; есть на x86, amd64, alpha,

ppc и ppc64).

Если на вашем установочном диске нет ни одного из этих средств, или сеть еще не подключена, приступайте к ручной настройке сети:

- пользователи обычной сети Ethernet — переходите к разделу по умолчанию: использование **net-setup**;
- пользователи **ADSL** — переходите к разделу альтернатива: использование **RP-PPPoE**;
- пользователи **PPTP** — переходите к разделу альтернатива: использование **PPTP**;

### По умолчанию: использование net-setup

Простейший способ настройки сети, если она не произошла автоматически — запуск сценария **net-setup**:

**Листинг 5:** Запуск сценария net-setup  
**# net-setup eth0**

**net-setup** задаст вам несколько вопросов о вашей сетевой среде. В результате его работы у вас должно появиться работоспособное подключение к сети. Проверьте сетевое подключение, как это описано выше. Если проверка прошла успешно, примите наши поздравления — теперь вы готовы к установке Gentoo. Пропустите оставшуюся часть этого раздела и приступайте к подготовке дисков.

Если ваша сеть все еще не заработала, переходите к ручной настройке сети.

### Альтернатива: использование RP-PPPoE

Для простоты подключения к интернету по **PPPoE**, в установочный диск (любой версии) включен гр-pppoe. Для настройки соединения используйте сценарий rppoe-setup, входящий в комплект. У вас будет запрошена информация о сетевом устройстве ethernet, подключенном к adsl-модему, имя пользователя, пароль, IP-адреса ваших серверов DNS. Также задается вопрос, нужно ли включать базовый межсетевой экран (firewall).

**Листинг 6:** Использование гр-pppoe  
**# rppoe-setup**  
**# rppoe-start**

Если что-то пошло не так, проверьте, правильно ли вы ввели имя пользователя и пароль, посмотрев в /etc/ppp/pap-secrets или /etc/ppp/chap-secrets, и убедитесь, что устройство ethernet указано верно. Если ваше устройство не видно в системе, потребуется загрузить соответствующие сетевые модули. Для этого нужно перейти к разделу ручная настройка сети где мы объясняем, как их загрузить.

Если же все заработало, переходите к подготовке дисков.

## Альтернатива: использование PPTP

Примечание: Поддержка **PPTP** имеется только для архитектуры x86.

Если вам нужна поддержка PPTP, можно использовать **pptpclient**, который входит в состав нашего установочного диска. Но сначала нужно обеспечить правильность настройки. Отредактируйте файлы **/etc/ppp/pap-secrets** или **/etc/ppp/chap-secrets** так, чтобы в них находилось правильное сочетание имени пользователя и пароля.

**Листинг 7:** Редактирование **/etc/ppp/chap-secrets**  
**# nano -w /etc/ppp/chap-secrets**

Затем, если нужно, измените параметры PPTP в файле **/etc/ppp/options.pptp**:

**Листинг 8:** Редактирование **/etc/ppp/options.pptp**  
**# nano -w /etc/ppp/options.pptp**

Когда все будет готово, просто запустите **pptp** (с параметрами, которые вы не стали прописывать в **options.pptp**), чтобы соединиться с сервером:

**Листинг 9:** Подключение к серверу коммутируемого доступа  
**# pptp <server ip>**

Теперь переходите к подготовке дисков.

## 3.с. Ручная настройка сети

### Загрузка нужных сетевых модулей

При загрузке установочный диск пытается выявить все установленные устройства и загружает подходящие модули ядра (драйверы) для поддержки вашего оборудования. В подавляющем большинстве случаев он очень хорошо справляется с этой работой. Однако, в некоторых случаях он может не справиться с автозагрузкой нужных вам модулей ядра.

Если **net-setup** или **pppoe-setup** не удалось загрузить нужный модуль, возможно, ваша сетевая плата сразу не обнаружена. При этом вам может потребоваться ручная загрузка необходимых модулей ядра.

Чтобы выяснить, какие модули ядра для поддержки сети существуют, используйте **ls**:

**Листинг 10:** Поиск имеющихся модулей  
**# ls /lib/modules/ `uname -r`/kernel/drivers/net**

Если вы найдете драйвер для своей сетевой платы, для загрузки модуля ядра используйте **modprobe**:

**Листинг 11:** Использование **modprobe** для загрузки модуля ядра  
(для примера загрузим модуль **pcnet32**)  
**# modprobe pcnet32**

Чтобы убедиться, что ваша сетевая плата теперь обнаружена, используйте **ifconfig**. Если сетевая плата обнаружена, результат выглядит подобным образом:

**Листинг 12:** Проверка доступности сетевой платы (удачная)

```
# ifconfig eth0
eth0    Link encap:Ethernet  HWaddr FE:FD:00:00:00:00
        BROADCAST NOARP MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Однако, если вы получите такое сообщение об ошибке, сетевая плата не обнаружена:

**Листинг 13:** Проверка доступности сетевой платы (неудачная)

```
# ifconfig eth0
eth0: error fetching interface information: Device not found
```

Если в вашей системе установлено несколько сетевых плат, они будут называться **eth0**, **eth1** и т.д. Убедитесь, что сетевая плата, которую вы собираетесь использовать, работает хорошо, и в дальнейшем не забудьте везде подставлять верное имя. Мы предполагаем, что используется сетевая плата **eth0**.

Когда ваша сетевая плата обнаружена, можно попробовать снова запустить **net-setup** или **pppoe-setup** (которые теперь должны сработать), но на случай, если вы из крутых, мы опишем, как настроить сеть вручную.

Выберите один из следующих разделов, в зависимости от необходимого вида настройки:

- [использование DHCP для автоматического присвоения IP-адреса;](#)
- [подготовка беспроводного доступа, если у вас есть беспроводная плата;](#)
- [освоение сетевой терминологии: рассказ о том, что нужно знать о подключении к сети;](#)
- [использование ifconfig и route: описание ручной настройки сети;](#)

## Использование DHCP

**DHCP** (Dynamic Host Configuration Protocol — протокол динамической настройки хоста) дает возможность автоматически получить параметры сетевого подключения (IP-адрес, маску сети, широковещательный адрес, шлюз, сервера имен и т. д.) Все это работает, только когда в вашей сети есть сервер DHCP (или ваш поставщик предоставляет услугу DHCP). Чтобы сетевой интерфейс получал эти сведения автоматически, используйте **dhcpcd**:

**Листинг 14:** Использование dhcpcd

```
# dhcpcd eth0
```

Некоторые сетевые администраторы требуют, чтобы вы использовали имя хоста и домена, назначенное сервером DHCP.

В этом случае используйте  
**# dhcpcd -HD eth0**

Если это сработало (попробуйте «попинговать» какой-нибудь сервер интернета, например, Google), то у вас все настроено, и можно двигаться дальше. Пропустите остаток этого раздела и приступайте к подготовке дисков.

## Подготовка беспроводного доступа

**Примечание:** Поддержка команды **iwconfig** есть только на установочных дисках для платформ x86, amd64 и ppc. В других случаях вы все же сможете запустить расширения, следуя инструкциям проекта [linux-wlan-ng](#) (англ.).

Если вы используете беспроводную плату (802.11), перед дальнейшими действиями может потребоваться настройка параметров беспроводного подключения. Для просмотра текущей настройки беспроводной платы можете использовать **iwconfig**. При запуске **iwconfig** вы увидите подобные сведения:

**Листинг 15:** Вывод текущих параметров беспроводного подключения

```
# iwconfig eth0
eth0 IEEE 802.11-DS ESSID:"GentooNode"
    Mode:Managed Frequency:2.442GHz Access Point: 00:09:5B:11:CC:F2
    Bit Rate:11Mb/s Tx-Power=20 dBm Sensitivity=0/65535
    Retry limit:16 RTS thr:off Fragment thr:off
    Power Management:off
    Link Quality:25/10 Signal level:-51 dBm Noise level:-102 dBm
    Rx invalid nwid:5901 Rx invalid crypt:0 Rx invalid frag:0 Tx
    excessive retries:237 Invalid misc:350282 Missed beacon:84
```

**Примечание:** Некоторые беспроводные платы могут называться подобно wlan0 или ra0, а не eth0. Для определения верного имени устройства запускайте **iwconfig** без каких-либо параметров командной строки.

Для большинства пользователей может потребоваться изменение только двух параметров: ESSID (названия беспроводной сети) и ключа шифрования WEP. Если ESSID и указанный адрес точки доступа уже принадлежат вашей точке доступа, и вы не используете WEP, значит, ваше беспроводное подключение работает. Если вам необходимо изменить свой ESSID или добавить ключ WEP, можно запустить следующие команды:

**Листинг 16:** Замена ESSID и/или добавление ключа WEP  
(так название сети устанавливается в "GentooNode")

```
# iwconfig eth0 essid GentooNode
```

(так устанавливается шестнадцатиричный ключ WEP)

```
# iwconfig eth0 key 1234123412341234abcd
```

(так устанавливается текстовый ключ (ASCII); вначале нужно добавлять "s:")

```
# iwconfig eth0 key s:some-password
```

Проверить сделанную настройку можно, снова запустив **iwconfig**. Как только ваша беспроводная сеть заработает, вы можете перейти к установке параметров сети,

относящихся к протоколу IP, которые описываются в следующем разделе (освоение сетевой терминологии), или использовать утилиту **net-setup**, как описано ранее.

## Освоение сетевой терминологии

**Примечание:** Если вы знаете свой IP-адрес, широковещательный адрес, маску сети и серверы имен, можете пропустить этот подраздел и перейти к разделу использование **ifconfig** и **route**.

Если все, рассказанное выше, не помогло, вам придется настроить свою сеть вручную. Это совсем нетрудно. Однако, вам понадобится освоить кое-какую сетевую терминологию, знание которой требуется для удовлетворительной настройки сети. Прочитав этот текст, вы узнаете, что такое шлюз, зачем служит маска сети, как формируется широковещательный адрес, и зачем нужны серверы имен.

В сети узлы (хосты, компьютеры) различаются по IP-адресу (адресу протокола интернета). Такой адрес — это сочетание четырех чисел от 0 до 255. Ну, по крайней мере, так мы его воспринимаем. В действительности, IP-адрес состоит из 32 бит (единиц и нулей). Давайте рассмотрим пример:

**Листинг 17:** Пример IP-адреса

```
IP-адрес (числа): 192.168.0.2
IP-адрес (биты): 11000000 10101000 00000000 00000010
                  -----
                  192   168   0     2
```

Такой IP-адрес уникален для узла в рамках всех доступных сетей (т. е. каждый узел, с которым вы можете связаться, должен иметь уникальный IP-адрес). Чтобы различать узлы, находящиеся внутри и извне сети, IP-адрес подразделяется на две части: часть сети и часть узла.

Это разделение записывается с помощью маски сети — набора единиц, за которым следует набор нулей. Часть IP-адреса, которая попадает на единицы — сетевая, оставшаяся часть — узловая. Как обычно, маска сети может записываться в виде IP-адреса.

**Листинг 18:** Пример разделения сети/узла

```
IP-адрес: 192 168 0 2
          11000000 10101000 00000000 00000010
Маска сети: 11111111 11111111 11111111 00000000
            255 255 255 0
            +-----+-----+
            Сеть   Узел
```

Другими словами, 192.168.0.14 — все еще входит в состав сети из нашего примера, а 192.168.1.2 — уже нет.

Широковещательный адрес — это IP-адрес с такой же сетевой частью, как у вашей сети, но у которого узловая часть состоит только из единиц. Каждый узел вашей сети слушает этот IP-адрес. Он действительно предназначен для широковещательной рассылки пакетов.



**Листинг 19:** Широковещательный адрес

```

IP-адрес:      192   168   0   2
               11000000 10101000 00000000 00000010
Широковещательный 11000000 10101000 00000000 11111111
адрес:         192   168   0   255
               +-----+-----+
                  Сеть      Узел

```

Чтобы бороздить просторы интернета, вы должны знать, через какой узел происходит подключение к интернету. Этот узел называется шлюзом. Так как это обычный узел, ему присвоен обычный IP-адрес (например, 192.168.0.1).

Выше мы говорили, что каждому узлу присваивается свой собственный IP-адрес. Чтобы связываться с узлом по имени (вместо IP-адреса), нужна служба, которая переводит имя (такое, как dev.gentoo.org) в IP-адрес (например, 64.5.62.82). Такая служба называется службой имен. Чтобы пользоваться ей, нужно указать необходимые серверы имен в /etc/resolv.conf.

Иногда ваш шлюз сам является сервером имен. В остальных случаях вам необходимо указывать серверы имен, предоставляемые поставщиком интернета.

В итоге, для дальнейшего вам потребуется следующая информация:

Параметр настройки сети	Пример
Ваш IP-адрес	192.168.0.2
Маска сети	255.255.255.0
Широковещательный адрес	192.168.0.255
Шлюз	192.168.0.1
Сервер(ы) имен	195.130.130.5, 195.130.130.133

## Использование ifconfig и route

Настройка вашей сети состоит из трех шагов. Сначала мы назначаем себе IP-адрес с помощью **ifconfig**. Затем мы настраиваем маршрутизацию к шлюзу, пользуясь **route**. И в завершение мы помещаем IP-адреса серверов имен в /etc/resolv.conf.

Для назначения IP-адреса потребуется ваш IP-адрес, широковещательный адрес и маска сети. Узнав их, выполните следующую команду, заменив `${IP_ADDR}` на свой IP-адрес, `${BROADCAST}` на свой широковещательный адрес, а `${NETMASK}` на свою маску сети:

**Листинг 20:** Использование ifconfig

```
# ifconfig eth0 ${IP_ADDR} broadcast ${BROADCAST} netmask ${NETMASK} up
```

Теперь настройте маршрутизацию с помощью **route**. Подставьте IP-адрес своего шлюза вместо `${GATEWAY}`:

**Листинг 21:** Использование route

```
# route add default gw ${GATEWAY}
```

Затем откройте **/etc/resolv.conf** в своем любимом редакторе (в нашем примере используется nano):

**Листинг 22:** Создание /etc/resolv.conf  
**# nano -w /etc/resolv.conf**

Заполните данные своих серверов имен по следующему образцу. Обязательно замените **\${NAMESERVER1}** и **\${NAMESERVER2}** на соответствующие адреса серверов имен:

**Листинг 23:** Образец /etc/resolv.conf  
nameserver \${NAMESERVER1}  
nameserver \${NAMESERVER2}

Готово. Теперь проверьте свою сеть, «попинговав» какой-либо сервер интернета (например, Google). Если все заработало, примите наши поздравления! Теперь вы готовы к установке Gentoo. Приступайте к подготовке дисков.

## **4. Подготовка дисков**

### **4.а. Общие сведения о блочных устройствах**

#### **Блочные устройства**

Мы достаточно подробно рассмотрим аспекты работы с дисками в Gentoo Linux и Linux вообще, включая файловые системы, разделы и блочные устройства. Позже, когда вы уже освоитесь с дисками и файловыми системами, мы проведем вас через процесс настройки разделов и файловых систем для установки Gentoo Linux.

Для начала, давайте познакомимся с блочными устройствами. Вероятно, наиболее известно блочное устройство, соответствующее в системе Linux первому IDE-диску, а именно /dev/hda. А если в вашей системе используются диски SCSI или SATA, то первым жестким диском будет /dev/sda.

Подобные блочные устройства представляют собой абстрактный интерфейс к диску. Прикладные программы могут использовать их для обращения к дискам, не беспокоясь, к какому типу те принадлежат: IDE, SCSI или какому-то еще. Программы могут просто обращаться к накопителю, как к набору смежных 512-байтных блоков с произвольным доступом.

#### **Разделы**

Хотя теоретически для размещения системы Linux диск можно использовать целиком, так почти никогда не делают на практике. Вместо этого большое физическое блочное устройство разбивают на меньшие блочные устройства, более удобные для обращения. В архитектуре x86 они называются разделами.

Существуют разделы трех типов: первичные (**primary**), расширенные (**extended**) и логические (**logical**).

Первичный раздел — это раздел, информация о котором хранится в **MBR** (Master Boot Record — главной загрузочной записи). Так как MBR очень мала (512 байт), можно определять всего четыре первичных раздела (например, от /dev/hda1 до /dev/hda4).

Расширенный раздел — это особый первичный раздел (имеется в виду, что расширенный раздел должен быть одним из четырех возможных первичных разделов), в котором содержатся другие разделы. Таких разделов изначально не существовало, но их введение помогло расширить существующую схему разметки без потери совместимости, как только четырех разделов перестало хватать.

Логический раздел — это раздел, входящий в расширенный раздел. Его определение находится не в **MBR**, а внутри расширенного раздела.

### Дополнительные возможности

На установочном компакт-диске для x86 предусмотрена поддержка EVMS и LVM2. Использование EVMS и LVM2 повышает гибкость разбиения диска. В ходе установки мы останавливаемся на «обычных» разделах, но стоит запомнить, что EVMS и LVM2 тоже поддерживаются.

## 4.b. Разработка схемы разбиения диска

### Схема разбиения по умолчанию

Если вам не интересно заниматься разработкой схемы для своей системы, можно воспользоваться схемой, используемой в этой книге:

Раздел	Файловая система	Размер	Описание
/dev/hda1	ext2	32МБ	загрузочный раздел
/dev/hda2	(swap)	512МБ	раздел подкачки
/dev/hda3	ext3	оставшаяся часть диска	корневой раздел

Если вам интересно узнать, какого размера должны быть разделы, и сколько их вам вообще может потребоваться, читайте дальше. В противном случае приступайте к созданию разделов, описанному в главе использование **fdisk** для создания разделов.

### Сколько и каких именно?

Количество разделов очень сильно зависит от назначения системы. Например, если у вас много пользователей, вам, скорее всего, захочется отделить /home для повышения безопасности и упрощения резервного копирования. Если вы устанавливаете Gentoo в роли почтового сервера, следует отделить /var, т.к. вся почта хранится там. Затем, правильно выбрав файловую систему, вы добьетесь максимальной производительности. Для игровых серверов потребуется отдельный раздел /opt, так как большинство программ для их работы устанавливается туда. Причины выделения те же, что и для /home: безопасность и резервное копирование.

Определенно не помешает побольше места для /usr: не только потому, что там хранится большинство приложений, а еще из-за того, что лишь дерево Portage, не считая размещенных в нем архивов с исходными кодами, занимает около 500 МБ.

Как видите, все зависит от ваших целей. Наличие отдельных разделов или томов имеет следующие плюсы:

- для каждого раздела или тома можно выбрать наиболее подходящую файловую систему;
- свободное место во всей системе не кончится «вдруг» из-за того, что одна-единственная сбойная программа постоянно записывает файлы в раздел или том
- необходимая проверка файловых систем будет занимать меньше времени, т.к. проверка разных разделов может выполняться параллельно (еще больший выигрыш времени дает использование нескольких физических дисков) ;
- можно повысить безопасность системы, монтируя часть разделов в режиме read-only (только для чтения), **nosuid** (игнорируется бит setuid), noexec (игнорируется бит исполнения) и т.д..

Однако, у создания множества разделов есть один большой минус: при неправильной настройке можно получить систему, в которой много свободного места на одном разделе, и совершенно нет на другом. Кроме того, на дисках SCSI и SATA возможно создание не более 15 разделов.

Для примера мы покажем разбиение диска объемом 20 ГБ, используемого в демонстрационном ноутбуке (с веб-сервером, почтовым сервером, средой Gnome и т.д.):

**Листинг 1:** Пример файловой системы

**\$ df -h**

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
/dev/hda5	ext3	509M	132M	351M	28%	/
/dev/hda2	ext3	5.0G	3.0G	1.8G	63%	/home
/dev/hda7	ext3	7.9G	6.2G	1.3G	83%	/usr
/dev/hda8	ext3	1011M	483M	477M	51%	/opt
/dev/hda9	ext3	2.0G	607M	1.3G	32%	/var
/dev/hda1	ext2	51M	17M	31M	36%	/boot
/dev/hda6	swap	516M	12M	504M	2%	<not mounted>

(свободное место для будущего использования: 2 ГБ)

Раздел /usr, как видим, почти заполнен (использовано 83%), но когда все программы установлены, /usr растет не слишком быстро. Хотя отведение нескольких гигабайт дискового пространства для /var может показаться расточительством, помните, что этот раздел по умолчанию используется Portage для компиляции пакетов. Если вы захотите удержать /var в рамках более разумного размера, например, 1ГБ, вам потребуется изменить переменную **PORTAGE\_TMPDIR** в /etc/make.conf, чтобы она указывала на раздел, где достаточно свободного места для компиляции чрезвычайно больших пакетов, таких как OpenOffice.

#### 4.с. Использование fdisk для создания разделов

В следующих разделах описывается, как разбить диск в соответствии с примерной схемой, описанной ранее, а именно:

Раздел	Описание
/dev/hda1	загрузочный раздел
/dev/hda2	раздел подкачки
/dev/hda3	корневой раздел

Эту схему вы можете изменять по своему усмотрению.

### Просмотр текущей схемы разбиения диска

**fdisk** — это популярная и очень мощная утилита для создания разделов на ваших дисках. Запустите **fdisk**, указав свой диск в качестве параметра (в примере мы используем `/dev/hda`):

**Листинг 2:** Запуск **fdisk**  
**# fdisk /dev/hda**

После запуска **fdisk** выдаст такое приглашение:

**Листинг 3:** Приглашение **fdisk**  
**Command (m for help):**

Нажмите **p**, чтобы вывести текущую схему разбиения диска:

**Листинг 4:** Пример схемы разделов диска  
**Command (m for help): p**

Disk /dev/hda: 240 heads, 63 sectors, 2184 cylinders  
Units = cylinders of 15120 \* 512 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1		1	14	105808+	83	Linux
/dev/hda2		15	49	264600	82	Linux swap
/dev/hda3		50	70	158760	83	Linux
/dev/hda4		71	2184	15981840	5	Extended
/dev/hda5		71	209	1050808+	83	Linux
/dev/hda6		210	348	1050808+	83	Linux
/dev/hda7		349	626	2101648+	83	Linux
/dev/hda8		627	904	2101648+	83	Linux
/dev/hda9		905	2184	9676768+	83	Linux

Command (m for help):

В данном случае на диске есть семь разделов Linux (для которых в графе «System» указано «Linux») и один раздел подкачки (в списке показан как «Linux swap»).

### Удаление всех разделов

Сначала удалим с диска все существующие разделы. Для удаления раздела вводите **d**. Например для удаления существующего `/dev/hda1`:

**Листинг 5:** Удаление раздела

**Command (m for help): d**

Partition number (1-4): 1

Удаление раздела будет запланировано. Он больше не будет отображаться при вводе **p**, но фактически не будет удален до тех пор, пока вы не сохраните внесенные изменения. Если вы ошиблись и хотите прервать разбиение без сохранения изменений, немедленно введите **q** и нажмите ENTER; тогда ваш раздел сохранится в неизменном виде.

Теперь, если вы действительно собираетесь удалить в своей системе все разделы, повторяйте ввод **p** для показа оставшихся разделов, затем **d** и номера удаляемого раздела до тех пор, пока разделы не кончатся. В итоге вы получите пустую таблицу разделов:

**Листинг 6:** Пустая таблица разделов

Disk /dev/hda: 30.0 GB, 30005821440 bytes

240 heads, 63 sectors/track, 3876 cylinders

Units = cylinders of 15120 \* 512 = 7741440 bytes

Device	Boot	Start	End	Blocks	Id	System
--------	------	-------	-----	--------	----	--------

Command (m for help):

Теперь, когда мы очистили таблицу разделов, хранящуюся в оперативной памяти, настало время создавать разделы. Мы воспользуемся примерной схемой разбиения, описанной выше. Естественно, не следуйте этим инструкциям буквально, если только вам не нужна таблица разделов, идентичная нашей!

## Создание загрузочного раздела

Сначала создадим маленький загрузочный раздел. Введите **n** для создания нового раздела, затем **p**, чтобы выбрать первичный раздел, и **1** для указания первого первичного раздела. На вопрос о первом цилиндре нажмите ввод. На вопрос о последнем цилиндре введите **+32M**, чтобы создать раздел размером **32МБ**:

**Листинг 7:** Создание загрузочного раздела

**Command (m for help): n**

Command action

e extended

p primary partition (1-4)

p

Partition number (1-4): 1

First cylinder (1-3876, default 1): (нажмите ввод)

Using default value 1

Last cylinder or +size or +sizeM or +sizeK (1-3876, default 3876): +32M

Теперь, введя **p**, вы должны увидеть следующий текст:

**Листинг 8:** Созданный загрузочный раздел

**Command (m for help): p**

Disk /dev/hda: 30.0 GB, 30005821440 bytes

240 heads, 63 sectors/track, 3876 cylinders

Units = cylinders of 15120 \* 512 = 7741440 bytes

Device	Boot	Start	End	Blocks	Id	System
--------	------	-------	-----	--------	----	--------

```
/dev/hda1      1      14  105808+ 83 Linux
```

Нам необходимо сделать этот раздел загрузаемым. Введите **a** для включения признака загрузки с раздела, затем нажмите **1**. Если снова ввести **p**, можно увидеть, что в столбце «boot» появился знак **\***.

### Создание раздела подкачки

Теперь создадим раздел подкачки. Для этого введите **n** (создание нового раздела), затем **p** для выбора первичного раздела. Потом нажмите **2**, чтобы создать второй первичный раздел, в нашем случае **/dev/hda2**. На вопрос о первом цилиндре просто нажмите ввод. На вопрос о последнем — ответьте **+512M**, чтобы создать раздел размером 512МБ. Сделав это, введите **t** для указания типа раздела, **2**, для выбора только что созданного раздела, и **82**, чтобы установить тип раздела «Linux Swap». При нажатии **p** по завершении этих действий, таблица разделов должна выглядеть примерно так:

**Листинг 9:** Список разделов после создания раздела подкачки  
**Command (m for help): p**

```
Disk /dev/hda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	14	105808+	83	Linux
/dev/hda2		15	81	506520	82	Linux swap

### Создание корневого раздела

Теперь создадим корневой раздел. Для этого введите **n** (создание нового раздела), затем **p** (первичный раздел). После этого нажмите **3** для создания третьего первичного раздела, в нашем случае **/dev/hda3**. На вопрос о первом цилиндре нажмите ввод. На вопрос о последнем — также нажмите ввод, чтобы раздел занял все оставшееся свободное место на диске. По завершении этих шагов, при вводе **p** должна выводиться подобная таблица разделов:

**Листинг 10:** Список разделов после создания корневого раздела  
**Command (m for help): p**

```
Disk /dev/hda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	14	105808+	83	Linux
/dev/hda2		15	81	506520	82	Linux swap
/dev/hda3		82	3876	28690200	83	Linux

### Сохранение схемы разбиения

Для сохранения схемы разбиения и выхода из **fdisk**, введите **w**.

**Листинг 11:** Сохранение и выход из **fdisk**  
**Command (m for help): w**

Теперь, создав все разделы, перейдем к созданию файловых систем.

## 4.d. Создание файловых систем

### Введение

Разделы созданы, настало время разместить на них файловые системы. Если вам безразлично, какую файловую систему использовать, и вы вполне довольны той, что мы предлагаем в книге по умолчанию, переходите к размещению файловой системы в разделе. Если нет, читайте дальше, чтобы разузнать о существующих файловых системах...

### Файловые системы (сравнение !)

В ядре Linux поддерживаются различные файловые системы. Мы расскажем о самых распространенных из них: **ext2**, **ext3**, **ReiserFS**, **XFS** и **JFS**.

- **ext2** — испытанная файловая система Linux, в которой отсутствуют средства журналирования метаданных. Поэтому ее рядовая проверка при запуске может быть очень долгой. Сегодня существует довольно много журналируемых файловых систем нового поколения, целостность которых проверяется очень быстро, и поэтому обычно есть смысл использовать их. Журналируемые файловые системы позволяют избежать долгих задержек при запуске, когда состояние файловой системы неустойчиво.
- **ext3** — это журналируемая версия ext2, где для ускорения восстановления появилось журналирование метаданных, а также другие режимы, наподобие полного или упорядоченного журналирования данных. ext3 — очень хорошая и надежная файловая система. В ней есть дополнительная возможность индексации через хэшированные сбалансированные деревья (b-tree), что в большинстве случаев обеспечивает высокую скорость. Такую индексацию можно включить, добавив к команде `mke2fs` параметр `-O dir_index`. Короче говоря, ext3 — превосходная файловая система.
- **ReiserFS** — файловая система, построенная на B\*-деревьях. У нее очень хорошая скорость, и она намного (часто в 10-15 раз) быстрее ext2 и ext3 при работе с маленькими файлами (менее 4КБ). ReiserFS также великолепно масштабируется; в ней есть средства журналирования. На дату выхода ядра 2.4.18, ReiserFS признана стабильной и пригодной как для систем общего назначения, так и для крайних случаев типа создания больших томов, использования множества маленьких файлов, гигантских файлов или же каталогов с десятками тысяч файлов.
- **XFS** — это файловая система с журналированием метаданных и дополнительными средствами, оптимизированными для работы с максимальным объемом данных. Ее применение рекомендуется только в Linux-системах с высококлассными дисками SCSI или дисками с оптическим подключением (fibre-channel), при наличии источников бесперебойного питания. Из-за того, что XFS выполняет очень агрессивное промежуточное кэширование в оперативной памяти, при внезапном отключении питания очень вероятно потеря небрежно спроектированными программами (не



принимающими адекватные меры предосторожности при записи файлов на диск, и таких довольно много) изрядного количества данных.

- **JFS** — высокопроизводительная журналируемая файловая система производства IBM. Она лишь недавно стала пригодной для широкого применения, и пока слишком мало данных, чтобы сказать что-то хорошее или плохое о ее общей стабильности.

## Размещение файловой системы в разделе

Для создания в разделе или томе файловой системы каждого типа существуют специальные средства:

Файловая система	Команда создания
ext2	mke2fs
ext3	mke2fs -j
reiserfs	mkreiserfs
xf	mkfs.xfs
jfs	mkfs.jfs

Например, чтобы у загрузочного раздела (/dev/hda1 в наших примерах) была файловая система ext2, а у корневого раздела (/dev/hda3 в наших примерах) — ext3, требуется выполнить:

**Листинг 12:** Создание файловых систем разделов  
# mke2fs /dev/hda1  
# mke2fs -j /dev/hda3

Теперь самостоятельно создайте файловые системы на своих только что созданных разделах (логических томах).

## Подключение раздела подкачки

**mkswap** — команда, используемая для инициализации разделов подкачки:

**Листинг 13:** Создание идентификатора раздела подкачки  
# mkswap /dev/hda2

Для подключения раздела подкачки воспользуйтесь **swapon**:

**Листинг 14:** Подключение раздела подкачки  
# swapon /dev/hda2

Теперь командами, приведенными выше, создайте и подключите раздел подкачки в своей системе.

## 4.e. Монтирование

Теперь, когда разделы созданы, а файловые системы размещены, настало время смонтировать (подключить к системе) эти разделы. Используйте команду **mount**. Не забудьте создать для каждого раздела соответствующие каталоги монтирования. Например, смонтируем корневой и загрузочный разделы:

**Листинг 15:** Монтирование разделов  
**# mount /dev/hda3 /mnt/gentoo**  
**# mkdir /mnt/gentoo/boot**  
**# mount /dev/hda1 /mnt/gentoo/boot**

**Примечание:** Если вы хотите разместить каталог `/tmp` в отдельном разделе, не забудьте изменить права доступа к этому каталогу после монтирования: **`chmod 1777 /mnt/gentoo/tmp`**. Это также относится к `/var/tmp`.

Еще нам потребуется смонтировать файловую систему **proc** (виртуальный интерфейс ядра) в каталог **/proc**. Но сначала надо поместить в разделы нужные файлы.

Переходите к установке установочных файлов Gentoo.

## 5. Установка установочных файлов Gentoo

### 5.а. Установка архива стадии

#### Установка текущей даты и времени

Перед тем, как продолжать, убедитесь в правильности системной даты и времени, и при необходимости обновите их. Если часы установлены неверно, это в будущем может привести к странным результатам!

Для проверки времени, введите команду **date**:

**Листинг 1:** Проверка даты и времени  
**# date**  
Fri Mar 29 16:21:18 UTC 2007

Если часы установлены неправильно, измените время командой **date ММДДччммГГГГГ** (формат: Месяц, День, часы, минуты и Год). На этом этапе следует указывать время UTC. Позднее вы сможете установить свой часовой пояс. Например, для установки часов на 29 марта, 16:21, 2005 год, выполните:

**Листинг 2:** Установка даты и времени  
**# date 032916212007**

#### Выбор источника

Следующий необходимый шаг — установка архива **stage3** на вашу систему. Нужный архив можно скачать из интернета, или, если вы загрузились с универсального установочного диска Gentoo, скопировать с самого диска. Если у вас есть универсальный диск, а на нем — нужная стадия, то загрузка из интернета — лишняя трата трафика и времени, т.к. файлы стадии окажутся совершенно одинаковы. В большинстве случаев, определиться с выбором архива стадии вам поможет команда

`uname -m`.

**по умолчанию:** использование файла стадии из интернета

**альтернатива:** использование файла стадии с установочного диска

## 5.b. По умолчанию: использование файла стадии из интернета

### Загрузка архива стадии

Перейдите в точку подключения файловой системы Gentoo (обычно — `/mnt/gentoo`):

**Листинг 3:** Переход к точке подключения Gentoo  
`# cd /mnt/gentoo`

Для загрузки файла стадии у вас уже есть различные инструменты, зависящие от типа диска, с которого производится установка. Если имеется `links`, можете сразу открыть список зеркал Gentoo и выбрать ближайшее.

Когда `links` нет, в вашем распоряжении должен оказаться `lynx`. Если требуется использовать прокси-сервер, экспортируйте переменные `http_proxy` и `ftp_proxy`:

**Листинг 4:** Настройка информации о прокси-сервере для `lynx`  
`# export http_proxy="http://proxy.server.com:port"`  
`# export ftp_proxy="http://proxy.server.com:port"`

Предположим, у вас под рукой есть `links`.

Перейдите в каталог `releases/`, в нем откройте каталог, соответствующий вашей архитектуре (например, `x86/`), затем каталог с версией Gentoo (`2006.1/`), и, наконец, каталог `stages/`. Здесь находятся все доступные архивы стадий для вашей архитектуры (они могут располагаться в подкаталогах с названиями разновидностей архитектуры). Выберите один из них и нажмите **D** для загрузки. По окончании — нажмите **Q**, чтобы выйти из браузера.

**Листинг 5:** Открытие списка зеркал в `links`  
`# links http://www.gentoo.org/main/ru/mirrors.xml`

(если в `links` нужна поддержка прокси-сервера:)  
`# links -http-proxy proxy.server.com:8080 http://www.gentoo.org/main/ru/mirrors.xml`

Удостоверьтесь, что вы скачали **stage3**-архив — установка с использованием файлов **stage1** и **stage2** более не поддерживается.

Если вы хотите проверить целостность загруженного архива стадии, с помощью `md5sum` сравните результат с контрольной суммой MD5, взятой с зеркала. Например, для проверки целостности архива стадии для `x86`:

**Листинг 6:** Пример проверки целостности архива стадии  
`# md5sum -c stage3-x86-2007.0.tar.bz2.DIGESTS`  
`stage3-x86-2007.0.tar.bz2: OK`

## Распаковка архива стадии

Распакуйте загруженный архив стадии в своей системе. Мы используем tar, т.к. это простейший способ:

**Листинг 7:** Распаковка архива стадии

```
# tar xvjpf stage3-*.tar.bz2
```

Убедитесь, что вы используете точно такие же параметры командной строки (**xvjpf**). Значения параметров:

- **x** — *извлечение*;
- **v** — *подробные сообщения, чтобы видеть, что происходит во время распаковки (необязательный параметр)*;
- **j** — *декомпрессия bzip2*;
- **p** — *сохранение прав доступа*;
- **f** — *указывает на то, что мы распаковываем файл, а не то, что подается на стандартный ввод*.

**Примечание:** На установочных компакт-дисках и загрузочных образах для некоторых архитектур (например, MIPS) tar встроен в оболочку BusyBox, которая на данный момент не поддерживает параметр **v**. Тогда вместо приведенных указывайте параметры **xjpf**.

После развертывания установки стадии переходите к установке дерева портежей.

## 5.с. Альтернатива: использование файла стадии с установочного диска

### Извлечение архива стадии

Важно: Если вы — в x86 и используете «живой диск» с установщиком, на нем нет стадий. Вам понадобится последовать указаниям по использованию файла стадии из интернета.

Архивы стадий находятся на компакт-диске в каталоге `/mnt/cdrom/stages`. Для получения списка доступных стадий, воспользуйтесь **ls**:

**Листинг 8:** Список доступных архивов стадий

```
# ls /mnt/cdrom/stages
```

Если система сообщает об ошибке, возможно, сначала нужно смонтировать CD-ROM:

**Листинг 9:** Монтирование CD-ROM

```
# ls /mnt/cdrom/stages
```

```
ls: /mnt/cdrom/stages: No such file or directory
```

```
# mount /dev/cdroms/cdrom0 /mnt/cdrom
```

```
# ls /mnt/cdrom/stages
```

Перейдите в точку подключения Gentoo (обычно — /mnt/gentoo):

**Листинг 10:** Смена каталога на /mnt/gentoo  
**# cd /mnt/gentoo**

Теперь утилитой `tar` распакуем выбранный архив стадии. Убедитесь, что вы используете точно такие же параметры (`xvjpf`)! Параметр `v` необязателен и может не работать в некоторых версиях `tar`. В следующем примере мы распакуем архив стадии `stage3-<архитектура>-2006.1.tar.bz2`. В качестве имени архива обязательно указывайте название файла нужной вам стадии.

**Листинг 11:** Распаковка архива стадии  
**# tar xvjpf /mnt/cdrom/stages/stage3-<архитектура>-2007.0.tar.bz2**

После установки стадии переходите к установке дерева портежей.

## 5.d. Установка дерева портежей

### Распаковка снимка портежей

Теперь вам нужно установить снимок дерева портежей — набор файлов, содержащих сведения для Portage о программном обеспечении, доступном для установки, имеющихся профилях и т.д.

### Загрузка и установка снимка дерева портежей

Перейдите к точке подключения вашей системы (обычно — /mnt/gentoo):

**Листинг 12:** Переход к точке подключения  
**# cd /mnt/gentoo**

Запустите **links** (или **lynx**) и откройте список зеркал Gentoo. Выберите ближайшее зеркало и перейдите в каталог `snapshots/`. Оттуда загрузите новейший снимок дерева портежей (`portage-latest.tar.bz2`), выбрав его из списка и нажав **D**.

**Листинг 13:** Просмотр списка зеркал Gentoo  
**# links http://www.gentoo.org/main/ru/mirrors.xml**

Закройте браузер нажатием **Q**. Теперь у вас в /mnt/gentoo появился снимок дерева портежей.

Если нужно убедиться в целостности загруженного снимка, воспользуйтесь `md5sum`, и сравните результат с контрольной суммой MD5, имеющейся на зеркале.

**Листинг 14:** Checking integrity of a Portage snapshot  
**# md5sum -c portage-latest.tar.bz2.md5sum**  
portage-latest.tar.bz2: OK

Теперь нужно распаковать снимок. Убедитесь в том, что вы используете в точности

такие же параметры; последний параметр — заглавная **C**, а не строчная **c**.

**Листинг 15:** Извлечение снимка дерева портежей

```
# tar xvjf /mnt/gentoo/portage-latest.tar.bz2 -C /mnt/gentoo/usr
```

## 5.e. Настройка параметров компиляции

### Введение

Для оптимизации Gentoo вы можете создать ряд переменных, которые повлияют на поведение Portage. Все их можно создавать как переменные среды (с помощью `export`), но тогда они будут лишь временными. Для хранения ваших настроек предназначен конфигурационный файл Portage, `/etc/make.conf`. Именно его мы сейчас будем редактировать.

**Примечание:** Список и описание всех допустимых переменных находятся в файле `/mnt/gentoo/etc/make.conf.example`. Для успешной установки Gentoo, достаточно установить значения только тех переменных, которые описаны ниже.

Запустите ваш любимый редактор (в этом руководстве используется **nano**) для изменения параметров оптимизации, которые обсуждаются далее.

**Листинг 16:** Открытие `/etc/make.conf`

```
# nano -w /mnt/gentoo/etc/make.conf
```

Как вы могли заметить, у файла **make.conf.example** обычная структура: строки комментария начинаются со знака «**#**», а в остальных строках выражениями вида **ПЕРЕМЕННАЯ**=**"значение"** определяются переменные. У файла `make.conf` такой же формат. Некоторые из переменных мы обсудим ниже.

### CHOST

В переменной **CHOST** определяется, для какой архитектуры скомпилирована система. В ней уже должно быть установлено правильное значение. Не изменяйте эту переменную, т.к. это может повредить вашу систему. Если переменная **CHOST**, по-вашему, выглядит неправильно, возможно, вы используете не тот архив `stage3`.

### CFLAGS и CXXFLAGS

Переменные **CFLAGS** и **CXXFLAGS** определяют параметры оптимизации компилятора `gcc` для языков **C** и **C++**, соответственно. При том, что общие значения следует устанавливать здесь, максимальная производительность достигается в том случае, когда для каждой компилируемой программы устанавливаются свои собственные оптимальные параметры. Причина в том, что все программы различны.

В файле **make.conf** следует указывать параметры оптимизации, которые, на ваш взгляд, повысят скорость системы в целом. Не помещайте сюда экспериментальные значения; излишняя оптимизация может привести к плохому поведению программ (аварийным завершениям, или, что хуже, неправильной работе).

Мы не будем описывать все возможные параметры оптимизации. Если вы хотите узнать о них всё, почитайте онлайн-руководства GNU (англ.) или страницу описания **gcc** (info gcc — доступна только на работающей системе Linux). Не забудьте прочитать сам файл `make.conf.example`: в нем также есть немало примеров и полезных сведений.

Первый параметр — флаг **-march=**, где указывается название целевой архитектуры. Возможные варианты описаны в комментариях в файле `make.conf.example`. Например, для архитектуры x86 Athlon XP:

**Листинг 17:** Значение `-march` для GCC

```
# пользователям AMD64, желающим получить действительно 64-битную
# систему, следует использовать -march=k8
# пользователи EM64T должны указать -march=nocona
-march=athlon-xp
```

**Примечание(JohnBat26):** Если Вы используете компилятор версии 4.2.0 и выше, то вместо указания специфичного типа процессора, необходимо указывать одно из двух (в параметрах: **-march** и **-mtune**):

1. **generic**: если Вы хотите, чтобы Ваш скомпилированный код запускался на всех процессорах, архитектуры x86;
2. **native**: если Вы хотите оптимизировать код только для Вашего процессора. В этом случае компилятор будет брать сведения о процессоре путем вызова `cpuid`! Т.е. **никаких nocona** (для Core 2 Duo) или **prescott** (для последних Intel Pentium 4 и т.п.).

Следующий — флаг **-O** (заглавная латинская «O», а не ноль), определяющий класс оптимизации в gcc. Допустимые значения:

- **s** — оптимизация по размеру;
- **0** — ноль, без оптимизации;
- **1, 2** или **3** — все большая оптимизация по скорости (в каждый класс входят все флаги предыдущего, и некоторые дополнительные).

Например, для оптимизации класса 2:

**Листинг 18:** Значение `-O` для GCC

```
-O2
```

Другой популярный флаг оптимизации — **-pipe** (использование для связи между различными проходами компилятора каналов вместо временных файлов).

Заметьте, что использование **-fomit-frame-pointer** (не хранить в регистре указатель стекового кадра для функций, которым он не нужен) может всерьез помешать отладке программ!

При определении переменных **CFLAGS** и **CXXFLAGS**, нужно объединять несколько флагов оптимизации, как в следующем примере:

**Листинг 19:** Определение переменных **CFLAGS** и **CXXFLAGS** (gcc < 4.2.0)

```
CFLAGS="-march=athlon-xp -pipe -O2" # для пользователей AMD64: -march=k8
# для пользователей EM64T: -march=nocona
CXXFLAGS="${CFLAGS}" # указывайте одинаковые значения обеих переменных
```

## MAKEOPTS

С помощью **MAKEOPTS** определяется, сколько параллельных процессов компиляции можно запускать при установке пакета. Хороший (но не всегда идеальный) вариант — значение, равное количеству процессоров в системе плюс один.

**Листинг 20:** MAKEOPTS для обычной однопроцессорной системы  
**MAKEOPTS="-j2"**

На старт, внимание, марш!

Измените **/mnt/gentoo/etc/make.conf** в соответствии со своими пожеланиями, и сохраните изменения (пользователям nano нужно нажать CTRL+X). Теперь вы готовы к установке базовой системы Gentoo.

## 6. Установка базовой системы Gentoo

### 6.a. Изменение корневого каталога

#### Дополнительно: выбор зеркала

Для ускорения загрузки исходного кода рекомендуется выбрать быстрое зеркало. Portage ищет переменную **GENTOO\_MIRRORS** в файле **make.conf** и использует зеркала, перечисленные в ней. Конечно, можно просмотреть наш список зеркал и выбрать одно или несколько, географически ближайших к вам (чаще всего они и будут самыми быстрыми), но мы предлагаем вам удобную утилиту **mirrorselect**, которая позволяет выбрать желаемые зеркала более удобным способом.

**Листинг 1:** Запуск **mirrorselect** для установки переменной GENTOO\_MIRRORS  
**# mirrorselect -i -o >>/mnt/gentoo/etc/make.conf**

**Предупреждение:** Не выбирайте зеркала IPv6. Файлы стадий пока не поддерживают протокол IPv6.

Вторая важная настройка — установка значения переменной **SYNC** в файле **make.conf**. Эта переменная указывает на сервер **rsync** (сервер удаленной синхронизации), выбранный вами для обновления дерева Portage (коллекции файлов ebuild — сборочных сценариев, содержащих все данные, нужные Portage для скачивания и установки программ). Хотя вписать адрес сервера в SYNC можно и вручную, **mirrorselect** позволяет упростить это действие:

**Листинг 2:** Выбор зеркала rsync с помощью **mirrorselect**  
**# mirrorselect -i -r -o >> /mnt/gentoo/etc/make.conf**

После выполнения **mirrorselect**, мы рекомендуем проверить все значения в файле **/mnt/gentoo/etc/make.conf**!

Копирование сведений о DNS



Перед тем, как перейти в новую среду, осталось сделать одно дело: скопировать информацию о DNS (системе доменных имен) в файл **/etc/resolv.conf**. Это нужно, чтобы при переходе сохранить работоспособность сети. В файле **/etc/resolv.conf** содержатся адреса серверов имен, используемых в вашей сети.

**Листинг 3:** Копирование информации о DNS  
(параметр «-L» нужен, чтобы случайно не скопировать  
символьную ссылку)  
**# cp -L /etc/resolv.conf /mnt/gentoo/etc/resolv.conf**

## Монтирование файловых систем /proc и /dev

Смонтируйте файловую систему **/proc** в **/mnt/gentoo/proc**, чтобы после изменения корневого каталога устанавливаемая система смогла обращаться к информации, предоставляемой ядром, а затем создайте привязку монтирования файловой системы **/dev**.

**Листинг 4:** Монтирование **/proc** и **/dev**  
**# mount -t proc none /mnt/gentoo/proc**  
**# mount -o bind /dev /mnt/gentoo/dev**

## Переход в новую среду

Итак, все разделы подготовлены, а базовая операционная среда установлена. Теперь пора войти в нее, изменив корневой каталог. Таким образом, мы перейдем из текущей установочной среды (с компакт-диска или другого установочного носителя) в свою устанавливаемую систему (находящуюся в недавно размеченных разделах).

Изменение выполняется в три этапа:

1. Сначала мы с помощью **chroot** изменим корневой каталог с **/** (находящийся на установочном носителе) на **/mnt/gentoo** (находящийся на ваших дисковых разделах).
2. Затем мы создадим новую среду, пользуясь утилитой **env-update**, которая, собственно, создает переменные среды.
3. Наконец, мы загрузим эти переменные в память при помощи **source**.

**Листинг 5:** Изменение корневого каталога для перехода в новую среду  
**# chroot /mnt/gentoo /bin/bash**  
**# env-update**  
**>> Regenerating /etc/ld.so.cache...**  
**# source /etc/profile**  
**# export PS1="(chroot) \$PS1"**

**Поздравляем!** Теперь вы в своей собственной среде Gentoo Linux. Конечно, на этом она далеко еще не готова, поэтому в руководстве еще осталось несколько разделов :-)

## 6.b. Настройка Portage

### Обновление дерева Portage

Теперь надо обновить дерево Portage до самой последней версии с помощью команды **emerge --sync**:

**Листинг 6:** Обновление дерева портежей

```
# emerge --sync
```

(если вы пользуетесь медленным терминалом, например, последовательным терминалом или кадровым буфером, для ускорения процесса можно добавить параметр `--quiet` :)

```
# emerge --sync --quiet
```

Если компьютер подключен к интернету через межсетевой экран, блокирующий прохождение rsync-пакетов, вы можете воспользоваться командой `emerge-webrsync`, которая скачивает и устанавливает снимок дерева.

Если выдано предупреждение, что имеется новая версия Portage и ее нужно обновить, выполните обновление командой **emerge portage**.

### Выбор нужного профиля

Сначала дадим небольшое определение.

**Профиль** — конструктивный элемент любой системы Gentoo. В нем указываются не только значения по умолчанию для **CHOST**, **CFLAGS** и других важных переменных, он также привязывает систему к определенному диапазону допустимых версий пакетов. Диапазоны поддерживаются разработчиками Gentoo.

Раньше пользователь редко касался профиля. Однако, пользователи **x86**, **hppa** и **alpha** могут выбирать из двух вариантов профиля: одного для ядра 2.4, другого для ядра 2.6. Это вызвано необходимостью улучшения интеграции ядер 2.6. Для архитектур ppc и ppc64 также существует несколько профилей. Мы поговорим о них позже.

Узнать, какой профиль используется в системе в данный момент, вы можете командой:

**Листинг 7:** Выяснение используемого профиля

```
# ls -FGg /etc/make.profile
```

```
lrwxrwxrwx 1 48 Apr 8 18:51 /etc/make.profile -> ../usr/portage/profiles/default-linux/x86/2007.0/
```

Если вы используете одну из трех упомянутых архитектур, профиль по умолчанию даст вам систему с ядром 2.6. Это рекомендуется по умолчанию, но за вами сохраняется и право выбора другого профиля.

Для некоторых архитектур также предусмотрены подпрофили `desktop` и `server`. Загляните в профиль `2007.0/`, чтобы узнать, есть ли подпрофили для вашей архитектуры. Возможно, вы захотите заглянуть в `make.defaults` профиля `desktop`, чтобы определить, подходит ли он вам.

Некоторым пользователям, возможно, захочется установить систему, основанную на старом профиле, с ядром 2.4. Если для этого есть веские основания, сначала надо проверить, существует ли такой дополнительный профиль. На системах x86 это

можно сделать следующей командой:

**Листинг 8:** Выяснение наличия дополнительного профиля

```
# ls -d /usr/portage/profiles/default-linux/x86/no-nptl/2.4
/usr/portage/profiles/default-linux/x86/no-nptl/2.4
```

В приведенном примере дополнительный профиль 2.4 существует (т.е. нет сообщений об отсутствующем файле или каталоге). Рекомендуется использовать профиль по умолчанию, но если вы все же хотите сменить его, это можно сделать так:

**Листинг 9:** Переключение профиля на 2.4

(убедитесь, что используете нужную архитектуру; этот пример - для x86)

```
# ln -snf /usr/portage/profiles/default-linux/x86/no-nptl/2.4 /etc/make.profile
```

(вывод списка файлов в профиле 2.4)

```
# ls -FGg /etc/make.profile/
```

total 12

```
-rw-r--r-- 1 939 Dec 10 14:06 packages
```

```
-rw-r--r-- 1 347 Dec 3 2004 parent
```

```
-rw-r--r-- 1 573 Dec 3 2004 virtuals
```

Для архитектуры **ppc** в выпуске 2006.1 появилось несколько новых профилей:

**Листинг 10:** Профили для PPC

(базовый профиль PPC, подходит для всех PPC-машин, минимальный)

```
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc32/2006.1 /etc/make.profile
```

(профиль для G3)

```
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc32/2006.1/G3 /etc/make.profile
```

(профиль для G3 Pegasos)

```
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc32/2006.1/G3/Pegasos/ /etc/make.profile
```

(профиль для G4 (Altivec))

```
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc32/2006.1/G4 /etc/make.profile
```

(профиль для G4 (Altivec) Pegasos)

```
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc32/2006.1/G4/Pegasos/ /etc/make.profile
```

Для архитектуры **ppc64** в выпуске 2006.1 также появилось несколько новых профилей:

**Листинг 11:** Профили для PPC64

(базовый профиль PPC64 с 64-битным режимом пользователя, для всех PPC64-машин)

```
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc64/2006.1/64bit-userland /etc/make.profile
```

(базовый профиль PPC64 с 32-битным режимом пользователя, для всех PPC64-машин)

```
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc64/2006.1/32bit-userland /etc/make.profile
```

(для каждого режима пользователя существуют подпрофили, как показано ниже.)

(«userland» необходимо заменять на нужный вариант режима из примеров выше)

(профиль 970 для JS20)

```
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc64/2006.1/(userland)/970 /etc/make.profile
```

(профиль для G5)

```
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc64/2006.1/(userland)/970/pmac /etc/make.profile
```

(профиль для POWER3)

```
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc64/2006.1/(userland)/power3 /etc/make.profile
```

(профиль для POWER4)

```
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc64/2006.1/(userland)/power4 /etc/make.profile
```

(профиль для POWER5)

```
# ln -snf /usr/portage/profiles/default-linux/ppc/ppc64/2006.1/(userland)/power5 /etc/make.profile
```

(многорежимные профили на дату выхода этого выпуска нестабильны)

## Настройка переменной USE

**USE** («использовать») — одна из самых мощных переменных, имеющих в распоряжении пользователей Gentoo. Она позволяет при компиляции программ включать или отключать поддержку определенных необязательных функций. Например, некоторые программы можно компилировать с поддержкой **gtk** или **qt** на выбор. Другие можно собирать, включая или отключая поддержку **SSL**. Некоторые программы можно даже компилировать с поддержкой кадрового буфера (**svgalib**) вместо поддержки X11 (X-сервера).

В большинстве дистрибутивов пакеты собраны с поддержкой практически всех мыслимых вариантов. Это увеличивает размер программ и время запуска, не говоря уже о чрезмерных зависимостях. В Gentoo вы сами можете определять, с какими возможностями следует компилировать пакет. Здесь играет роль переменная **USE**.

В переменной **USE** указываются ключевые слова, которые отражаются на параметрах компиляции. Например, параметр **ssl** включает компиляцию с поддержкой **ssl** всех программ, которые способны его поддерживать. **-X** отключает поддержку X-сервера (обратите внимание на предшествующий знак «минус»). Параметры **gnome** **gtk** **-kde** **-qt** обеспечивают компиляцию ваших программ с поддержкой Gnome и **gtk**, но без поддержки KDE и **qt**, делая систему оптимальной для GNOME.

Настройки **USE** по умолчанию хранятся в файлах **make.defaults** вашего профиля. Файлы **make.defaults** находятся в каталоге, на который указывает **/etc/make.profile**, а также во всех родительских каталогах. Значение **USE** по умолчанию — это сумма всех значений **USE** во всех файлах **make.defaults**. Все, что вы вносите в **/etc/make.conf**, рассчитывается относительно этих значений. Когда вы добавляете что-либо к значению **USE**, оно добавляется в список по умолчанию. Когда удаляете что-либо (указывая ключевое слово с предшествующим знаком минус), оно удаляется из списка по умолчанию (если оно там вообще было). Никогда ничего не меняйте в **/etc/make.profile**. Все, что там находится, перезаписывается при обновлении Portage!

Полное описание переменной **USE** находится во второй части настольной книги Gentoo в главе **USE-флаги**. Полное описание возможных значений признаков использования находится в вашей системе в файле **/usr/portage/profiles/use.desc**.

**Листинг 12:** Просмотр допустимых **USE**-флагов

```
# less /usr/portage/profiles/use.desc
```

(для прокрутки пользуйтесь клавишами стрелок, для выхода нажимайте "q")

Например, приведем значение **USE** для системы, базирующейся на KDE, с включением поддержки DVD, ALSA и записи CD:

**Листинг 13:** Открытие файла **/etc/make.conf**

```
# nano -w /etc/make.conf
```

**Листинг 14:** Значение **USE**

```
USE="-gtk -gnome qt kde dvd alsa cdr"
```

**Дополнительно: региональные параметры GLIBC**

Скорее всего, вы будете использовать в системе один-два набора региональных параметров. Нужные регионы можно указать в `/etc/locale.gen`.

**Листинг 15:** Открытие `/etc/locale.gen`  
`# nano -w /etc/locale.gen`

Вот пример одновременного подключения как английского (США), так и немецкого (Германия) с соответствующими кодировками (например, UTF-8).

**Листинг 16:** Укажите свои региональные настройки  
`en_US/ISO-8859-1`  
`en_US.UTF-8/UTF-8`  
`de_DE/ISO-8859-1`  
`de_DE@euro/ISO-8859-15`

Следующий шаг — запустить **locale-gen**. Это сгенерирует настройки для всех регионов, указанных вами в файле `/etc/locale.gen`.

Примечание: `locale-gen` имеется в `glibc-2.3.6-r4` и более новой. Если у вас старшая версия `glibc`, сейчас ее следует обновить.

**Примечание:** Настройки, подобной приведенной, для поддержки русского языка недостаточно. За дополнительными сведениями обращайтесь к отдельным руководствам по русификации Gentoo. — прим. пер.

Теперь приступим к настройке ядра.

## 7. Настройка ядра

### 7.a. Часовой пояс

Сначала необходимо выбрать часовой пояс (time zone), чтобы система знала, где вы находитесь. Найдите подходящий пояс в `/usr/share/zoneinfo`, затем скопируйте его в `/etc/localtime`. Постарайтесь, пожалуйста, избегать использования часовых поясов `/usr/share/zoneinfo/Etc/GMT*`, т. к. их названия не соответствуют ожидаемым поясам. Например, `GMT-8` фактически является поясом `GMT+8`.

**Листинг 1:** Установка часового пояса  
`# ls /usr/share/zoneinfo`  
(допустим, вы хотите использовать GMT)  
`# cp /usr/share/zoneinfo/GMT /etc/localtime`

**Примечание (JohnBat26):** Если возникают проблемы со временем (особенно с часовыми поясами), то выполните следующее (получено с форума):

1. Выстави правильное время.
2. Удали файл `/etc/adjtime`. Все, часы идут при перезагрузке нормально.
3. `nano -w /etc/conf.d/clock`. Ставь там вместо `UTC Local`, и все!!!

## 7.b. Установка исходных кодов ядра

### Выбор ядра

**Ядро Linux** — это то, вокруг чего формируются все дистрибутивы. Это слой между пользовательскими программами и вашим оборудованием. Пользователи Gentoo могут выбирать из нескольких типов ядра. Их полный список и описание находится в руководстве по выбору ядра Gentoo (англ.).

Для x86 систем, помимо прочего, есть такие ядра, как

- **vanilla-sources** (ядро по умолчанию от разработчиков linux ядра);
- **gentoo-sources** (ядро с заплатками, улучшающими производительность).

С помощью команды **emerge** выберите и установите ядро. `USE="-doc"` требуется, чтобы на этом этапе избежать установки `horg-x11` или других связанных пакетов. `USE="symlink"` при новой установке не требуется, но обеспечивает правильное создание символической ссылки `/usr/src/linux`:

**Листинг 2:** Установка исходных кодов ядра  
`# USE="-doc symlink" emerge gentoo-sources`

Просмотрев содержимое каталога `/usr/src`, вы увидите символическую ссылку `linux`, которая указывает на каталог с исходными кодами ядра. В данном случае, установленный исходный код ядра указывает на `gentoo-sources-2.6.12-r10`. Имейте в виду, что ваша версия может отличаться.

**Листинг 3:** Просмотр папки с исходными кодами ядра  
`# ls -l /usr/src/linux`  
`lrwxrwxrwx 1 root root 12 Oct 13 11:04 /usr/src/linux -> linux-2.6.12-gentoo-r10`

Теперь настало время настройки и компиляции ядра. Можно использовать сценарий **genkernel**, который сформирует стандартное ядро, аналогичное используемому на установочном компакт-диске. Однако сначала мы расскажем о «ручной» настройке ядра, так как это лучший из способов оптимизации системы.

Если вы собираетесь настроить ядро вручную, то переходите к разделу по умолчанию: ручная настройка. Если вы хотите использовать **genkernel**, то читайте альтернатива: использование **genkernel**.

## 7.c. По умолчанию: ручная настройка

### Введение

Ручная настройка ядра считается одним из самых сложных вопросов для пользователей Linux. Зато после настройки нескольких вариантов ядра вы и не вспомните, что это когда-то было трудно ;)

Однако есть одна непреложная истина: запуская ручную настройку ядра, следует знать состав аппаратуры своего компьютера. Большую часть информации можно собрать, установив пакет **pciutils** (**emerge pciutils**), в который входит **lspci**. **lspci**

можно использовать, находясь в среде с временным корневым каталогом. Можно не обращать внимания на предупреждения, касающиеся pcilib (например: pcilib: cannot open /sys/bus/pci/devices [pcilib: не могу открыть /sys/bus/pci/devices]), которые выводит на экран lspci. Можно запустить lspci и вне временной среды установки. Результат тот же. Вы также можете запустить lsmod для просмотра модулей ядра, используемых установочным компакт-диском (это даст хорошую подсказку, какие параметры включать).

Теперь войдите в каталог с исходными кодами ядра и введите команду **make menuconfig**. Этой командой вы вызовете меню настройки, использующее ncurses.

**Листинг 4:** Вызов меню настройки

```
# cd /usr/src/linux
# make menuconfig
```

Перед вами появятся несколько разделов настройки. Сначала перечислим пункты, включение которых обязательно (иначе Gentoo не заработает, или для его работы потребуются дополнительные ухищрения).

### Включение нужных пунктов

Прежде всего, разрешите использование разрабатываемого и экспериментального кода и драйверов. Без этого не появятся некоторые важные пункты:

**Листинг 5:** Выбор экспериментальных кодов/драйверов, основные параметры

```
Code maturity level options --->
[*] Prompt for development and/or incomplete code/drivers"
General setup --->
[*] Support for hot-pluggable devices
```

Удостоверьтесь, что каждый драйвер, жизненно необходимый для загрузки системы (например, SCSI-контроллера), собран внутри ядра, а не как модуль, иначе система совсем не сможет загрузиться.

Далее выберите нужное семейство процессоров:

**Листинг 6:** Выбор подходящего семейства процессоров

```
Processor type and features --->
(измените в соответствии со своей системой)
(Athlon/Duron/K7) Processor family
```

Теперь войдите в File Systems и выберите поддержку всех нужных файловых систем. Не компилируйте их как модули, иначе система Gentoo не сможет смонтировать дисковые разделы. Также выберите Virtual memory и /proc file system. При использовании ядра 2.4, вам потребуется включить /dev file system, т.к. это ядро не поддерживает udev.

**Листинг 7:** Выбор необходимых файловых систем

```
(при использовании ядра 2.4.x)
File systems --->
[*] Virtual memory file system support (former shm fs)
[*] /proc file system support
[*] /dev file system support (EXPERIMENTAL)
[*] automatically mount /dev at boot
```

[ ] /dev/pts file system for Unix98 PTYs

(при использовании ядра 2.6.x)

File systems --->

Pseudo Filesystems --->

[\*] /proc file system support

[\*] Virtual memory file system support (former shm fs)

(включите поддержку всех нужных вам файловых систем)

<\*> Reiserfs support

<\*> Ext3 journalling file system support

<\*> JFS filesystem support

<\*> Second extended fs support

<\*> XFS filesystem support

Если ваш **BIOS** не поддерживает жесткие диски большого размера, и вы ограничили перемычками сообщаемый диском объем, то для получения доступа ко всему диску нужно включить следующие параметры ядра:

**Листинг 8:** Включение поддержки автокоррекции геометрии

(только в ядре 2.4.x)

ATA/IDE/MFM/RLL support --->

IDE, ATA and ATAPI Block devices --->

<\*> Include IDE/ATA-2 DISK support

[ ] Use multi-mode by default

[\*] Auto-Geometry Resizing support

Не забудьте включить поддержку режима **DMA** для ваших дисков:

**Листинг 9:** Включение DMA

Device Drivers --->

ATA/ATAPI/MFM/RLL support --->

[\*] Generic PCI bus-master DMA support

[\*] Use PCI DMA by default when available

Если вы используете **PPPoE** или коммутируемого соединения для соединения с интернетом, потребуется включить в ядре следующие функции:

**Листинг 10:** Выбор нужных драйверов PPPoE

(при использовании ядра 2.4.x)

Network device support --->

<\*> PPP (point-to-point protocol) support

<\*> PPP support for async serial ports

<\*> PPP support for sync tty ports

(при использовании ядра 2.6.x)

Device Drivers --->

Networking support --->

<\*> PPP (point-to-point protocol) support

<\*> PPP support for async serial ports

<\*> PPP support for sync tty ports

Два параметра сжатия данных не являются обязательными, но и не повредят. То же относится и к параметру **PPP over Ethernet** который нужен только для гр-рррое, для поддержки **PPPoE** в режиме ядра.



При необходимости, не забудьте включить в ядре поддержку своей сетевой платы.

Если у вас процессор Intel с поддержкой HyperTreading или многопроцессорная система, следует включить «симметричную многопроцессорную обработку»:

**Листинг 11:** Включение поддержки SMP

```
Processor type and features --->
[*] Symmetric multi-processing support
```

Если вы пользуетесь устройствами ввода на шине USB, (например, мышью или клавиатурой), не забудьте включить их поддержку:

**Листинг 12:** Включение поддержки USB для устройств ввода

```
USB Support --->
<*> USB Human Interface Device (full HID) support
```

Владельцам ноутбуков, которым нужна поддержка **PCMCIA**, не нужно включать драйверы **PCMCIA** в ядре, выбрав ядро версии 2.4. В пакете pcmcia-cs, который будет устанавливаться позже, есть более свежие драйверы. Пользователям ядра 2.6 следует использовать драйверы **PCMCIA**, включенные в ядро.

Наряду с включением поддержки **PCMCIA** в ядре 2.6, не забудьте включить поддержку моста PCMCIA своей системы:

**Листинг 13:** Включение поддержки PCMCIA для ядра 2.6

```
Bus options (PCI, PCMCIA, EISA, MCA, ISA) --->
PCCARD (PCMCIA/CardBus) support --->
<*> PCCard (PCMCIA/CardBus) support
(если нужно (обычно нужно), включите поддержку старых 16-битных карт PCMCIA)
<*> 16-bit PCMCIA support
[*] 32-bit CardBus support
(выберите наиболее подходящий мост)
--- PC-card bridges
<*> CardBus yenta-compatible bridge support (NEW)
<*> Cirrus PD6729 compatible bridge support (NEW)
<*> i82092 compatible bridge support (NEW)
<*> i82365 compatible bridge support (NEW)
<*> Databook TCIC host bridge support (NEW)
```

Закончив настройку ядра, приступайте к компиляции и установке.

## Компиляция и установка

Теперь, когда ядро настроено, настало время его скомпилировать и установить. Выйдите из настройки и запустите процесс компиляции:

**Листинг 14:** Компиляция ядра

```
(для ядра 2.4)
# make dep && make bzImage modules modules_install
```

```
(для ядра 2.6)
# make && make modules_install
```

По завершении компиляции, скопируйте образ ядра в каталог /boot. Используйте любое имя, которое вам покажется подходящим для избранного ядра, и запомните его, так как позже при настройке начального загрузчика оно потребуется. Не забудьте заменить <версия-ядра> на имя и версию своего ядра.

**Листинг 15:** Установка ядра

```
# cp arch/i386/boot/bzImage /boot/<версия-ядра>
```

Теперь переходите к модулям ядра.

## 7.d. Альтернатива: использование genkernel

Если вы читаете эти строки, значит, вы собираетесь использовать наш сценарий **genkernel** для настройки ядра за вас.

Теперь когда исходные коды ядра установлены, настало время скомпилировать ядро, собрав его автоматически при помощи сценария genkernel. genkernel выполняет настройку ядра практически так же, как настраивается ядро на установочном компакт-диске. То есть система, ядро которой собрано genkernel, как правило, будет выполнять определение всех устройств при загрузке. Поскольку genkernel не требует никакой ручной настройки ядра, это идеальное решение для тех, кому неуютно от одной мысли, что придется компилировать ядро.

Рассмотрим, как использовать **genkernel**. Сначала установите его:

**Листинг 16:** Установка genkernel

```
# emerge genkernel
```

Теперь, если вы собираетесь использовать ядро ветви 2.6, скопируйте настройку ядра, используемую установочным диском, в каталог, где genkernel ищет настройку по умолчанию:

**Листинг 17:** Копирование файла настройки ядра с установочного диска (только если вы собираетесь настраивать ядро 2.6.x)

```
# zcat /proc/config.gz > /usr/share/genkernel/x86/kernel-config-2.6
```

Теперь скомпилируйте исходный код ядра, запустив genkernel all. Имейте в виду, что компиляция займет приличное время, поскольку genkernel собирает ядро, поддерживающее практически любое оборудование.

Если в загрузочном разделе не используется файловая система ext2 или ext3, то вам придется вручную настроить ядро, запустив genkernel --menuconfig all, добавив поддержку используемой файловой системы в ядро (т.е. не модулем). Пользователям EVMS2 или LVM2, вероятно, захочется также добавить параметр командной строки --evms2 или --lvm2.

**Листинг 18:** Запуск genkernel

```
# genkernel all
```

В итоге работы **genkernel** будет создано ядро, полный набор модулей и начальный корневой диск (**initrd**). Ядро и **initrd** будут использоваться в процессе настройки загрузчика системы, которая описана далее в руководстве. Запишите имена ядра и начального корневого диска, так как они понадобятся при составлении конфигурационного файла загрузчика. **initrd** запускается сразу при начальной загрузке, выполняя автоматическое определение устройств (точно так же, как при запуске установочного диска), до запуска «настоящей» системы.

**Листинг 19:** Уточнение названия созданного образа ядра и **initrd**

```
# ls /boot/kernel* /boot/initramfs*
```

Теперь давайте еще на шаг приблизим нашу систему к установочному диску: соберем **coldplug**. Если **initrd** определяет оборудование, необходимое для загрузки системы, **coldplug** выполняет автоопределение всех остальных устройств. Для установки и подключения **coldplug** введите команду:

**Листинг 20:** Установка и подключение **coldplug**

```
# emerge coldplug
# rc-update add coldplug boot
```

## 7.е. Модули ядра

### Настройка модулей

Модули, которые требуется загружать автоматически, нужно указать в `/etc/modules.autoload.d/kernel-2.4` (или `kernel-2.6`). Также, при желании, модулям можно сообщить дополнительные параметры.

Для просмотра всех доступных модулей запустите команду `find`, заменив «<версия ядра>» на версию только что собранного ядра:

**Листинг 21:** Просмотр перечня доступных модулей

```
# find /lib/modules/<kernel version>/ -type f -iname '*.o' -or -iname '*.ko'
```

Например, для автоматической загрузки модуля `3c59x.o` измените файл `kernel-2.4` или `kernel-2.6`, указав в нем имя модуля.

**Листинг 22:** Изменение `/etc/modules.autoload.d/kernel-2.4`

(показан пример для ядра 2.4.x)

```
# nano -w /etc/modules.autoload.d/kernel-2.4
```

**Листинг 23:** `/etc/modules.autoload.d/kernel-2.4` или `kernel-2.6`

```
3c59x
```

Теперь переходите к настройке параметров системы.

## 8. Настройка параметров системы

### 8.а. Параметры файловых систем

#### Что такое fstab?

В Linux все разделы, используемые системой, должны быть перечислены в **/etc/fstab**. В этом файле указываются точки подключения разделов (mountpoints, местоположение разделов в файловой системе), порядок подключения, а также дополнительные параметры (автоматический или ручной режим подключения, достаточность прав пользователя для подключения и т.п.)

#### Создание /etc/fstab

В **/etc/fstab** используется специальный формат. Каждая строка состоит из шести полей, разделяемых пробелами, знаками табуляции или их сочетанием. Каждое поле имеет свое назначение:

1. Первое поле обозначает раздел (partition) (путь к файлу устройства).
2. Второе поле указывает точку подключения (mountpoint), в которую монтируется раздел.
3. Третье поле задает тип файловой системы (filesystem), используемой в разделе.
4. В четвертом поле указываются параметры подключения (mountoptions), используемые mount при подключении раздела. Поскольку для каждой файловой системы существуют свои параметры, рекомендуется прочитать страницу справки по mount (man mount), где приведен их полный перечень. При указании нескольких параметров подключения их следует разделять запятыми.
5. Пятое поле используется dump для определения, требуется ли резервное копирование раздела средствами dump. Обычно это поле можно просто установить в 0 (ноль).
6. Шестое поле используется **fsck** для определения порядка проверки (check) файловых систем после некорректного завершения работы системы. Для корневой файловой системы рекомендуется значение 1, а для остальных — 2 (или 0, когда проверка файловой системы не требуется).

**Важно:** Вариант файла **/etc/fstab** по умолчанию, входящий в Gentoo, не является работоспособным. Вам потребуется создать свой собственный **/etc/fstab**.

**Листинг 1:** Открытие /etc/fstab  
**# nano -w /etc/fstab**

Укажите правила, соответствующие вашей схеме разбивки, и добавьте правила для **/proc**, для **tmpfs**, для своих дисководов CD-ROM (если есть другие разделы или устройства, их тоже можно указать).

Теперь на основе приведенного примера создайте собственный файл **/etc/fstab**:

Параметр `auto` позволяет `mount` определять тип файловой системы автоматически (рекомендуется для съемных носителей, которые могут оказаться размечены в одной из множества существующих файловых систем), а `user` позволяет монтировать компакт-диски обычным пользователям.

Чтобы повысить быстродействие, большинству пользователей стоит добавить параметр `noatime` в параметры подключения, что приведет к ускорению за счет отключения регистрации отметки времени доступа к файлам (обычно в ней все равно нет необходимости):

Перепроверьте свой файл `/etc/fstab`, сохраните его, и выйдите из редактора, чтобы продолжить настройку.

## Примечание (JohnBat26): пример моего `/etc/fstab`:

### Листинг 2: Пример `/etc/fstab`

```
/dev/sda2      /boot      ext2      defaults,noatime      1 2
/dev/sda5      /           xfs       defaults,noatime      0 1
/dev/sda3      none       swap      sw                    0 0
/dev/sda7      /home      xfs       defaults,noatime      0 1
/dev/sr0       /mnt/cdrom auto      noauto,ro,user,icharset=utf8 0 0
# раздел с FAT32
/dev/sda1      /windows/c vfat      auto,icharset=utf8,quiet,user,umask=000,noatime 0 0
# раздел NTFS только для чтения
/dev/sda6      /windows/d ntfs      auto,nls=utf8,umask=0,user,noatime      0 0
/dev/sda8      /windows/e vfat      auto,icharset=utf8,user,quiet,umask=000,noatime 0 0
/dev/sda9      /home/media xfs       defaults,noatime      0 1
# раздел NTFS для чтения и записи при помощи ntfs-3g
/dev/sda10     /mnt/windows/f ntfs-3g   -o silent,umask=0,locale=ru_RU.utf8
# NOTE: The next line is critical for boot!
proc           /proc      proc      defaults              0 0
# flash drive
/dev/sdb       /mnt/flash auto      user,noauto,rw,async,icharset=utf8      0 0
shm            /dev/shm   tmpfs     nodev,nosuid,noexec   0 0
```

## 8.b. Параметры сети

### Hostname, Domainname и т. д.

Еще один вопрос, который нужно решить пользователю — как назвать свой компьютер. Он кажется довольно простым, но многие затрудняются дать подходящее имя для своей Linux-системы. Чтобы вам стало легче, запомните, что какое бы имя вы не выбрали, потом его всегда можно изменить. Например, вы могли бы просто назвать свою систему `tux`, а домен — `homenetwork`.

Листинг 2: Установка имени узла  
**# nano -w /etc/conf.d/hostname**

(присвойте переменной `HOSTNAME` имя своего узла)  
**HOSTNAME="tux"**

Во-вторых, установим имя домена (`domainname`) в `/etc/conf.d/net`:

### Листинг 3: Установка имени домена

```
# nano -w /etc/conf.d/net  
(присвойте переменной DNSDOMAIN имя своего домена)  
dns_domain_lo="homenetwork"
```

Если у вас есть домен NIS (а если вы не знаете, что это такое, то у вас его точно нет), его также необходимо указать:

### Листинг 4: Установка имени NIS-домена

```
# nano -w /etc/conf.d/net  
  
(укажите название своего домена NIS переменной nis_domain)  
nis_domain_lo="my-nisdomain"
```

## Настройка сети

Прежде, чем возмутиться: «Эй, мы же все это уже делали!» — вспомните, что подключение к сети, настроенное вначале, было предназначено лишь для установки Gentoo. Теперь же вы настраиваете сеть для постоянного использования.

Примечание: Более подробные сведения о сетях, включая дополнительные темы, такие как объединение, образование мостов, настройка виртуальных сетей (VLAN) 802.1Q или беспроводных сетей, представлены в разделе настройка сети в Gentoo.

Все настройки сети собраны в файле **/etc/conf.d/net**. В нем используется простой формат, хотя, если вы не знакомы с ручной настройкой сети, он не слишком очевиден. Но не бойтесь, мы все объясним. В файле **/etc/conf.d/net.example** приведен подробно прокомментированный пример, охватывающий много различных конфигураций.

По умолчанию используется **DHCP**. Чтобы **DHCP** заработал, требуется установить **DHCP**-клиент, как описано далее в разделе Установка нужных системных средств. Не забудьте установить DHCP-клиент.

Если настройка сетевого подключения нужна вам для указания специфических параметров DHCP, или из-за того, что вы вообще не используете DHCP, откройте **/etc/conf.d/net** в своем любимом редакторе (в этом примере использован nano):

### Листинг 5: Открытие /etc/conf.d/net для изменения

```
# nano -w /etc/conf.d/net
```

Вы увидите следующее:

### Листинг 6: /etc/conf.d/net по умолчанию

```
# This blank configuration will automatically use DHCP for any net.*  
# scripts in /etc/init.d. To create a more complete configuration,  
# please review /etc/conf.d/net.example and save your configuration  
# in /etc/conf.d/net (this file :!)).
```

```
(# Этот пустой файл настройки приводит к автоматическому использованию  
# DHCP всеми сценариями net.* из /etc/init.d. Для создания более полной  
# настройки, пожалуйста, просмотрите /etc/conf.d/net.example, а свою  
# настройку сохраните в /etc/conf.d/net (в этом файле :!)). )
```

Чтобы указать свой собственный адрес IP, маску сети и шлюз, потребуется настроить как **config\_eth0**, так и **routes\_eth0**:

**Листинг 7:** Ручная настройка параметров IP для eth0  
**config\_eth0=( "192.168.0.2 netmask 255.255.255.0 brd 192.168.0.255" )**  
**routes\_eth0=( "default gw 192.168.0.1" )**

Чтобы при использовании DHCP указать специфические параметры, определите **config\_eth0** и **dhcp\_eth0**:

**Листинг 8:** Автоматическое получение адреса IP для eth0  
**config\_eth0=( "dhcp" )**  
**dhcp\_eth0="nodns nontp nonis"**

Список допустимых параметров дан в файле `/etc/conf.d/net.example`.

Если у вас несколько сетевых интерфейсов, повторите эти шаги для **config\_eth1**, **config\_eth2** и т.д.

Теперь сохраните параметры и выйдите из редактора, чтобы продолжить настройку.

### **Автоматический запуск сетевого подключения при загрузке**

Для запуска сетевых интерфейсов при загрузке необходимо добавить их в уровень запуска по умолчанию. Если у вас интерфейсы типа PCMCIA, пропустите этот шаг, поскольку интерфейсы PCMCIA запускаются сценарием инициализации PCMCIA.

**Листинг 9:** Добавление net.eth0 в уровень запуска default  
**# rc-update add net.eth0 default**

Если у вас несколько сетевых интерфейсов, потребуется создать для них соответствующие сценарии инициализации net.eth1, net.eth2 и т.д. Для этого можно использовать ln:

**Листинг 10:** Создание дополнительных сценариев инициализации  
**# cd /etc/init.d**  
**# ln -s net.lo net.eth1**  
**# rc-update add net.eth1 default**

### **Указание сетевых узлов**

Теперь расскажем системе Linux о вашей сети. Эти сведения указываются в **/etc/hosts**, и помогают разрешению имен в IP-адреса для узлов, не обрабатываемых сервером имен. Требуется определить вашу систему. Также можно определить другие системы в сети, если вы не собираетесь устанавливать собственную систему DNS.

**Листинг 11:** Открытие /etc/hosts  
**# nano -w /etc/hosts**

**Листинг 12:** Указание сведений об узлах сети  
(определение текущей системы)

```
127.0.0.1 tux.homenetwork tux localhost
```

(определите другие машины в своей сети,  
для этого у них должен быть статический IP-адрес.)

```
192.168.0.5 jenny.homenetwork jenny
```

```
192.168.0.6 benny.homenetwork benny
```

Чтобы продолжить настройку, сохраните файл и выйдите из редактора.

## 8.с. Параметры системы

### Пароль root

Прежде всего, нужно установить пароль root (администратора), набрав:

**Листинг 13:** Установка пароля root  
`# passwd`

Если вы хотите, чтобы root мог входить в систему через последовательный терминал, добавьте `tts/0` в `/etc/securetty`:

**Листинг 14:** Добавление `tts/0` to `/etc/securetty`  
`# echo "tts/0" >> /etc/securetty`

### Параметры системы

Для общей настройки системы в Gentoo используется `/etc/rc.conf`. Откройте `/etc/rc.conf` и с удовольствием прочитайте все комментарии, находящиеся в этом файле :)

**Листинг 15:** Открытие `/etc/rc.conf`  
`# nano -w /etc/rc.conf`

Завершив изменение `/etc/rc.conf` сохраните файл и выйдите из редактора.

Как видите, этот файл подробно прокомментирован, что поможет вам в настройке необходимых конфигурационных переменных. Можно настроить систему на использование **unicode**, а также указать редактор по умолчанию и диспетчер отображения (например, **gdm** или **kdm**).

Для управления раскладками клавиатуры в Gentoo используется `/etc/conf.d/keymaps`. Для настройки своей клавиатуры измените его.

**Листинг 16:** Открытие `/etc/conf.d/keymaps`  
`# nano -w /etc/conf.d/keymaps`



Будьте особенно тщательны при установке переменной раскладки клавиатуры (KEYMAP): выбрав неверную раскладку, вы можете получить непредсказуемый результат при попытке ввода с клавиатуры.

По завершении изменения **/etc/conf.d/keymaps** сохраните файл и выйдите из редактора.

Для настройки параметров часов в Gentoo используется **/etc/conf.d/clock**. Изменяйте его в соответствии со своими потребностями.

Если аппаратные часы вашей системы настроены не на часовой пояс UTC (Гринвич), в файл необходимо добавить строку **CLOCK="local"**. В противном случае вы заметите сдвиг часового пояса.

После завершения настройки **/etc/conf.d/clock** сохраните файл и выйдите из редактора.

Переходите к установке нужных системных средств.

## **9. Установка нужных системных средств**

### **9.a. Диспетчер устройств**

Если вы используете ядро 2.4, и при этом устанавливаете Gentoo из файла третьей стадии (stage3), вам необходимо еще кое-что сделать. Так как теперь в Gentoo по умолчанию используется udev, а ядром 2.4 udev не поддерживается, вам потребуется установить devfsd, и убрать udev.

**Листинг 1:** Установка devfsd

(для тех, кто использует ядро 2.4.x при установке с третьей стадии)

```
# emerge --unmerge udev
```

```
# emerge devfsd
```

### **9.b. Системные службы журналирования**

Некоторые средства не включены в архив stage3, поскольку одинаковые возможности можно обеспечить различными пакетами. Теперь вы сами выберете, какие именно установить.

Первый инструмент, который вам необходимо выбрать, должен дать системе возможность журналирования. У Unix и Linux превосходная история развития журналирования — при желании в файлах журналов можно регистрировать любой чих, происходящий в вашей системе. Это обеспечивается системной службой журналирования.

В Gentoo предлагается несколько служб журналирования на выбор. Это **syslogd**, традиционный набор журналирующих демонов, **syslog-ng**, расширенная служба журналирования, и **metalog** — служба журналирования с очень гибкими возможностями настройки. Возможно, в Portage имеются и другие службы

журналирования: количество доступных пакетов растет день ото дня.

Если вы планируете использовать **sysklogd** или **syslog-ng**, позднее может потребоваться установка **logrotate**, поскольку в этих службах журналирования не предусмотрен никакой механизм ротации системных журналов.

Чтобы установить выбранную службу журналирования, воспользуйтесь **emerge**, а затем добавьте ее в уровень запуска по умолчанию с помощью **rc-update**. В следующем примере показана установка **syslog-ng**. Вместо нее вы можете подставить другую службу журналирования:

**Листинг 2:** Установка системной службы журналирования

```
# emerge syslog-ng
# rc-update add syslog-ng default
```

## 9.c. Дополнительно: демон cron

Следующий демон — **cron**. Хотя он является дополнительным, и не обязателен для работы вашей системы, будет благоразумным установить его. Но что же такое демон **cron**? Демон **cron** выполняет команды по расписанию. Это очень удобно, когда нужно выполнять какие-либо команды регулярно (например, ежедневно, еженедельно или ежемесячно).

В Gentoo предлагаются три варианта демона **cron** на выбор: **dcron**, **fcron** и **vixie-cron**. Установка любого из них подобна установке системной службы журналирования. Однако, при установке **dcron** или **fcron** для настройки требуется выполнение дополнительной команды, а именно **crontab /etc/crontab**. Если вы не знаете, что выбрать, используйте **vixie-cron**.

При бессетевой установке доступен только **vixie-cron**. Если вам нужен другой демон **cron**, его можно установить позже.

**Листинг 3:** Установка демона **cron**

```
# emerge vixie-cron
# rc-update add vixie-cron default
(только если выбран dcron или fcron) # crontab /etc/crontab
```

## 9.d. Дополнительно: индексация файлов

Если вы хотите индексировать файлы в своей системе, чтобы быстро находить их с помощью **locate**, нужно установить **sys-apps/slocate**:

**Листинг 4:** Установка **slocate**

```
# emerge slocate
```

## 9.e. Утилиты для обслуживания файловых систем

Для проверки целостности файловых систем, создания дополнительных

файловых систем, и т.п., вам потребуются определенные утилиты, состав которых зависит от используемых файловых систем.

В следующей таблице перечислены утилиты, которые необходимо устанавливать для обслуживания файловых систем различных типов:

Файловая система	Утилиты	Команда установки
XFS	xfspgrog	emerge xfspgrog
ReiserFS	reiserfspgrog	emerge reiserfspgrog
JFS	jfsutils	emerge jfsutils

Если вы используете EVMS, также необходимо установить evms:

**Листинг 5:** Установка утилит EVMS  
**# USE="-gtk" emerge evms**

**USE="-gtk"** предотвратит установку пакетов, от которых зависит данный. При желании включить графические средства evms, потом можно перекомпилировать evms.

Если вам не нужны никакие дополнительные сетевые средства (типа **rp-pppoe** или клиента dhcp), переходите к настройке начального загрузчика.

## 9.f. Сетевые средства

Дополнительно: установка клиента DHCP

Если требуется, чтобы Gentoo автоматически получала IP-адрес для ваших сетевых интерфейсов, необходимо установить dhcpd (или любой другой клиент DHCP, список имеющихся клиентов DHCP см. в главе Модульное построение сети). Если не сделать этого сейчас, то после завершения установки вы не сможете подключиться к интернету!

**Листинг 6:** Установка dhcpd  
**# emerge dhcpd**

Дополнительно: установка клиента PPPoE

Если для выхода в сеть требуется rp-pppoe, его нужно установить.

**Листинг 7:** Установка rp-pppoe  
**# USE="-X" emerge rp-pppoe**

**USE="-X"** запрещает установку xorg-x11 в порядке зависимости (в rp-pppoe есть графические средства; если их нужно подключить, можно перекомпилировать rp-pppoe позже, или же установить xorg-x11 сейчас: при его установке потребуется много времени на компиляцию).

Дополнительно: утилиты RAID для оборудования IBM

Если в составе системы на базе POWER5 вы используете RAID-массивы SCSI, стоит задуматься об установке `iprutils`, которые, среди прочего, позволяют работать с дисковыми массивами, выяснять состояние дисков в составе массива и обновлять микрокод.

**Листинг 8:** Установка `iprutils`  
`# emerge iprutils`

Теперь переходите к настройке начального загрузчика.

## 10. Настройка начального загрузчика

### 10.a. Выбор загрузчика

#### Введение

Теперь, когда ядро настроено и собрано, а нужные конфигурационные файлы заполнены как надо, пришло время установить программу, которая будет запускать ваше ядро при старте системы. Такого рода программа называется загрузчиком. Для архитектуры x86 в Gentoo Linux есть загрузчики GRUB и LILO. Но прежде, чем приступить к установке одного из двух загрузчиков, мы расскажем, как настроить кадровый буфер (естественно, если вы захотите). С помощью кадрового буфера можно работать в командной строке Linux на фоне графических элементов (например, симпатичного изображения из загрузочной заставки Gentoo).

#### Дополнительно: кадровый буфер

Если вы настроили в ядре поддержку кадрового буфера (или оставили настройки по умолчанию при использовании **genkernel**), вы можете включить буфер, добавив параметр `vga` и/или `video` в файл конфигурации своего загрузчика.

Для начала, вам надо узнать тип используемого кадрового буфера. При использовании исходных кодов ядра, доработанных для Gentoo (таких как `gentoo-sources`), у вас есть возможность выбрать `vesafb-tng` в качестве типа драйвера VESA (в этих исходных кодах ядра он используется по умолчанию). При использовании `vesafb-tng` параметр `vga` указывать не нужно. В других случаях используется драйвер `vesafb`, а параметр `vga` необходимо указывать.

Параметр `vga` устанавливает разрешение и глубину цвета, используемые кадровым буфером **vesafb**. Как отмечено в `/usr/src/linux/Documentation/fb/vesafb.txt` (который устанавливается в составе пакета с деревом исходных кодов ядра), кадровому буферу необходимо передавать код режима VESA, соответствующий нужному разрешению и цветности.

В следующей таблице приведены коды режимов для указания в параметре `vga`, а также соответствующие им значения разрешения и цветности.

	640x480	800x600	1024x768	1280x1024
256	0x301	0x303	0x305	0x307
32 тыс	0x310	0x313	0x316	0x319

<b>64 тыс</b>	0x311	0x314	0x317	0x31A
<b>16 млн</b>	0x312	0x315	0x318	0x31B

Параметр **video** отвечает за порядок отображения кадрового буфера. В нем указывается название драйвера кадрового буфера (**vesafb** для ядра 2.6 или **vesa** для ядра 2.4), а затем режимы, которые нужно включить. Все режимы приведены в `/usr/src/linux/Documentation/fb/vesafb.txt`, а здесь мы расскажем о трех самых используемых:

1. **ywrap** считать, что видеоплата может закольцовывать свою память (продолжать выборку с начального адреса, когда достигнут конечный);
2. **mtrr:n** установка регистров MTRR; допустимые значения n: 0 - отключено 1 - без кэширования 2 - отложенная запись (write-back) 3 - объединенная запись (write-combining) 4 - сквозная запись (write-through);
3. **mode** (только для **vesafb-tng**) Установить разрешение, цветность и частоту кадров. Например, **1024x768-32@85** для разрешения 1024x768, 32-битного цвета и частоты кадров 85 Гц.

В результате должно получиться что-то вроде **vga=0x318 video=vesafb:mtrr:3,ywrap** или **video=vesafb:mtrr:3,ywrap,1024x768-32@85**. Запомните (или запишите) составленное значение, скоро оно понадобится.

Перейдем к установке GRUB или LILO.

## 10.b. По умолчанию: использование GRUB

### Введение в терминологию GRUB

Самое сложное в освоении **GRUB** — освоиться с тем, как в нем именуются жесткие диски и разделы. Ваш Linux-раздел **/dev/hda1**, скорее всего, в **GRUB** станет называться **(hd0,0)**. Обратите внимание на круглые скобки вокруг **hd0,0** — они **обязательны**.

Жесткие диски нумеруются, начиная с нуля, а не с «а»; разделы — с нуля, а не с единицы. Нужно помнить, что в виде **hd** нумеруются только жесткие диски, но не устройства **atapi-ide**, такие как приводы компакт-дисков. Та же нумерация используется для устройств **SCSI** (обычно им присваиваются номера большие, чем устройствам **IDE**, кроме случаев, когда BIOS настроен на загрузку с устройства **SCSI**). Когда BIOS настроен на загрузку с другого жесткого диска (например, с первичного ведомого), именно этот жесткий диск и становится **hd0**.

Например, если у вас есть жесткий диск **/dev/hda**, привод CD-ROM **/dev/hdb**, записывающий CD **/dev/hdc**, второй жесткий диск **/dev/hdd**, а устройств **SCSI** нет, то разделу **/dev/hdd7** будет соответствовать **(hd1,6)**. Возможно, это покажется запутанным (так и есть), но, как мы увидим, в **GRUB** есть механизм автодополнения по **tab**, облегчающий жизнь обладателям множества жестких дисков и разделов, а также тем, кто теряется в схеме нумерации устройств **GRUB**.

Почувствовав, что к чему, пора установить GRUB.

## Установка GRUB

Для установки GRUB сначала добавим его в систему:

**Листинг 1:** Установка GRUB  
**# emerge grub**

Хотя GRUB уже установлен, нам еще потребуется подправить его файл конфигурации, и поместить GRUB в MBR, чтобы он автоматически загружал ядро. С помощью nano (или другого редактора) создайте /boot/grub/grub.conf:

**Листинг 2:** Создание /boot/grub/grub.conf  
**# nano -w /boot/grub/grub.conf**

Теперь заполним grub.conf своими значениями. Ниже приведены два варианта grub.conf для показанного примера разбиения дисков. Первый вариант grub.conf мы подробно прокомментировали. Удостоверьтесь, что у себя вы указываете имя своего файла образа ядра, и при необходимости имя своего образа начального корневого диска (initrd).

первый вариант grub.conf — для тех, кто при сборке ядра обходился без genkernel

второй вариант grub.conf — для тех, кто при сборке ядра пользовался genkernel

**Примечание:** Если ваша корневая файловая система — JFS, необходимо добавить «го» в строку kernel, поскольку JFS «накатывает» свой журнал перед тем, как разрешить монтирование раздела на чтение-запись.

**Листинг 3:** grub.conf для тех, кто обошелся без genkernel  
# какой пункт загружать по умолчанию: 0 - первый, 1 - второй и т.д.  
**default 0**  
# сколько секунд ждать до начала загрузки пункта по умолчанию  
**timeout 30**  
# симпатичная заставка, добавить по вкусу :)  
# прокомментируйте, если у вас не установлена графическая видеоплата  
**splashimage=(hd0,0)/boot/grub/splash.xpm.gz**

**title=Gentoo Linux 2.6.12-r10**  
# раздел с файлом образа ядра (или операционной системой)  
**root (hd0,0)**  
**kernel /boot/kernel-2.6.12-gentoo-r10 root=/dev/hda3**

# следующие четыре строки нужны только для двойной загрузки с Windows  
# в этом примере Windows находится на /dev/hda6  
**title=Windows XP**  
**rootnoverify (hd0,5)**  
**makeactive**  
**chainloader +1**

**Листинг 4:** grub.conf для тех, кто пользовался genkernel  
**default 0**  
**timeout 30**  
**splashimage=(hd0,0)/boot/grub/splash.xpm.gz**

**title=Gentoo Linux 2.6.12-r10**

```
root (hd0,0)
kernel /boot/kernel-genkernel-x86-2.6.12-gentoo-r10 root=/dev/ram0 init=/linuxrc ramdisk=8192
real_root=/dev/hda3 udev
initrd /boot/initramfs-genkernel-x86-2.6.12-gentoo-r10
```

# нужно только для двойной загрузки

```
title=Windows XP
rootnoverify (hd0,5)
makeactive
chainloader +1
```

**Примечание:** Параметр `udev`, указанный в конце строки `kernel`, необходим для обхода ошибки в некоторых версиях `genkernel`, если вы вообще используете `udev` (по умолчанию — используется).

**Примечание:** Если вы разбили жесткий диск по-другому, или у вас другое ядро, внесите необходимые изменения. При этом убедитесь, что все пути, следующие за упоминанием устройства GRUB (например **(hd0,0)**), приведены относительно точки подключения, а не корня файловой системы. Другими словами, `(hd0,0)/grub/splash.xpm.gz` — на самом деле `/boot/grub/splash.xpm.gz`, так как **(hd0,0)** — это `/boot`.

Кроме того, если вы выбрали другую схему разбиения диска, и не выделяли для `/boot` отдельный раздел, префикс `/boot`, использованный в примерах выше, обязателен. Если же вы следовали рекомендованному нами плану разбиения, префикс `/boot` не требуется, но все работает благодаря символьной ссылке `boot`. Короче говоря, приведенные примеры должны работать независимо от того, есть у вас отдельный раздел для `/boot` или нет.

Если вам надо передать ядру дополнительные параметры, просто добавьте их в конец строки `kernel`. Один параметр мы уже передаем ядру (`root=/dev/hda3` или `real_root=/dev/hda3`); можно добавлять и другие, например, такие как параметры `video` и/или `vga` для кадрового буфера, обсуждавшиеся выше.

Если вы используете ядро 2.6.7 или выше, а объем жесткого диска ограничили перемычками из-за того, что BIOS не в состоянии работать с дисками большого размера, вам потребуется добавить `hdx=stroke`.

Тем, кто использует **genkernel**, нужно помнить, что их ядро использует такие же загрузочные параметры, как на установочном компакт-диске. Например, если у вас есть устройства SCSI, следует передать ядру параметр `doscsi`.

Теперь сохраните `grub.conf` и выйдите из редактора. Вам по-прежнему необходимо записать GRUB в MBR (Master Boot Record), чтобы GRUB автоматически запускался при загрузке системы.

Разработчики GRUB рекомендуют использовать `grub-install`. Однако, на случай некорректной работы `grub-install` есть возможность записать GRUB вручную.

Переходите к разделу по умолчанию: установка GRUB с помощью `grub-install` или альтернатива: установка GRUB вручную.

### По умолчанию: установка GRUB с помощью `grub-install`

Для установки GRUB вам надо выполнить команду `grub-install`. Однако, `grub-install` не

заработает сам по себе, т.к. мы находимся в среде с измененным корневым каталогом. Нам нужно создать файл `/etc/mtab`, перечислив в нем все смонтированные файловые системы. К счастью, для этого есть очень легкий способ: просто скопируйте содержимое `/proc/mounts` поверх `/etc/mtab`, исключив строку `rootfs`, если вы не создавали отдельный загрузочный раздел. Следующая команда подойдет в обоих случаях:

**Листинг 5:** Создание `/etc/mtab`  
`# grep -v rootfs /proc/mounts > /etc/mtab`

Теперь мы можем установить GRUB, используя `grub-install`:

**Листинг 6:** Выполнение `grub-install`  
`# grub-install /dev/hda`

Если у вас есть вопросы о GRUB, пожалуйста, обратитесь к GRUB FAQ (англ.) или руководству по GRUB (англ.).

**Переходите к перезагрузке системы.**

### Альтернатива: установка GRUB вручную

Для начала настройки, введите `grub`. Вы увидите приглашение `grub>` — это командная строка `grub`. Теперь потребуется набрать команды, нужные для установки загрузочной записи GRUB на ваш жесткий диск.

**Листинг 7:** Запуск оболочки GRUB  
`# grub`

**Примечание:** Если у вас нет приводов для дискет, к приведенной команде добавьте `--no-floppy`, чтобы `grub` зря не опрашивал несуществующие дисководы.

В приводимом примере мы хотим установить GRUB так, чтобы он считывал нужную информацию с загрузочного раздела `/dev/hda1`, а загрузочная запись GRUB находилась в MBR (Master Boot Record) жесткого диска, чтобы первое, что мы видели после включения компьютера — это приглашение GRUB. Естественно, если вы при установке отклонялись от предлагаемой схемы, внесите необходимые поправки.

Находясь в GRUB, можно использовать автодополнение по клавише TAB. К примеру, если ввести «`root (»`, а затем TAB, появится список устройств (таких как `hd0`). Если ввести «`root (hd0,»` и нажать TAB, появится список для выбора раздела из возможных (такого как `hd0,0`).

Благодаря автодополнению установка GRUB не так сложна. Теперь приступим к настройке GRUB.

**Листинг 8:** Установка GRUB в MBR  
`grub> root (hd0,0)` (указание расположения раздела с `/boot`)  
`grub> setup (hd0)` (установка GRUB в MBR)  
`grub> quit` (выход из оболочки GRUB)

**Примечание:** Если вы хотите установить GRUB в определенный раздел вместо



MBR, команду `setup` потребуется исправить так, чтобы она указывала на нужный раздел. Например, команда для установки GRUB в `/dev/hda3` — `setup (hd0,2)`. Однако, так поступают немногие.

С дополнительными вопросами о GRUB, пожалуйста, обращайтесь к GRUB FAQ (англ.) или руководству по GRUB (англ.).

**Примечание:** В случае переустановки ядра в будущем, вам больше не потребуется копировать содержимое файлов. После компиляции ядра просто запускайте `make install`: копирование файлов и изменение конфигурации GRUB произойдет автоматически.

Переходите к перезагрузке системы.

## 10.с. Альтернатива: использование LILO

### Установка LILO

**LILO** (сокращение от **L**inux **L**Oader) — это проверенная временем рабочая лошадка среди загрузчиков Linux-систем. Но ей недостает ряда возможностей, которые есть в GRUB (и в том числе в этом заключается причина растущей популярности GRUB). LILO все еще используется, потому что на некоторых системах он работает, а GRUB — нет. Конечно же, он используется еще и потому, что многие просто знакомы с LILO и сроднились с ним. Так или иначе, в Gentoo поддерживаются оба загрузчика, и вы, видимо, решили использовать LILO.

Установка LILO в систему проста как пробка: просто используйте `emerge`.

**Листинг 9: Установка LILO**  
`# emerge lilo`

### Настройка LILO

Для настройки LILO нужно создать файл `/etc/lilo.conf`. Запустите свой любимый редактор (в руководстве мы для единообразия используем `nano`) и создайте файл.

**Листинг 10: Создание `/etc/lilo.conf`**  
`# nano -w /etc/lilo.conf`

Несколькими разделами раньше мы попросили вас запомнить название созданного файла образа ядра. В следующем примере используется предложенная нами схема разделения диска.

Пример разделен на две части:

- одна — для тех, кто не пользовался для сборки ядра `genkernel`;
- другая — для тех, кто при сборке ядра пользовался `genkernel`;

Удостоверьтесь, что у себя вы указываете имя своего файла образа ядра, и при необходимости имя своего образа начального корневого диска (`initrd`).

**Примечание:** Если ваша корневая файловая система — JFS, необходимо добавить «ro» в строку kernel, поскольку JFS «накатывает» свой журнал перед тем, как разрешить монтирование раздела на чтение-запись.

**Листинг 11:** Пример /etc/lilo.conf

```
boot=/dev/hda      # установка LILO в MBR
prompt             # предоставление шанса выбора другого варианта
timeout=50         # ожидание пять секунд до загрузки варианта по умолчанию
default=gentoo     # по истечении времени загрузка варианта gentoo

# для тех, кто не использует genkernel
image=/boot/kernel-2.6.12-gentoo-r10
label=gentoo       # название этого варианта
read-only          # запуск с корневой ФС только для чтения; не менять!
root=/dev/hda3     # расположение корневой файловой системы

# для тех, кто использует genkernel
image=/boot/kernel-genkernel-x86-2.6.12-gentoo-r10
label=gentoo
read-only
root=/dev/ram0
append="init=/linuxrc ramdisk=8192 real_root=/dev/hda3 udev"
initrd=/boot/initramfs-genkernel-2.6.12-gentoo-r10

# следующие две строки нужны только для двойной загрузки с Windows
# в этом примере Windows находится на /dev/hda6
other=/dev/hda6
label=windows
```

**Примечание:** Параметр udev, указанный в конце строки kernel, необходим для обхода ошибки в некоторых версиях genkernel, если вы вообще используете udev (по умолчанию — используется).

**Примечание:** Если вы разбили жесткий диск по-другому, или у вас другое ядро, внесите необходимые изменения.

Если нужно передать ядру дополнительные параметры, добавьте в соответствующий раздел файла выражение append. Например, добавим параметр video для включения кадрового буфера:

**Листинг 12:** Использование append для добавления параметров ядра

```
image=/boot/kernel-2.6.12-gentoo-r10
label=gentoo
read-only
root=/dev/hda3
append="video=vesafb:mtrr,ywrap,1024x768-32@85"
```

Если вы используете ядро 2.6.7 или выше, а объем жесткого диска ограничили перемычками из-за того, что BIOS не в состоянии работать с дисками большого размера, вам потребуется добавить hdx=stroke.

Тем, кто использует genkernel, нужно помнить, что их ядро использует такие же загрузочные параметры, как на установочном компакт-диске. Например, если у вас

есть устройства SCSI, следует передать ядру параметр doscsi.

Теперь сохраните файл и выйдите из редактора. Для окончания установки нужно запустить /sbin/lilo, чтобы LILO смог отразить настройки, сделанные в /etc/lilo.conf, в вашей системе (т.е. записался на диск). Имейте в виду, что при каждой установке нового ядра или изменении меню вам потребуется выполнять /sbin/lilo заново.

**Листинг 13:** Завершение установки LILO  
`# /sbin/lilo`

**Примечание:** При переустановке ядра вам больше не придется копировать файлы. Просто запустите `make install` после компиляции ядра; копирование файлов и изменение конфигурации LILO выполнится автоматически.

Переходите к перезагрузке системы.

## 10.d. Перезагрузка системы

Выйдите из изолированной среды и размонтируйте все смонтированные разделы. Затем введите ту самую волшебную команду, которую вы так заждались: `reboot`.

**Листинг 14:** Размонтирование всех разделов и перезагрузка  
`# exit`  
`cdimage ~# cd`  
`cdimage ~# umount /mnt/gentoo/boot /mnt/gentoo/dev /mnt/gentoo/proc /mnt/gentoo`  
`cdimage ~# reboot`

Естественно, не забудьте вынуть загрузочный компакт-диск, иначе он загрузится сам вместо вашей новой системы Gentoo.

Загрузив вновь установленную систему, переходите к завершению установки Gentoo.

## 11. Завершение установки Gentoo

### 11.a. Управление учетными записями

#### Добавление учетной записи для повседневной работы

Работа в учетной записи root (администратора) в системе Unix/Linux опасна, и ее следует всячески избегать. Поэтому настоятельно рекомендуется добавить учетную запись пользователя для повседневной работы.

Членством пользователя в группах определяется, какие действия он сможет выполнять. В следующей таблице перечислено несколько важных групп, в которые вы, возможно, захотите включать пользователей.

Группа	Описание
--------	----------

<b>audio</b>	возможность доступа к аудиоустройствам
<b>cdrom</b>	возможность прямого доступа к оптическим накопителям
<b>floppy</b>	возможность прямого доступа к гибким дискам
<b>games</b>	возможность играть в игры
<b>portage</b>	возможность использования emerge --pretend с правами пользователя
<b>usb</b>	возможность доступа к устройствам USB
<b>plugdev</b>	возможность монтирования и использования подключаемых устройств типа камер и USB-брелков
<b>video</b>	возможность доступа к средствам видеозахвата и выполнения аппаратного ускорения видео
<b>wheel</b>	возможность использования команды su

Например, для создания учетной записи пользователя по имени john, входящего в группы wheel, users и audio, сначала войдите в систему как root (только root может создавать учетные записи пользователей), а затем запустите useradd:

**Листинг 1:** Создание учетной записи на каждый день

**Login:** root

**Password:** (ваш пароль root)

```
# useradd -m -G users,wheel,audio -s /bin/bash john
```

```
# passwd john
```

**Password:** (введите пароль для john)

**Re-enter password:** (повторно введите пароль для подтверждения)

Если пользователю потребуется выполнить задачу от имени root, для временного получения привилегий root можно использовать su -. Другой способ — пользоваться пакетом sudo, который при правильной настройке вполне безопасен.

## 12. Чем заняться дальше?

### 12.a. Документация

Примите поздравления! У вас теперь появилась работающая система Gentoo. И что же делать дальше? Какие у вас появились возможности? На что стоит взглянуть прежде всего? Gentoo дает своим пользователям богатый выбор возможностей, а следовательно — и множество документированных (или не очень) свойств.

Вам обязательно нужно прочитать следующую часть настольной книги Gentoo, работа в Gentoo, в которой рассказывается, как поддерживать программное обеспечение в актуальном состоянии, как до устанавливать программы, что такое «USE-флаги», как происходит инициализация в Gentoo и т.д.

Если вас интересует оптимизация системы с точки зрения пользователя, или вы хотите узнать, как настроить полноценный «рабочий стол», обратитесь к более подробной документации по графической среде Gentoo. Кроме того, вы, возможно захотите прочитать наше руководство по локализации (англ.), что бы чувствовать себя более уютно.

**Примечание:** В настоящее время для русификации рекомендуется пользоваться

Также имеется настольная книга по безопасности в Gentoo (англ.), которую определенно стоит прочитать.

Полный список существующих материалов имеется на странице документации.

## 12.b. Gentoo в интернете

Естественно, мы всегда рады видеть вас на форумах Gentoo (англ.), как и на любом из множества IRC-каналов Gentoo (англ.).

Кроме того, мы можем предложить вашему вниманию несколько списков рассылки, открытых для всех наших пользователей. Сведения о порядке подписки находятся на той же странице.

На этом мы замолкаем, чтобы позволить вам в полной мере насладиться результатом установки :)

## В. Работа с Gentoo

### 1. Введение в Portage

#### 1.a. Добро пожаловать в Portage

Система Portage — вероятно, самое известное нововведение Gentoo в управлении программным обеспечением. Благодаря высокой гибкости и чрезвычайно богатым возможностям, она зачастую считается лучшим средством управления программным обеспечением из существующих в Linux.

Portage полностью написана на Python и Bash, и в результате полностью прозрачна для пользователей, поскольку оба — языки сценариев.

Большинство пользователей взаимодействует с Portage с помощью команды `emerge`. Эта глава не призвана заменить страницу справки `emerge`. Для просмотра всех возможных параметров команды `emerge`, обращайтесь к странице справки:

**Листинг 1:** Чтение страницы справки `emerge`  
`$ man emerge`

## 1.b. Дерево портежей

### Сборочные файлы ebuild

Говоря о пакетах, мы часто имеем в виду программы, доступные пользователям Gentoo через дерево портежей. Дерево портежей — это набор сборочных файлов ebuild, содержащих всю информацию, необходимую Portage для управления программным обеспечением (установки, поиска, извлечения и т.п.) По умолчанию сборочные файлы находятся в /usr/portage.

Когда Portage по вашему поручению выполняет любые действия над пакетами программ, эти действия опираются на сборочные файлы, имеющиеся в системе. Поэтому необходимо регулярно обновлять сборочные файлы, чтобы Portage знала о новых программах, обновлениях, связанных с безопасностью и т.д.

### Обновление дерева портежей

Дерево портежей обычно обновляется с помощью **rsync** (англ.), средства быстрой разностной передачи файлов. Обновление выполнить довольно просто, так как запуск rsync обеспечивается командой emerge :

**Листинг 2:** Обновление дерева портежей  
**# emerge --sync**

Если rsync выполнить невозможно из-за ограничений межсетевого экрана, дерево портежей все-таки можно обновить из ежедневных «снимков», создаваемых нами. Для автоматического извлечения и установки в системе новейшего снимка служит утилита **emerge-webrsync**:

**Листинг 3:** Запуск emerge-webrsync  
**# emerge-webrsync**

## 1.c. Обслуживание программного обеспечения

### Поиск программ

Для поиска программ в дереве портежей по названию можно использовать встроенные возможности команды emerge. По умолчанию команда **emerge --search** выдает названия пакетов, соответствующих (как полностью, так и частично) заданному условию поиска.

Например, чтобы найти все пакеты, содержащие «pdf» в названии:

**Листинг 4:** Поиск пакетов с pdf в названии  
**\$ emerge --search pdf**

Для поиска пакетов еще и по тексту описания можно использовать параметр **--searchdesc** (или **-S**):

**Листинг 5:** Поиск пакетов, связанных с pdf

```
$ emerge --searchdesc pdf
```

Посмотрев на сообщения команды, вы отметите, что вам дается множество информации. Поля четко обозначены, поэтому мы не будем вдаваться в подробности их значения:

**Листинг 6:** Пример вывода `emerge --search`

```
* net-print/cups-pdf
  Latest version available: 1.5.2
  Latest version installed: [ Not Installed ]
  Size of downloaded files: 15 kB
  Homepage:  http://cip.physik.uni-wuerzburg.de/~vrbehr/cups-pdf/
  Description: Provides a virtual printer for CUPS to produce PDF files.
  License:    GPL-2

(
* net-print/cups-pdf
  Последняя доступная версия: 1.5.2
  Последняя установленная версия: [ не установлен ]
  Размер загружаемых файлов: 15 kB
  Веб-страница: http://cip.physik.uni-wuerzburg.de/~vrbehr/cups-pdf/
  Описание:     Снабжает CUPS виртуальным принтером для печати PDF-файлов.
  Лицензия:     GPL-2 )
```

**Примечание (JohnBat26):** Для более быстрого поиска программ используйте утилиту: **eix** (установка **emerge eix**). Она работает быстрее благодаря индексированию.

## Установка программ

После того, как вы нашли нужное программное обеспечение, его можно легко установить с помощью команды `emerge`. Вот пример установки пакета **gnumeric**:

**Листинг 7:** Установка `gnumeric`  
**# emerge gnumeric**

Так как множество приложений зависит друг от друга, любая попытка установить какой-либо пакет программ может повлечь за собой также установку дополнительных пакетов. Не беспокойтесь, Portage справится и с этим. Если вы захотите выяснить, что именно Portage собирается установить вместе с нужным вам пакетом, добавьте параметр `--pretend`. Например:

**Листинг 8:** Проверка зависимостей пакета `gnumeric`  
**# emerge --pretend gnumeric**

После команды на установку пакета, Portage загружает из интернета необходимый исходный код (при необходимости), и по умолчанию сохраняет его в каталоге `/usr/portage/distfiles`. После этого пакет распаковывается, компилируется и устанавливается. Если вы хотите, чтобы Portage только загрузила исходный код без его установки, добавьте к команде `emerge` параметр `--fetchonly`:

**Листинг 9:** Загрузка исходного кода пакета `gnumeric`  
**# emerge --fetchonly gnumeric**

## Обнаружение документации к пакету

Многие пакеты содержат собственную документацию. Иногда USE-флаг `doc` определяет, следует ли устанавливать документацию к пакету. Проверить наличие USE-флага `doc` можно командой `emerge -vp <название пакета>`.

**Листинг 10:** Проверка наличия USE-флага `doc`

(alsa-lib - это всего лишь пример)

```
# emerge -vp alsa-lib
```

```
[ebuild N ] media-libs/alsa-lib-1.0.9_rc3 +doc -jack 674 kB
```

USE-флаг `doc` можно включить или отключить как глобально в файле `/etc/make.conf`, так и для отдельных пакетов в файле `/etc/portage/package.use`. Также можно, создав каталог с названием `/etc/portage/package.use`, указать флаг в файле внутри этого каталога. В главе USE-флаги этот вопрос описывается более подробно.

Документация от вновь установленного пакета обычно находится в подкаталоге каталога `/usr/share/doc`, соответствующем названию пакета. Кроме того, можно вывести список всех установленных файлов утилитой `equery`, которая входит в пакет `gentoolkit` (англ.) — `app-portage/gentoolkit`.

**Листинг 11:** Обнаружение документации пакета

```
# ls -l /usr/share/doc/alsa-lib-1.0.9_rc3
```

```
total 28
```

```
-rw-r--r-- 1 root root 669 May 17 21:54 ChangeLog.gz
```

```
-rw-r--r-- 1 root root 9373 May 17 21:54 COPYING.gz
```

```
drwxr-xr-x 2 root root 8560 May 17 21:54 html
```

```
-rw-r--r-- 1 root root 196 May 17 21:54 TODO.gz
```

(или используйте для поиска интересных файлов команду `equery` :)

```
# equery files alsa-lib | less
```

```
media-libs/alsa-lib-1.0.9_rc3
```

```
* Contents of media-libs/alsa-lib-1.0.9_rc3:
```

```
/usr
```

```
/usr/bin
```

```
/usr/bin/alsalisp
```

```
(выдача обрезана)
```

## Удаление пакета

Когда вы захотите удалить пакет из системы, используйте команду `emerge --unmerge`. Это приведет к удалению из системы всех файлов, установленных пакетом, кроме конфигурационных файлов приложения, изменявшихся после установки. Сохранение конфигурационных файлов позволяет вернуться к работе с пакетом, если вы когда-нибудь решите снова его установить.

**Внимание:** Portage не проверяет, зависят ли другие пакеты от удаляемого! Однако вы получите предупреждение, если удаление пакета приведет к неработоспособности системы.



**Листинг 12:** Удаление пакета `gnumeric` из системы  
**# emerge --unmerge gnumeric**

После удаления пакета из системы, пакеты, установленные автоматически, потому что от них зависел удаленный пакет, остаются. Чтобы Portage выявила все когда-то нужные пакеты, которые теперь можно удалить, используйте команду **emerge --depclean**. Мы вернемся к этому ниже.

## Обновление системы

Чтобы система сохранялась в отличной форме (не говоря уже об установке свежайших обновлений, связанных с безопасностью), ее нужно регулярно обновлять. Так как Portage просматривает сборочные файлы только в локальном дереве портежей, сперва потребуется обновить его. Обновив дерево портежей, вы сможете обновить систему командой `emerge --update world`. В следующем примере мы также пользуемся параметром `--ask`, который поручает Portage вывести список пакетов, которые она собирается обновить, и спросить вас, можно ли продолжать:

**Листинг 13:** Обновление системы  
**# emerge --update --ask world**

Portage будет искать более новые версии установленных приложений. Однако проверяется только версии приложений, явно установленных вами, а не тех, от которых они зависят. Если вы хотите обновить каждый пакет в системе, добавьте аргумент `--deep`:

**Листинг 14:** Обновление всей системы  
**# emerge --update --deep world**

Поскольку обновления, относящиеся к безопасности, случаются и в пакетах, явным образом не устанавливались (но были «подтянуты» из-за того, что он них зависят другие программы), рекомендуется изредка запускать эту команду.

Если вы меняли какие-либо из USE-флагов, возможно, потом вы также захотите добавить параметр `--newuse`. Тогда Portage проверит, требует ли изменение установки новых пакетов или перекомпиляции существующих:

**Листинг 15:** Выполнение полного обновления  
**# emerge --update --deep --newuse world**

## Метапакеты

У некоторых пакетов в дереве портежей нет содержимого как такового, и они используются для установки набора других пакетов. Например, пакет **kde** полностью устанавливает среду KDE в вашей системе, привлекая различные KDE-пакетов в качестве зависимостей.

Если вы когда-либо захотите удалить из системы такой пакет, запуск **emerge--unmerge** не возымеет должного эффекта, так как пакеты, от которых он зависит, останутся в системе.

В Portage существует возможность удаления остаточных зависимостей, но так как зависимости программ меняются со временем, доступность программного обеспечения, прежде всего требуется полностью обновить всю систему, включая реализацию изменений, произведенных путем модификации USE-флагов. После этого можно запустить **emerge --depclean**, чтобы удалить остаточные зависимости. Когда это сделано, вам потребуется пересобрать приложения, ранее динамически связанные с удаленными пакетами, в которых они теперь не нуждаются.

Со всем этим управляются следующие три команды:

**Листинг 16:** Удаление ненужных зависимостей  
**# emerge --update --deep --newuse world**  
**# emerge --depclean**  
**# revdep-rebuild**

**revdep-rebuild** входит в пакет **gentoolkit**; не забудьте сначала его установить:

**Листинг 17:** Установка пакета gentoolkit  
**# emerge gentoolkit**

## 1.d. Когда Portage жалуется...

### Слоты, виртуалы, ветви, архитектуры и профили

Как уже сказано, **Portage** — чрезвычайно мощная система, поддерживающая множество возможностей, не хватающих другим системам управления программами. Чтобы это стало понятно, разберем несколько аспектов Portage, не вникая в подробности.

С помощью Portage разные версии отдельного пакета могут сосуществовать в одной системе. В то время, как другие системы управления стремятся называть пакеты в соответствии с версией (например **freetype** и **freetype2**), **в Portage используется технология слотов (SLOT), или областей**. Пакет присваивает определенный слот своей версии. Пакеты с разными слотами способны сосуществовать в одной системе. Например, у пакета **freetype** есть ebuild как со **SLOT="1"**, так и со **SLOT="2"**.

Существуют также пакеты, выполняющие одни и те же функции, но отличающиеся в реализации. Например **metalogd**, **sysklogd** и **syslog-ng** являются системными службами журналирования. Приложения, использующие «системный журнал», не могут зависеть от одной конкретной программы, например от **metalogd**, так как остальные программы ничем не хуже. В Portage предусмотрены виртуальные пакеты: каждая служба журналирования предоставляет **virtual/syslog**, и в результате в приложениях можно указывать зависимость от **virtual/syslog**.

Программное обеспечение может располагаться в различных ветвях дерева портежей. По умолчанию в системе разрешено только использование стабильных пакетов. Большинство новых программ при поступлении включаются в тестовую ветвь, что указывает на необходимость дополнительного тестирования перед тем, как включить их в стабильные. Хотя в дереве портежей и видны сборочные файлы для таких программ, Portage не станет обновлять их до тех пор, пока они не будут

помещены в стабильную ветвь.

Некоторые программы имеются не для всех архитектур. Либо они не работают в определенных архитектурах, либо требуют дополнительного тестирования, или у разработчика нет возможности проверить, работает ли пакет в различных архитектурах.

Каждая установка Gentoo придерживается определенного профиля, который содержит, помимо прочего, список пакетов, необходимых для работоспособности системы.

## Блокировка пакетов

**Листинг 18:** Предупреждение о заблокированных пакетах (с `--pretend`)  
[blocks B ] mail-mta/ssmtp (is blocking mail-mta/postfix-2.2.2-r1)

**Листинг 19:** Предупреждение о заблокированных пакетах (без `--pretend`)  
!!! Error: the mail-mta/postfix package conflicts with another package.  
!!! both can't be installed on the same system together.  
!!! Please use 'emerge --pretend' to determine blockers.

( !!! Ошибка: пакет mail-mta/postfix конфликтует с другим пакетом.  
!!! оба не могут находиться в системе одновременно. Пожалуйста,  
!!! запустите 'emerge --pretend' для выявления блокирующих пакетов. )

В файлах ebuild есть специальные поля, сообщающие Portage о зависимостях. Возможны два вида зависимости: зависимость сборки, объявленная в **DEPEND**, и зависимость выполнения, объявленная в **RDEPEND**. Когда одна из этих зависимостей явно указывает на несовместимость пакета или виртуального пакета, это вызывает блокировку.

Для разблокировки можно отказаться от установки пакета или предварительно удалить конфликтующего пакета. В данном примере можно отказаться от установки **postfix** или сначала удалить **ssmtp**.

Также возможно, что два пакета, подлежащие установке, блокируют друг друга. В этом редчайшем случае следует определить, зачем вам устанавливать оба пакета. В большинстве случаев можно обойтись одним. Если это окажется не так, то, пожалуйста, заведите отчет об ошибке в системе распределения запросов Gentoo.

## Маскировка пакетов

**Листинг 20:** Предупреждение о замаскированных пакетах  
!!! all ebuilds that could satisfy "bootsplash" have been masked.

(!!! все сборки, удовлетворяющие "bootsplash", замаскированы.)

**Листинг 21:** Предупреждение о замаскированных пакетах с указанием причины  
!!! possible candidates are:

- gnome-base/gnome-2.8.0\_pre1 (masked by: ~x86 keyword)
- lm-sensors/lm-sensors-2.8.7 (masked by: -sparc keyword)
- sys-libs/glibc-2.3.4.20040808 (masked by: -\* keyword)
- dev-util/cvsd-1.0.2 (masked by: missing keyword)

- media-video/ati-gatos-4.3.0 (masked by: package.mask)  
- sys-libs/glibc-2.3.2-r11 (masked by: profile)

( !!! возможные кандидаты:

- gnome-base/gnome-2.8.0\_pre1 (маскировка: ключ ~x86)  
- lm-sensors/lm-sensors-2.8.7 (маскировка: ключ -sparc)  
- sys-libs/glibc-2.3.4.20040808 (маскировка: ключ -\*)  
- dev-util/cvsd-1.0.2 (маскировка: ключ отсутствует)  
- media-video/ati-gatos-4.3.0 (маскировка: package.mask)  
- sys-libs/glibc-2.3.2-r11 (маскировка: profile) )

Когда вы собираетесь установить пакет, не предназначенный для вашей системы, выдается ошибка маскировки. Нужно попытаться установить другую программу, существующую для вашей системы, или дождаться, пока пакет станет доступным. Всегда есть причина, по которой пакет замаскирован:

- ключ **~arch**: пакет недостаточно проверен для помещения в стабильную ветвь. Подождите несколько дней или недель и попробуйте установить его еще раз;
- ключ **-arch** или ключ **-\***: пакет не работоспособен в вашей архитектуре. Если вы полагаете, что он работает, сообщите об этом в **bugzilla**;
- ключ отсутствует: пакет еще не тестировался в вашей архитектуре. Попросите группу портирования в архитектуру проверить пакет, или протестируйте его за них и сообщите о своих изысканиях в bugzilla;
- **package.mask**: обнаружено повреждение пакета, нестабильность или что-то худшее, и пакет заблокирован специально;
- **profile**: пакет считается не предназначенным для вашего профиля. В случае установки приложение может вызвать сбой системы или просто несовместимо с используемым профилем.

## Отсутствие нужных пакетов

**Листинг 22:** Предупреждение об отсутствии пакета  
emerge: there are no ebuilds to satisfy ">=sys-devel/gcc-3.4.2-r4".

!!! Problem with ebuild sys-devel/gcc-3.4.2-r2  
!!! Possibly a DEPEND/\*DEPEND problem.

( emerge: нет сборок, удовлетворяющих ">=sys-devel/gcc-3.4.2-r4".

!!! Проблема с ebuild sys-devel/gcc-3.4.2-r2  
!!! Возможно, ошибка в DEPEND/\*DEPEND. )

Приложение, которое вы пытаетесь установить, зависит от другого пакета, недоступного вашей системе. Пожалуйста, проверьте, есть ли такой запрос в **bugzilla**, а если нет, сообщите об ошибке. Если вы не смешиваете ветви, такого не должно происходить, и это — явная ошибка.

## Неоднозначность названия пакета

**Листинг 23:** Предупреждение о повторяющихся именах ebuild  
!!! The short ebuild name "aterm" is ambiguous. Please specify

!!! one of the following fully-qualified ebuild names instead:

```
dev-libs/aterm
x11-terms/aterm
```

( !!! Короткое название ebuild "aterm" неоднозначно. Пожалуйста,  
!!! вместо него укажите одно из полных названий ebuild:

```
dev-libs/aterm
x11-terms/aterm )
```

Название приложения, которое вы собираетесь установить, соответствует более чем одному пакету. Требуется также указать название категории. Portage предложит вам возможные варианты.

## Циклические зависимости

**Листинг 24:** Предупреждение Portage о циклических зависимостях  
!!! Error: circular dependencies:

```
ebuild / net-print/cups-1.1.15-r2 depends on ebuild /
app-text/ghostscript-7.05.3-r1
ebuild / app-text/ghostscript-7.05.3-r1 depends on ebuild /
net-print/cups-1.1.15-r2
```

( !!! Ошибка: циклические зависимости:

```
ebuild / net-print/cups-1.1.15-r2 зависит от ebuild /
app-text/ghostscript-7.05.3-r1
ebuild / app-text/ghostscript-7.05.3-r1 зависит от ebuild /
net-print/cups-1.1.15-r2 )
```

Два или более пакета, которые вы хотите установить, взаимно зависимы, и в результате их установка невозможна. Скорее всего, это ошибка в дереве портежей. Пожалуйста, выждав время, обновите дерево портежей, и попробуйте снова. Вы можете также проверить, есть ли эта ошибка в bugzilla, и если нет, сообщить о ней.

## Ошибка извлечения

**Листинг 25:** Предупреждение Portage об ошибке извлечения  
!!! Fetch failed for sys-libs/ncurses-5.4-r5, continuing...

(...)

!!! Some fetch errors were encountered. Please see above for details.

( !!! Ошибка при извлечении sys-libs/ncurses-5.4-r5, продолжение...

(...)

!!! При извлечении произошли ошибки. Подробности выше. )

Portage не смогла загрузить исходный код данного приложения и попытается продолжить установку других приложений (если запланирована). Эта ошибка может произойти из-за неправильно синхронизированного зеркала, или из-за того, что ebuild указывает на неверное место. Сервер, где находятся исходные коды, также может почему-либо не работать.

Повторите действие через час, чтобы посмотреть, повторится ли эта ошибка.

## Защита системного профиля

**Листинг 26:** Предупреждение Portage о пакете, защищенном профилем  
!!! Trying to unmerge package(s) in system profile. 'sys-apps/portage'  
!!! This could be damaging to your system.

( !!! Попытка удаления пакетов из системного профиля. 'sys-apps/portage'  
!!! Это может повредить вашей системе. )

Вы попросили удалить пакет, входящий в состав базовых пакетов вашей системы. Он отмечен в вашем профиле как обязательный, и его не следует удалять из системы.

## 2. USE-флаги

### 2.а. Что такое USE-флаги?

#### Смысл USE-флагов

Устанавливая Gentoo (или любой другой дистрибутив, или даже операционную систему вообще), вы выбираете те или иные возможности в зависимости от среды, с которой работаете. Установка сервера отличается от установки рабочей станции, а установка игровой станции — от платформы 3D-рендеринга.

Это касается не только того, какие пакеты устанавливать, но и какие функции определенных пакетов должны поддерживаться. Если вам не нужен OpenGL, то зачем вам его ставить и встраивать поддержку OpenGL в большинство программ? Если вы не собираетесь использовать KDE, зачем собирать пакеты с его поддержкой, если они работают и без этого?

Чтобы помочь пользователям в выборе того, что устанавливать/активировать, а что — нет, мы захотели дать им простой способ описания рабочей среды. Это позволяет пользователю решить, что же ему на самом деле надо, и облегчить работу с Portage, нашей системой управления пакетами.

#### Определение USE-флагов

Рассмотрим USE-флаги. USE-флаг — это ключевое слово, включающее сведения о поддержке и зависимостях определенного понятия или функции. При определении какого-либо USE-флага, Portage узнает, что вам нужна поддержка соответствующей функции. Конечно, это также влияет на сведения о зависимостях пакета.

Давайте рассмотрим конкретный пример — ключевое слово **kde**. Если в вашей переменной USE нет этого слова, то все пакеты, где поддержка KDE является необязательной, собираются без нее. Все пакеты, где зависимость от KDE является необязательной, устанавливаются без установки библиотек KDE (по зависимости). Если же вы определите ключевое слово **kde**, то эти пакеты будут собираться с поддержкой KDE, а KDE будет установлен в качестве необходимого.

Правильно определяя ключевые слова, вы создаете систему, подогнанную специально для ваших нужд.

## Какие USE-флаги существуют?

Есть два типа USE-флагов: **глобальные и локальные**.

**Глобальный** USE-флаг используется несколькими пакетами и является системным. Это то, что большинство видит в качестве USE-флагов.

**Локальный** USE-флаг используется единичным пакетом для настройки определенных параметров самого пакета.

Список доступных глобальных USE-флагов можно найти в сети или локально в `/usr/portage/profiles/use.desc`.

Список локальных USE-флагов находится в вашей системе в `/usr/portage/profiles/use.local.desc`.

**Примечание (JohnBat26):** Для удобной установки и чтения описания USE флагов используйте программу: **Use Flag Editor** или **ufed (emerge ufed)**

## 2.b. Использование USE-флагов

### Объявление постоянных USE-флагов

В надежде, что вы убедились в важности USE-флагов, теперь мы расскажем, как их объявлять.

Как сказано ранее, все USE-флаги объявляются в переменной USE. Чтобы упростить пользователям поиск и выбор флагов, мы предлагаем значение USE по умолчанию, которое представляют собой список USE-флагов, как нам кажется, наиболее часто используемых пользователями Gentoo. Это значение приведено в файле `make.defaults` вашего профиля.

Профиль, на который ориентируется ваша система, указывается символьной ссылкой `/etc/make.profile`. Каждый профиль основывается на предыдущем, более крупном, а итоговый складывается из всех профилей. Верхним является базовый профиль (`/usr/portage/profiles/base`).

Давайте взглянем на значение по умолчанию для профиля 2004.3:

**Листинг 1:** Итоговая переменная USE `make.defaults` для профиля 2004.3

(в этом примере объединяются значения из `base`, `default-linux`, `default-linux/x86` и `default-linux/x86/2004.3`)

```
USE="x86 oss apm arts avi berkdb bitmap-fonts crypt cups encode fortran f77
foomaticdb gdbm gif gpm gtk imlib jpeg kde gnome libg++ libwww mad
mikmod motif mpeg ncurses nls oggvorbis opengl pam pdflib png python qt
quicktime readline sdl spell ssl svga tcpd truetype X xml2 xmms xv zlib"
```

Как видите, эта переменная уже содержит достаточно много ключевых слов. Не меняйте файл `make.defaults` для настройки переменной USE под свои нужды: изменения в этом файле аннулируются при обновлении Portage!

Для изменения значения по умолчанию, нужно добавлять или удалять ключевые слова из переменной USE. Это делается глобально, определением переменной USE в `/etc/make.conf`. В эту переменную можно добавить нужные вам USE-флаги, или удалить ненужные. Для удаления флага, его надо указывать со знаком минус в виде

приставки («-»).

Например, чтобы убрать поддержку **KDE** и **QT**, но добавить поддержку **Idap**, можно определить в `/etc/make.conf` переменную `USE` следующего вида:

**Листинг 2:** Пример значения `USE` в `/etc/make.conf`  
`USE="-kde -qt Idap"`

## Объявление `USE`-флагов для отдельных пакетов

Иногда нужно определить некоторые `USE`-флаги только для одного или нескольких пакетов, не трогая системных настроек. Для этого необходимо создать каталог `/etc/portage` (если его еще нет) и отредактировать файл `/etc/portage/package.use`.

Например, вам не нужна глобальная поддержка `berkdb`, но она необходима в `mysql`:

**Листинг 3:** Пример `/etc/portage/package.use`  
`dev-db/mysql berkdb`

Естественно, можно в явном виде отключить `USE`-флаги для определенного пакета. Например, если вам не нужна поддержка `java` в `PHP`:

**Листинг 4:** Второй пример `/etc/portage/package.use`  
`dev-php/php -java`

## Объявление временных `USE`-флагов

Иногда необходимо установить какой-то `USE`-флаг только на один раз. Вместо того, чтобы дважды редактировать `/etc/make.conf` (сначала добавить изменения `USE`, а потом удалить), можно просто объявить `USE` как переменную среды. Помните, что при переустановке или обновлении приложения (явном или в составе обновления системы) ваши изменения будут утеряны!

Например, уберем **java** из значения `USE` на время установки `mozilla`.

**Листинг 5:** Использование `USE` в виде переменной среды  
`# USE="-java" emerge mozilla`

## Наследование

Конечно же, существует определенная последовательность формирования значения `USE`. Вы же не хотите объявить `USE="-java"` только для того, чтобы узнать, что `java` все еще включена из-за значения с более высоким приоритетом. Последовательность установки значения `USE` в порядке приоритета (от меньшего к большему) такова:

- значение `USE` по умолчанию, объявленное в файлах `make.defaults` в составе вашего профиля;
- значение, определенное пользователем в `/etc/make.conf` ;
- значение, указанное пользователем в `/etc/portage/package.use` ;
- значение, определенное пользователями в переменной среды.



Чтобы узнать, какие же настройки USE в конечном счете видит Portage, запустите `emerge --info`. Эта команда выводит значения всех переменных (включая USE), используемые Portage.

**Листинг 6:** Запуск `emerge --info`  
# `emerge --info`

## Адаптация всей системы под новые USE-флаги

Если вы изменили свои USE-флаги и хотите обновить всю систему в соответствии с новым значением USE, запустите `emerge` с параметром **--newuse**:

**Листинг 7:** Пересборка всей системы  
# `emerge --update -deep --newuse world`

Теперь запустите функцию Portage **depclean**, чтобы удалить условные зависимости, присутствующие в «старой» системе, но больше не нужные при новом составе USE-флагов.

Предупреждение: Запуск **emerge --depclean** является опасной операцией, которую следует использовать с осторожностью. Дважды проверьте список «ненужных» пакетов и убедитесь, что не удалятся нужные пакеты. В следующем примере мы добавляем ключ `-p`, чтобы `depclean` только перечислил пакеты, не удаляя их.

**Листинг 8:** Удаление ненужных пакетов  
# `emerge -p --depclean`

Когда **depclean** закончит свою работу, запустите **revdep-rebuild**, чтобы пересобрать программы, динамически связанные с библиотеками, входящими в потенциально удаленные пакеты. `revdep-rebuild` входит в пакет `gentoolkit`, так что не забудьте сначала установить его.

**Листинг 9:** Запуск `revdep-rebuild`  
# `revdep-rebuild`

После выполнения всех этих действий, ваша система будет полностью использовать новые значения USE-флагов.

## 2.с. USE-флаги отдельных пакетов

### Просмотр доступных USE-флагов

Возьмем, к примеру, `mozilla` — какие USE-флаги она может использовать? Чтобы это выяснить, запустим `emerge` с параметрами `--pretend` и `--verbose`:

**Листинг 10:** Просмотр используемых USE флагов  
# `emerge --pretend --verbose mozilla`  
These are the packages that I would merge, in order:

Calculating dependencies ...done!

```
[ebuild R ] www-client/mozilla-1.7.12-r2 USE="crypt gnome java mozsvg ssl
trueunicode xprint -debug -ipv6 -ldap -mozcalendar -mozdevelop -mozcompose
-moznoirc -moznomail -moznoxft -postgres -xinerama" 0 kB
```

**emerge** — не единственное средство для решения этой задачи. Существует программа, специально предназначенная для вывода информации о пакетах. Она называется **equery** и находится в пакете **gentoolkit**. Для начала установим этот пакет:

**Листинг 11:** Установка **gentoolkit**  
**# emerge gentoolkit**

Теперь для просмотра USE-флагов какого-нибудь пакета запустим **equery** с аргументом **uses**. Пусть это будет пакет **gnnumeric**:

**Листинг 12:** Запуск **equery** для просмотра доступных USE-флагов  
**# equery uses =gnnumeric-1.6.3 -a**  
[ Searching for packages matching =gnnumeric-1.6.3... ]  
[ Colour Code : set unset ]  
[ Legend : Left column (U) - USE flags from make.conf ]  
[ : Right column (I) - USE flags packages was installed with ]  
[ Found these USE variables for app-office/gnumeric-1.6.3 ]  
U I  
- - debug : Tells configure and the makefiles to build for debugging.  
Effects vary across packages, but generally it will at  
least add -g to CFLAGS. Remember to set FEATURES=nostrip too  
- - gnome : Adds GNOME support  
+ + python : Adds support/bindings for the Python language  
- - static : !!do not set this during bootstrap!! Causes binaries to be  
statically linked instead of dynamically

### 3. Возможности Portage

#### 3.а. Возможности Portage

В Portage есть несколько дополнительных возможностей (features), которые значительно улучшат ваше впечатление от Gentoo. Многие из этих возможностей полагаются на определенные программы, повышающие производительность, надежность, безопасность и т.п.

Для включения и выключения определенных возможностей Portage нужно редактировать в файле **/etc/make.conf** переменную **FEATURES**, в которой перечислены ключевые слова, разделенные пробелами, обозначающие различные возможности. Иногда для использования соответствующих возможностей потребуется установка дополнительных утилит.

Здесь перечислены не все возможности, поддерживаемые Portage. Полный перечень представлен на странице справки **make.conf**:

**Листинг 1:** Вызов страницы справки **make.conf**  
**\$ man make.conf**

Чтобы узнать, какие возможности включены по умолчанию, запустите `emerge --info` и найдите переменную `FEATURES` (или отфильтруйте ее с помощью `grep`):

**Листинг 2:** Выявление уже включенных возможностей  
`$ emerge --info | grep FEATURES`

## 3.b. Распределенная компиляция

### Использование `distcc`

**distcc** — программа, распределяющая компиляцию по нескольким, не обязательно одинаковым, машинам в сети. Клиент `distcc` посылает всю необходимую информацию на доступные серверы `distcc` (на которых выполняется `distccd`), чтобы они могли компилировать для клиента части исходного кода. Чистый выигрыш — более быстрая компиляция.

Подробная информация о **distcc** (и как заставить его заработать в Gentoo) находится в нашем описании **distcc** в Gentoo.

### Установка `distcc`

Distcc поставляется с графическим монитором (средством контроля), позволяющим отслеживать задачи, которые ваш компьютер отправляет на компиляцию. Если вы используете Gnome, тогда добавьте «gnome» к переменной `USE`. А если вы не пользуетесь Gnome, но при этом хотите пользоваться монитором, добавьте «gtk» к переменной `USE`.

**Листинг 3:** Установка `distcc`  
`# emerge distcc`

### Подключение поддержки Portage

Добавьте **distcc** к переменной **FEATURES** в файле `/etc/make.conf`. Затем отредактируйте переменную `MAKEOPTS`, как вам нравится. Известная рекомендация — указывать директиву «-jX», где X — число центральных процессоров, на которых работает `distccd` (включая текущий компьютер) плюс один; у вас могут получиться лучшие результаты и с другими значениями.

Теперь запустите `distcc-config` и введите список доступных серверов `distcc`. Для простоты примера, предположим, что доступные серверы DistCC — 192.168.1.102 (текущий компьютер), 192.168.1.103 и 192.168.1.104 (два «удаленных» компьютера):

**Листинг 4:** Настройка `distcc` для использования трех доступных серверов `distcc`  
`# distcc-config --set-hosts "192.168.1.102 192.168.1.103 192.168.1.104"`

Не забудьте также запустить демон `distccd`:

**Листинг 5:** Запуск демонов `distccd`  
`# rc-update add distccd default`  
`# /etc/init.d/distccd start`

## 3.с. Кэширование компиляции

### О средстве ccache

**ccache** — это быстрый кэш компилятора. Когда вы компилируете программу, он кэширует промежуточные результаты так, что всякий раз, когда вы перекомпилируете ту же самую программу, время компиляции значительно сокращается. В типичных случаях общее время компиляции может сокращаться в 5 —10 раз.

Если вы интересуетесь подробностями ccache, пожалуйста, посетите домашнюю страницу ccache.

### Установка ccache

Для установки ccache, выполните **emerge ccache**:

**Листинг 6:** Установка ccache  
**# emerge ccache**

### Подключение поддержки Portage

Откройте `/etc/make.conf` и добавьте ccache к переменной `FEATURES`. Затем добавьте новую переменную по имени `CCACHE_SIZE` (размер кэша), и установите её равной «2G»:

**Листинг 7:** Редактирование `CCACHE_SIZE` в `/etc/make.conf`  
`CCACHE_SIZE="2G"`

Для проверки работоспособности ccache, запросите статистику ccache. Из-за того, что Portage использует другой домашний каталог ccache, вам также потребуется установить переменную `CCACHE_DIR`:

**Листинг 8:** Просмотр статистики ccache  
**# CCACHE\_DIR="/var/tmp/ccache" ccache -s**

Домашний каталог ccache по умолчанию — `/var/tmp/ccache`; изменить это назначение можно, определив переменную `CCACHE_DIR` в `/etc/make.conf`.

Однако, при запуске ccache используется каталог по умолчанию, `${HOME}/.ccache`, вот почему при запросе статистики (Portage) ccache требуется определять переменную `CCACHE_DIR`.

### Использование ccache для компиляции Си не в Portage

Если вы хотите использовать ccache для компиляций не в Portage, добавьте `/usr/lib/ccache/bin` в начало вашей переменной `PATH` (перед `/usr/bin`). Это можно сделать, отредактировав `/etc/env.d/00basic`, который является первым

файлом среды, где определяется переменная PATH:

**Листинг 9:** Редактирование /etc/env.d/00basic  
PATH="/usr/lib/ccache/bin:/opt/bin"

### 3.d. Поддержка двоичных пакетов

#### Создание готовых (заранее собранных) пакетов

Portage поддерживает установку заранее собранных готовых пакетов. Несмотря на то, что в саму Gentoo не входят заранее собранные пакеты (за исключением снимков GRP), Portage можно настроить на полноценное управление готовыми пакетами.

Чтобы создать двоичный пакет, можно использовать **quickpkg**, если пакет уже установлен в вашей системе, или **emerge** с параметрами **--buildpkg** или **--buildpkgonly**.

Если вы хотите, чтобы Portage создавал двоичные пакеты из каждого пакета, который вы будете устанавливать, добавьте **buildpkg** к переменной **FEATURES**.

Расширенная поддержка создания наборов готовых пакетов имеются в **catalyst**. Для получения подробной информации о **catalyst**, пожалуйста, прочитайте справочное руководство по **catalyst** (англ.) и распространенные вопросы о **catalyst** (англ.).

#### Установка двоичных пакетов

Хотя в Gentoo такого хранилища нет, вы можете создать централизованное хранилище для заранее скомпилированных двоичных пакетов. Чтобы использовать такое хранилище, потребуется указать Portage путь к нему с помощью переменной **PORTAGE\_BINHOST**. Например, если двоичные пакеты находятся на **ftp://buildhost/gentoo**:

**Листинг 10:** Установка PORTAGE\_BINHOST в /etc/make.conf  
PORTAGE\_BINHOST="ftp://buildhost/gentoo"

При установке двоичных пакетов, указывайте в команде **emerge** параметр **--getbinpkg** вместе с параметром **--usepkg**. Первый указывает **emerge** загрузить двоичный пакет с сервера, определенного раньше, а второй сообщает **emerge**, что до загрузки исходных кодов и их компиляции сначала нужно попытаться установить этот двоичный пакет.

Например, чтобы установить **gnumeric** из двоичных пакетов:

**Листинг 11:** Установка двоичного пакета **gnumeric**  
**# emerge --usepkg --getbinpkg gnumeric**

Подробную информацию о параметрах установки двоичных пакетов можно найти на странице справки **emerge**:

**Листинг 12:** Чтение справки по **emerge**  
**\$ man emerge**

## 4. Сценарии инициализации

### 4.а. Уровни запуска

#### Процесс загрузки системы

При загрузке вашей системы по экрану пробегает много текста. Если присмотреться, заметно, что этот текст не меняется от загрузки к загрузке. Последовательность всех этих действий называется последовательностью загрузки и в той или иной степени постоянна.

Во-первых, загрузчик размещает в памяти образ ядра, который вы указали в файле его конфигурации. После этого ядро запускается. Когда ядро загружено и запущено, оно инициализирует относящиеся к ядру структуры и задания, и запускает процесс **init**.

Этот процесс удостоверяется, что все файловые системы (определенные в **/etc/fstab**) смонтированы и готовы к использованию. Затем он выполняет несколько сценариев, находящихся в каталоге **/etc/init.d**, которые запускают службы, необходимые для нормального запуска системы.

И, наконец, когда все сценарии выполнены, **init** подключает терминалы (чаще всего просто виртуальные консоли, которые видны при нажатии **ALT+F1**, **ALT+F2** и т.д.), прикрепляя к каждой консоли специальный процесс под названием **agetty**. Этот процесс впоследствии обеспечивает возможность входа в систему с помощью **login**.

#### Сценарии инициализации

Сейчас процесс **init** запускает сценарии из каталога **/etc/init.d** не просто в случайном порядке. Более того, запускаются не все сценарии из **/etc/init.d**, а только те, которые предписано исполнять. Решение о запуске сценария принимается в результате просмотра каталога **/etc/runlevels**.

Во-первых, **init** запускает все сценарии из **/etc/init.d**, на которые есть символичные ссылки из **/etc/runlevels/boot**. Обычно сценарии запускаются в алфавитном порядке, но в некоторых сценариях имеется информация о зависимостях от других сценариев, указывающая системе на необходимость их предварительного запуска.

Когда все сценарии, указанные в **/etc/runlevels/boot**, будут выполнены, **init** переходит к запуску сценариев, на которые есть символичные ссылки из **/etc/runlevels/default**. И снова запуск происходит в алфавитном порядке, пока в сценарии не встретится информация о зависимостях; тогда порядок изменяется для обеспечения правильного порядка запуска.

#### Как работает init

Конечно, **init** не принимает решений сам по себе. Ему необходим конфигурационный

файл, где описаны необходимые действия. Этот файл — **/etc/inittab**.

Если вы запомнили последовательность загрузки, описанную чуть ранее, вы вспомните, что первое действие **init** — это монтирование всех файловых систем. Это определяется в строке **/etc/inittab**, приведенной ниже:

**Листинг 1:** Строка инициализации системы из **/etc/inittab**  
si::sysinit:/sbin/rc sysinit

Этой строкой процессу **init** предписывается выполнить **/sbin/rc sysinit** для инициализации системы. Самой инициализацией занимается сценарий **/sbin/rc**, так что можно сказать, что **init** делает не слишком много — он просто делегирует задачу по инициализации системы другому процессу.

Во-вторых, **init** выполняет все сценарии, на которые есть символьные ссылки из **/etc/runlevels/boot**. Это определяется следующей строкой:

**Листинг 2:** Инициализация системы, продолжение  
rc::bootwait:/sbin/rc boot

И снова все необходимые действия выполняются сценарием **rc**. Заметьте, что параметр, переданный **rc** (**boot**), совпадает с названием используемого подкаталога в **/etc/runlevels**.

Теперь **init** проверяет свой конфигурационный файл, чтобы определить, какой уровень запуска использовать. Для этого из **/etc/inittab** считывается строка:

**Листинг 3:** Строка **initdefault**  
id:3:initdefault:

В приведенном примере (который подходит для подавляющего большинства пользователей Gentoo) номер уровня запуска — 3. Пользуясь этой информацией, **init** проверяет, что нужно выполнить для запуска уровня запуска 3:

**Листинг 4:** Определение уровней запуска  
l0:0:wait:/sbin/rc shutdown  
l1:S1:wait:/sbin/rc single  
l2:2:wait:/sbin/rc nonetwork  
l3:3:wait:/sbin/rc default  
l4:4:wait:/sbin/rc default  
l5:5:wait:/sbin/rc default  
l6:6:wait:/sbin/rc reboot

В строке, определяющей уровень 3, для запуска служб снова используется сценарий **rc** (на этот раз с аргументом **default**). Опять-таки, обратите внимание, что аргумент, передаваемый сценарию **rc**, совпадает с названием подкаталога из **/etc/runlevels**.

По окончании работы **rc**, **init** принимает решение о том, какие виртуальные консоли включить и какие команды выполнить в каждой из них:

**Листинг 5:** Определение виртуальных консолей  
c1:12345:respawn:/sbin/agetty 38400 tty1 linux

```
c2:12345:respawn:/sbin/agetty 38400 tty2 linux
c3:12345:respawn:/sbin/agetty 38400 tty3 linux
c4:12345:respawn:/sbin/agetty 38400 tty4 linux
c5:12345:respawn:/sbin/agetty 38400 tty5 linux
c6:12345:respawn:/sbin/agetty 38400 tty6 linux
```

## Что такое уровень запуска?

Как вы заметили, `init` применяет нумерацию для определения уровня запуска, который надо использовать. Уровень запуска — это то состояние, в котором запускается ваша система, он содержит набор сценариев (сценариев уровня запуска или сценариев инициализации [initscript]), которые следует выполнять, при входе и выходе из определенного уровня запуска.

В Gentoo определено семь уровней запуска: три служебных и четыре определяемых пользователем. Служебные называются **sysinit**, **shutdown** и **reboot**. Действия, совершаемые ими, в точности соответствуют их названиям: инициализация системы, выключение системы и ее перезагрузка.

Определяемые пользователем уровни — это те, которым соответствуют подкаталоги в `/etc/runlevels`: **boot**, **default**, **nonetwork** и **single**. Уровень **boot** запускает все службы, необходимые системе и используемые всеми остальными уровнями. Остальные уровни отличаются друг от друга запускаемыми службами: **default** используется для повседневной работы, **nonetwork** — для тех случаев, когда не требуется сеть, а **single** — при необходимости восстановления системы.

## Работа со сценариями инициализации

Сценарии, запускаемые процессом `rc`, называются сценариями инициализации. Каждый сценарий из `/etc/init.d` может запускаться с аргументами **start**, **stop**, **restart**, **pause**, **zap**, **status**, **inneed**, **iuse**, **needsme**, **usesme** и **broken**.

Для запуска, остановки или перезапуска службы (и всех, зависящих от нее) следует использовать **start**, **stop** и **restart**:

**Листинг 6:** Запуск postfix  
`# /etc/init.d/postfix start`

**Примечание:** Останавливаются или перезапускаются только те службы, которым необходима данная служба. Остальные зависимые службы (те, которые используют службу, но не нуждаются в ней) эта операция не затрагивает.

Если вы хотите остановить службу, но оставить зависимые от нее работающими, можно использовать аргумент **pause**:

**Листинг 7:** Остановка postfix без остановки зависимых служб  
`# /etc/init.d/postfix pause`

Чтобы узнать текущее состояние службы (запущена, остановлена, приостановлена и т.д.), можно использовать аргумент **status**:

**Листинг 8:** Информация о состоянии postfix  
`# /etc/init.d/postfix status`



Если указано, что служба работает, но вы знаете, что это не так, можно сбросить состояние на **stopped** (остановлена), используя аргумент **zap**:

**Листинг 9:** Сброс информации о состоянии postfix  
**# /etc/init.d/postfix zap**

Для того, чтобы выяснить зависимости службы, можно использовать аргументы **iuse** или **ineed**. С помощью **ineed** вы увидите те службы, которые действительно необходимы для правильного функционирования интересующей вас службы. С другой стороны, **iuse** покажет те службы, которые могут использоваться нашей службой, но не обязательны для ее работы.

**Листинг 10:** Запрос списка всех необходимых служб, от которых зависит Postfix  
**# /etc/init.d/postfix ineed**

Аналогично вы можете узнать, какие службы нуждаются в данной службе (**needsme**) или могут ее использовать (**usesme**):

**Листинг 11:** Запрос списка всех служб, которым необходим Postfix  
**# /etc/init.d/postfix needsme**

Наконец, можно просмотреть список служб, требующихся для данной, но отсутствующих в системе:

**Листинг 12:** Запрос списка служб, необходимых Postfix, но отсутствующих  
**# /etc/init.d/postfix broken**

## 4.b. Использование rc-update

### Что такое rc-update?

Система инициализации Gentoo использует дерево зависимостей для определения служб, которые запускаются в первую очередь. Т. к. это очень утомительное занятие, и мы не хотели, чтобы пользователь занимался этим вручную, мы разработали инструменты, упрощающие управление уровнями запуска и сценариями инициализации.

Используя **rc-update**, можно включать и исключать сценарии инициализации из уровней запуска. Из **rc-update** автоматически запускается сценарий **depscan.sh** для перестроения дерева зависимостей.

### Добавление и удаление служб

В процессе установки Gentoo вы уже добавляли сценарии инициализации в уровень запуска «default». В тот момент вы, возможно, не имели понятия, что такое «default» и зачем он нужен, но теперь вы все это знаете. Сценарию **rc-update** требуется второй аргумент, определяющий действие: **add** (добавить), **del** (удалить) или **show** (показать).

Для того, чтобы добавить или удалить сценарий, просто введите **rc-update** с аргументом **add** или **del**, затем название сценария и уровня запуска. Например:

**Листинг 13:** Удаление Postfix из уровня запуска default  
**# rc-update del postfix default**

По команде **rc-update show** выводится список всех доступных сценариев с указанием соответствующих уровней запуска:

**Листинг 14:** Получение информации о сценариях инициализации  
**# rc-update show**

## 4.с. Настройка служб

### Почему нужна дополнительная настройка?

Сценарии инициализации могут быть весьма сложны. Поэтому нежелательно допускать непосредственное редактирование сценария пользователями, т.к. это может привести в систему множество ошибок. Но, с другой стороны, необходимо правильно настроить службу. Например, может понадобиться передать службе дополнительные параметры.

Вторая причина, по которой настройки хранятся отдельно от самого сценария — это возможность обновления сценария без опасения, что все ваши настройки будут утеряны.

### Каталог `/etc/conf.d`

В Gentoo предусмотрен очень простой способ настройки служб: для каждого сценария, предполагающего настройку, в каталоге `/etc/conf.d` есть конфигурационный файл. Например, у сценария, запускающего **apache2** (под названием `/etc/init.d/apache2`) есть конфигурационный файл `/etc/conf.d/apache2`, где могут храниться нужные вам параметры, передаваемые серверу Apache 2 при запуске:

**Листинг 15:** Переменная, определенная в `/etc/conf.d/apache2`  
`APACHE2_OPTS="-D PHP4"`

Такие файлы настроек содержат одни переменные (наподобие `/etc/make.conf`), облегчая настройку служб. Это также позволяет нам давать больше информации о переменных (в комментариях).

## 4.d. Написание сценариев инициализации

### Мне тоже придется?..

Нет, написание сценариев инициализации обычно не требуется, т.к. Gentoo

содержит готовые сценарии для всех поддерживаемых служб. Однако, вы можете установить какую-либо службу, не используя систему Portage; в таком случае, вероятно, вам придется создавать сценарий инициализации самостоятельно.

**Внимание:** не используйте сценарий, идущий со службой, если он не написан специально для Gentoo: сценарии инициализации Gentoo не совместимы со сценариями, используемыми в других дистрибутивах!

## Структура

Основная структура сценария инициализации показана ниже.

**Листинг 16:** Основная структура сценария

```
#!/sbin/runscript

depend() {
    (информация о зависимостях)
}

start() {
    (команды, необходимые для запуска службы)
}

stop() {
    (команды, необходимые для остановки службы)
}

restart() {
    (команды, необходимые для перезапуска службы)
}
```

В любом сценарии должна быть определена функция `start()`. Все остальные разделы необязательны.

## Зависимости

Можно определять два типа зависимостей: **use** (использую) и **need** (нуждаюсь). Как упоминалось ранее, **need**-зависимость более строга, чем **use**-зависимость. Вслед за типом зависимости указывается название службы, от которой существует зависимость, или ссылка на виртуальную (**virtual**) зависимость.

**Виртуальная зависимость** — это зависимость от функций, предоставляемых службой, но не какой-то единственной службой. Сценарий может зависеть от службы системного журнала, но таких достаточно много (**metalogd**, **syslog-ng**, и и т.п.). Поскольку нельзя нуждаться в каждой из них (ни в одной вразумительной системе они не запущены все сразу), мы обеспечили предоставление виртуальной зависимости всеми этими службами.

Давайте взглянем на информацию о зависимостях postfix.

**Листинг 17:** Информация о зависимостях Postfix

```
depend() {
    need net
    use logger dns
    provide mta
}
```

}

Как можно увидеть, postfix:

- требует сеть (net): виртуальная зависимость, удовлетворяемая, например, /etc/init.d/net.eth0;
- использует журнал (logger): виртуальная зависимость, удовлетворяемая, например, /etc/init.d/syslog-ng;
- использует службу имен (dns): виртуальная зависимость, удовлетворяемая, например, /etc/init.d/named);
- предоставляет почтовый агент (mta): виртуальная зависимость, общая для всех программ — почтовых серверов.

## Порядок запуска

Иногда вам нужна не сама служба, а запуск вашей службы до (или после) другой службы, если та присутствует в системе (обратите внимание на условие: это уже не зависимость) и запускается на том же уровне запуска (отметьте условие: это относится только к службам из одинакового уровня запуска). Такую очередность можно указать, используя значения **before** (до) или **after** (после).

Например, рассмотрим значения для службы Portmap:

**Листинг 18:** Функция depend() службы Portmap

```
depend() {  
    need net  
    before inetd  
    before xinetd  
}
```

Также можно использовать знак «\*», чтобы охватить все службы данного уровня запуска, хотя это не рекомендуется.

**Листинг 19:** Запуск сценария первым на уровне запуска

```
depend() {  
    before *  
}
```

## Стандартные функции

Следом за разделом **depend()** вам потребуется определить функцию **start()**. В ней содержатся все команды, необходимые для запуска вашей службы. Рекомендуется применять функции **ebegin** и **epend** для сообщений пользователю о том, что происходит:

**Листинг 20:** Пример функции start()

```
start() {  
    ebegin "Запуск - моя_служба"  
    start-stop-daemon --start --quiet --exec /path/to/my_service  
    eend $?  
}
```

Если вам нужны дополнительные примеры функции **start()**, пожалуйста, прочитайте исходные коды сценариев инициализации, находящихся в каталоге `/etc/init.d`. Что касается команды **start-stop-daemon**, то на случай, если вам нужны дополнительные сведения, есть превосходная страница справки:

**Листинг 21:** Вызов страницы справки по `start-stop-daemon`  
`# man start-stop-daemon`

Другими функциями, которые можно определить — `stop()` и `restart()`. От вас не требуется определение этих функций! Система инициализации, применяемая нами, достаточно развита и в состоянии самостоятельно заполнить эти функции, если вы используете **start-stop-daemon**.

Синтаксис сценариев инициализации, применяемых в Gentoo, основан на оболочке Борна (Bourne Again Shell — `bash`), поэтому вы можете свободно использовать внутри своих сценариев `bash`-совместимые конструкции.

### Добавление дополнительных параметров

Если вы хотите ввести в сценарий дополнительные параметры, кроме упоминавшихся, нужно добавить к переменной `opts` название параметра и создать функцию с названием, соответствующим параметру. Например, для поддержки параметра `restartdelay`:

**Листинг 22:** Создание дополнительной функции `restartdelay`  
`opts="${opts} restartdelay"`

```
restartdelay() {  
  stop  
  sleep 3  # пауза в 3 секунды перед повторным запуском  
  start  
}
```

### Переменные для настройки служб

Для поддержки конфигурационного файла в каталоге **`/etc/conf.d`** ничего дополнительно делать не нужно: при запуске вашего сценария инициализации автоматически включаются следующие файлы (т.е., переменные из них становятся доступными):

```
/etc/conf.d/<ваш сценарий инициализации>  
/etc/conf.d/basic  
/etc/rc.conf
```

Если ваш инициализационный сценарий предоставляет виртуальную зависимость (например, `net`), то также включается файл, соответствующий этой зависимости (например, `/etc/conf.d/net`).

## 4.е. Изменение поведения уровней запуска

### Кто от этого выиграет?

Большинству пользователей ноутбуков знакома ситуация: дома вам нужен

запуск **net.eth0**, и наоборот, в дороге запуск **net.eth0** не нужен (так как сеть недоступна). В Gentoo можно изменять поведение уровней запуска по своему усмотрению.

Например вы можете создать второй загружаемый уровень запуска «по умолчанию», в котором будут другие сценарии. Затем при загрузке вы сможете выбрать, какой из уровней по умолчанию следует использовать.

### Использование программного уровня (softlevel)

Прежде всего, создайте каталог для своего второго уровня запуска «по умолчанию». Например, создадим уровень запуска **offline**:

**Листинг 23:** Создание каталога уровня запуска  
**# mkdir /etc/runlevels/offline**

Добавьте необходимые сценарии инициализации в только что созданный уровень запуска. Например, чтобы получить точную копию уровня **default**, за исключением **net.eth0**:

**Листинг 24:** Добавление нужных сценариев инициализации  
(копирование всех служб с уровня default в уровень offline)  
**# cd /etc/runlevels/default**  
**# for service in \*; do rc-update add \$service offline; done**  
(удаление ненужных сценариев с уровня offline)  
**# rc-update del net.eth0 offline**  
(просмотр сценариев, запускаемых на уровне offline)  
**# rc-update show offline**  
(часть выведенного списка)  
acpid | offline  
domainname | offline  
local | offline  
net.eth0 |

Теперь необходимо отредактировать конфигурацию загрузчика, добавив запись об уровне **offline**. Например, в файле **/boot/grub/grub.conf**:

**Листинг 25:** Добавление записи об уровне offline  
title Автономное использование Gentoo Linux  
root (hd0,0)  
kernel (hd0,0)/kernel-2.4.25 root=/dev/hda3 softlevel=offline

Вуаля, все готово. Теперь, если при загрузке вы выберете вновь созданную запись, то вместо default будет использоваться уровень offline.

### Использование загрузочного уровня (bootlevel)

Использование загрузочного уровня полностью аналогично использованию программного уровня. Единственная разница состоит в том, что вы определяете второй уровень «boot» вместо «default».

## 5. Переменные среды

### 5.а. Переменные среды

Что это такое?

**Переменная среды** — это именованный объект, который содержит информацию, используемую одним или несколькими приложениями.

Многие пользователи (особенно новички в Linux) находят этот подход несколько странным или неуправляемым. Но это впечатление ошибочно: используя переменные среды, можно очень легко изменить настройку разнообразных программ.

#### Важные примеры

В следующей таблице описывается ряд переменных, используемых в системе Linux. Примеры их значений приведены далее.

**Таблица 1.** Переменные среды Linux

Переменная	Описание
<b>PATH</b>	В этой переменной содержится список каталогов, разделенных двоеточиями, в которых система ищет исполняемые файлы. Если вы вводите имя исполняемого файла например, <b>ls</b> , <b>rc-update</b> или <b>emerge</b> ), который не находится ни в одной из перечисленных здесь каталогов, этот файл не запустится (если, конечно, вы не указали полный путь, например, <b>/bin/ls</b> ).
<b>ROOTPATH</b>	У этой переменной такое же значение, что и у <b>PATH</b> , но в ней перечисляются только те каталоги, которые нужно просматривать при вводе команды пользователем с правами <b>root</b> .
<b>LDPATH</b>	В этой переменной содержится список каталогов, разделенных двоеточиями, в которых динамический компоновщик ищет библиотеки.
<b>MANPATH</b>	В этой переменной содержится список каталогов, разделенных двоеточиями, в которых команда <b>man</b> ищет страницы справки.
<b>INFODIR</b>	В этой переменной содержится список каталогов, разделенных двоеточиями, в которых команда <b>info</b> ищет <b>info</b> -страницы.
<b>PAGER</b>	В этой переменной содержится путь к программе, позволяющей постранично просматривать содержимое файлов, например <b>less</b> или <b>more</b> .
<b>EDITOR</b>	В этой переменной содержится путь к программе, используемой для изменения файлов, например <b>vi</b>

	или <b>nano</b> .
<b>KDEDIRS</b>	В этой переменной содержится список каталогов, разделенных двоеточиями, в которых находятся ресурсы KDE.
<b>CLASSPATH</b>	В этой переменной содержится список каталогов, разделенных двоеточиями, в которых находятся классы Java.
<b>CONFIG_PROTECT</b>	В этой переменной содержится список каталогов, защищаемых Portage при обновлении, разделенных пробелами.
<b>CONFIG_PROTECT_MASK</b>	В этой переменной содержится список каталогов, исключаемых из защиты Portage при обновлении, разделенных пробелами

Ниже представлен пример определения всех этих переменных:

**Листинг 1:** Пример определения

```
PATH="/bin:/usr/bin:/usr/local/bin:/opt/bin:/usr/games/bin"
ROOTPATH="/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin"
LDPATH="/lib:/usr/lib:/usr/local/lib:/usr/lib/gcc-lib/i686-pc-linux-gnu/3.2.3"
MANPATH="/usr/share/man:/usr/local/share/man"
INFODIR="/usr/share/info:/usr/local/share/info"
PAGER="/usr/bin/less"
EDITOR="/usr/bin/vim"
KDEDIRS="/usr"
CLASSPATH="/opt/blackdown-jre-1.4.1/lib/rt.jar:."
CONFIG_PROTECT="/usr/X11R6/lib/X11/xkb /opt/tomcat/conf \
    /usr/kde/3.1/share/config /usr/share/texmf/tex/generic/config/ \
    /usr/share/texmf/tex/platex/config/ /usr/share/config"
CONFIG_PROTECT_MASK="/etc/gconf"
```

## 5.b. Глобальное определение переменных

### Каталог `/etc/env.d`

Для того, чтобы определить эти переменные централизованно, в Gentoo появился каталог `/etc/env.d`. В нём находится ряд файлов, например, **00basic**, **05gcc** и так далее, в которых определяются переменные, необходимые программам, указанным в названии файлов.

Например, при установке **gcc ebuild** создает файл **05gcc**, содержащий следующие определения переменных:

**Листинг 2:** `/etc/env.d/05gcc`

```
PATH="/usr/i686-pc-linux-gnu/gcc-bin/3.2"
ROOTPATH="/usr/i686-pc-linux-gnu/gcc-bin/3.2"
MANPATH="/usr/share/gcc-data/i686-pc-linux-gnu/3.2/man"
INFOPATH="/usr/share/gcc-data/i686-pc-linux-gnu/3.2/info"
CC="gcc"
CXX="g++"
```



```
LDPATH="/usr/lib/gcc-lib/i686-pc-linux-gnu/3.2.3"
```

В других дистрибутивах вам предлагается изменять или добавлять определения переменных среды в **/etc/profile** или где-нибудь еще. Gentoo, с другой стороны, облегчает вам (и Portage) поддержку и управление переменными среды, избавляя от необходимости уделять внимание многочисленным файлам, содержащим определения переменных.

Например, когда обновляется **gcc**, также без малейшего участия пользователя обновляется и **/etc/env.d/05gcc**.

От этого выигрывает не только Portage, но и вы, пользователь. Иногда от вас может потребоваться глобальная установка какой-нибудь переменной. Возьмем, к примеру, переменную **http\_proxy**. Вместо того, чтобы возиться с **/etc/profile**, теперь можно просто создать файл (**/etc/env.d/99local**) и добавить нужные определения туда:

**Листинг 3:** **/etc/env.d/99local**  
`http_proxy="proxy.server.com:8080"`

Используя один и тот же файл для всех своих переменных, вы можете быстро увидеть все определенные вами переменные вместе.

## Сценарий **env-update**

Переменная **PATH** определяется в нескольких файлах в **/etc/env.d**. Нет, нет это не ошибка: при запуске **env-update** различные определения объединяются перед обновлением переменных среды, позволяя пакетам (или пользователям) добавлять собственные значения переменных, не влияя на уже существующие.

Сценарий **env-update** объединяет значения переменных из файлов, находящихся в **/etc/env.d**, в алфавитном порядке. Имена файлов должны начинаться с двух десятичных цифр.

**Листинг 4:** Порядок обновления, используемый **env-update**

```
00basic    99kde-env    99local
+-----+-----+-----+
PATH="/bin:/usr/bin:/usr/kde/3.2/bin:/usr/local/bin"
```

**Внимание:** объединение выполняется не всегда, а только для следующих переменных: **KDEDIRS**, **PATH**, **CLASSPATH**, **LDPATH**, **MANPATH**, **INFODIR**, **INFOPATH**, **ROOTPATH**, **CONFIG\_PROTECT**, **CONFIG\_PROTECT\_MASK**, **PRELINK\_PATH** и **PRELINK\_PATH\_MASK**. Для всех остальных переменных используется значение, определенное в последнем из файлов (по алфавиту в каталоге **/etc/env.d**).

При запуске сценария **env-update** создаются все переменные среды, и помещаются в **/etc/profile.env** (используемый файлом **/etc/profile**). Кроме того, на основе значения **LDPATH** создается **/etc/ld.so.conf**. После этого запускается **ldconfig**, чтобы вновь создать файла **/etc/ld.so.cache**, используемый динамическим компоновщиком.

Если вы хотите, чтобы результаты работы **env-update** вступили в силу немедленно,

для обновления среды выполните следующую команду. Пользователи, самостоятельно устанавливавшие Gentoo, возможно, помнят ее из указаний по установке:

**Листинг 5:** Обновление среды  
**# env-update && source /etc/profile**

**Примечание:** Эта команда обновляет переменные только в текущем терминале, в новых консолях и их потомках. То есть, если вы работаете в X11, потребуется или набирать **source /etc/profile** в каждом открываемом терминале, или перезапустить X, чтобы все новые терминалы обращались к новым переменным. Если вы используете диспетчер входа в систему, станьте пользователем с правами **root** и наберите **/etc/init.d/xdm restart**. Если нет, вам придется выйти и снова войти в систему, чтобы X порождала потомков, использующих новые значения переменных.

## 5.с. Локальное определение переменных

### Пользовательские переменные

Далеко не всегда нужно определять переменные глобально. Например, вам может понадобиться добавить **/home/my\_user/bin** и текущий рабочий каталог (где вы находитесь) к переменной **PATH**, но при этом не нужно, чтобы это добавление появилось и в переменной **PATH** у всех остальных пользователей. Если вы хотите определить переменную среды локально, используйте **~/.bashrc** или **~/.bash\_profile**:

**Листинг 6:** Расширение **PATH** в **~/.bashrc** для локальных нужд  
(двоеточие без последующего указания каталога означает текущий рабочий каталог)  
**PATH="\${PATH}:/home/my\_user/bin:"**

Обновление вашей переменной **PATH** произойдет, когда вы выйдете и снова войдете в систему.

### Сеансовые переменные

Иногда нужны еще более жесткие ограничения. Вам может потребоваться возможность запуска исполняемых файлов из специально созданного временного каталога без указания полного пути к ним, и без изменения файла **~/.bashrc** ради нескольких минут.

В этом случае можно просто определить переменную **PATH** для текущего сеанса командой **export**. Переменной будет присвоено временное значение до тех пор, пока вы не завершите сеанс.

**Листинг 7:** Определение сеансовой переменной среды  
**# export PATH="\${PATH}:/home/my\_user/tmp/usr/bin"**

## С. Работа с Portage

### 1. Файлы и каталоги

#### 1.а. Файлы Portage

##### Директивы настройки

Настройки Portage по умолчанию хранятся в **/etc/make.globals**. Когда вы откроете этот файл, вы увидите, что все настройки представляют собой переменные. Что означает каждая из переменных, описано ниже.

Так как многие директивы отличаются в зависимости от используемой архитектуры, к Portage прилагаются настройки по умолчанию, которые входят в ваш профиль. На ваш профиль указывает символическая ссылка **/etc/make.profile**. Настройка Portage выполняется с помощью файлов **make.defaults** вашего профиля и всех родительских профилей. Более подробно о профилях и каталоге **/etc/make.profile** мы расскажем позже.

Если вы планируете вносить изменения в конфигурационные переменные, не изменяйте **/etc/make.globals** или **make.defaults**. Вместо этого пользуйтесь файлом **/etc/make.conf**, который имеет приоритет перед вышеуказанными файлами. Вы также обнаружите файл **/etc/make.conf.example**. Как понятно из его названия, это просто пример — Portage не использует этот файл.

Переменные Portage также можно устанавливать как переменные среды, но мы не рекомендуем этого делать.

##### Конфигурация, определяемая профилем

Мы уже встречались с каталогом **/etc/make.profile**. На самом деле это не каталог, а символическая ссылка на профиль, по умолчанию на тот, что содержится в **/usr/portage/profiles**, однако вы можете создавать свои собственные профили где угодно и ссылаться на них. Профиль, указанный ссылкой, является профилем, к которому принадлежит ваша система.

В профиле содержатся сведения для Portage, специфичные для архитектуры, такие как список пакетов, принадлежащих соответствующей системе, список неработоспособных (или замаскированных) пакетов, и т.д.

##### Конфигурация, задаваемая пользователем

Если вам необходимо изменить поведение Portage относительно установки программного обеспечения, вам потребуется отредактировать файлы, находящиеся в **/etc/portage**. Мы настоятельно рекомендуем вам пользоваться файлами из **/etc/portage**, всеми силами отговариваем от настройки поведения Portage через переменные среды!

Внутри **/etc/portage** вы можете создать следующие файлы:

- **package.mask**, в котором перечислены пакеты, которые Portage никогда не следует устанавливать;

- **package.unmask**, со списком пакетов, для которых вы хотите иметь возможность установки, даже если разработчики Gentoo отговаривают вас от этого;
- **package.keywords**, где перечислены пакеты, которые должны быть доступны для установки, несмотря на то, что они не подходят для вашей системы или архитектуры (пока);
- **package.use**, где перечислены значения USE-флагов, которые необходимо указывать для конкретных пакетов, а не для всей системы.

Дополнительные сведения о каталоге **/etc/portage**, а также список всех файлов, которые там можно создавать, находятся на справочной странице Portage:

**Листинг 1:** Вызов справки по Portage  
\$ man portage

## Изменение файлов Portage и размещения каталогов

Ранее упомянутые конфигурационные файлы нельзя хранить где угодно — Portage всегда ищет свои настроечные файлы в строго определенных местах. Однако Portage также использует множество каталогов для других целей: каталог для сборки, место для хранения исходных кодов, место для дерева Portage, и т.д.

Для этих целей существуют хорошо известные каталоги по умолчанию, положение которых можно изменить на свой вкус, внося изменения в **/etc/make.conf**. Оставшаяся часть этой главы посвящена описанию того, какие специальные места Portage использует для своих целей, и как изменить их расположение в файловой системе.

Этот документ не претендует на статус справочника. Если вам необходим полный объем информации, пожалуйста, обратитесь к страницам справки по Portage и **make.conf**:

**Листинг 2:** Вызов справки по Portage и **make.conf**  
\$ man portage  
\$ man make.conf

## 1.b. Хранение файлов

### Дерево Portage

Дерево Portage размещается, по умолчанию, в **/usr/portage**. Это определяется значением переменной **PORTDIR**. Когда вы храните дерево Portage где-либо в другом месте (изменив эту переменную), не забывайте соответственно изменить символическую ссылку **/etc/make.profile**.

Если вы измените переменную **PORTDIR**, вам может потребоваться изменить и следующие переменные: **PKGDIR**, **DISTDIR**, **RPMDIR**, так как они не замечают изменений **PORTDIR**. Это связано с особенностями их обработки Portage.

### Двоичные пакеты

Несмотря на то, что Portage по умолчанию не использует прекомпилированное программное обеспечение, для него предусмотрена очень мощная поддержка. Если вы укажете Portage работать с прекомпилированными пакетами, они будут разыскиваться в **/usr/portage/packages**. Это расположение определяется переменной **PKGDIR**.

### Исходные коды

Исходные коды приложений хранятся в **/usr/portage/distfiles** по умолчанию. Это определяется переменной **DISTDIR**.

### Файлы RPM

Несмотря на то, что Portage не может использовать RPM-файлы, есть возможность их создания командой **ebuild** (см. Приложение Ebuild). По умолчанию Portage хранит RPM файлы в каталоге **/usr/portage/rpm**, как определяется переменной **RPMDIR**.

### База данных Portage

Portage хранит состояние вашей системы (какие пакеты установлены, какие файлы относятся к определенным пакетам и т. п.) в **/var/db/pkg**. Не изменяйте эти файлы вручную! Это может разрушить знание вашей системы Portage.

### Кэш Portage

Кэш Portage (включая сведения о времени изменения, виртуальные пакеты, информацию дерева зависимостей и т. д.) хранится в **/var/cache/edb**. Это место действительно является кэшем: вы можете его очистить в любой момент, когда не запущены приложения, связанные с Portage.

## 1.с. Сборка программного обеспечения

### Временные файлы Portage

По умолчанию Portage хранит временные файлы в **/var/tmp**. За это отвечает переменная **PORTAGE\_TMPDIR**.

Если вы измените переменную **PORTAGE\_TMPDIR**, вам может потребоваться изменить и переменную **BUILD\_PREFIX**, так как она не замечает изменений **PORTAGE\_TMPDIR**. Это связано с особенностями ее обработки Portage.

### Каталог сборки

Portage создает специфичные каталоги сборки для каждого пакета внутри **/var/tmp/portage**. Это расположение задается переменной **BUILD\_PREFIX**.

### Размещение «живой файловой системы»

По умолчанию Portage устанавливает все файлы в текущую файловую систему (/), но это можно изменить, установив переменную окружения **ROOT**. Это может оказаться полезным при построении новых образов системы.

## 1.d. Ведение журнала

### Журнал Ebuild

Portage может создавать отдельные файлы журнала для каждого файла **ebuild**, но только тогда, когда переменная **PORT\_LOGDIR** указывает на место, доступное для записи для Portage (пользователя portage). По умолчанию эта переменная не установлена.

## 2. Настройка с помощью переменных

### 2.a. Настройка Portage

Как отмечалось ранее, Portage настраивается с помощью множества переменных, которые задаются в файле **/etc/make.conf**. За более полной и подробной информацией обращайтесь к странице справки по **make.conf**:

**Листинг 1:** Чтение страницы справки по make.conf  
**\$ man make.conf**

### 2.b. Параметры сборки

#### Параметры конфигурирования и компиляции

Когда Portage собирает приложения, компилятору и сценарию конфигурации передаются значения следующих переменных:

- **CFLAGS** и **CXXFLAGS** определяют желаемые флаги компилятора для C и C++;
- **CHOST** определяет информацию об используемой платформе для сценария конфигурации приложения;
- **MAKEOPTS** передается команде make и обычно применяется для установки степени распараллеливания компиляции. Более подробная информация о параметрах команды make находится на странице справки по make.

Переменная USE также используется при конфигурировании и компиляции, но о ней уже много и подробно говорилось в предыдущих главах.

#### Параметры установки

Когда Portage устанавливает (**merge**) новую версию программного продукта, файлы более старых версий удаляются из системы. Portage дает пользователю 5-ти секундную задержку перед стиранием старых версий. Эти 5 секунд задаются переменной **CLEAN\_DELAY**.

## 2.c. Защита конфигурационных файлов

### Места, защищаемые Portage

Portage записывает файлы, предоставляемые новой версией программы, поверх старых, если только эти файлы не расположены в защищенном месте. Защищенные каталоги определяются переменной **CONFIG\_PROTECT**. Обычно, это места расположения файлов конфигурации. Каталоги в списке разделяются пробелами.

Файл, который должен быть записан в такой защищенный каталог, переименовывается, а пользователь получает предупреждение о наличии новой версии (обычно) файла конфигурации.

Узнать текущее значение **CONFIG\_PROTECT** можно из сообщений emerge --info:

**Листинг 2:** Получение значения CONFIG\_PROTECT

```
$ emerge --info | grep 'CONFIG_PROTECT='
```

Более подробная информация о защите конфигурационных файлов, осуществляемой системой Portage, доступна по команде emerge:

**Листинг 3:** Подробная информация о защите конфигурационных файлов

```
$ emerge --help config
```

### Исключение каталогов

Чтобы снять защиту с определенных подкаталогов защищенного каталога, можно использовать переменную **CONFIG\_PROTECT\_MASK**.

## 2.d. Параметры скачивания

### Расположение сервера

Если запрошенная информация или данные отсутствуют в вашей системе, Portage обращается за ними в интернет. Расположение серверов для различных каналов получения информации задается следующими переменными:

- **GENTOO\_MIRRORS** определяет список адресов серверов, содержащих исходный код (distfiles);
- **PORTAGE\_BINHOST** указывает расположение определенного сервера, содержащего двоичные пакеты (prebuilt packages) для вашей системы;
- Третья переменная содержит расположение сервера **rsync**, который используется при обновлении вашего дерева портежей:  
**SYNC** указывает сервер, с которого Portage извлекает дерево портежей.

Переменные **GENTOO\_MIRRORS** и **SYNC** можно установить автоматически программой **mirrorselect**. Перед тем, как использовать, ее нужно установить, выполнив **emerge mirrorselect**. За дополнительной информацией обращайтесь к оперативной справке **mirrorselect**:

Если вы вынуждены использовать прокси-сервер, для его указания можно использовать переменные HTTP\_PROXY, FTP\_PROXY и RSYNC\_PROXY.

### Команды для извлечения

Когда Portage требуется извлечь исходный код, по умолчанию используется **wget**. Вы можете это изменить с помощью переменной **FETCHCOMMAND**.

Portage может возобновлять скачивание частично загруженного исходного кода. По умолчанию используется **wget**, но это можно переопределить переменной **RESUMECOMMAND**.

Удостоверьтесь, что ваши команды **FETCHCOMMAND** и **RESUMECOMMAND** сохраняют исходный код в нужном месте. Внутри этих переменных следует использовать `\${URI}` и `\${DISTDIR}`, для указания расположения исходных кодов и distfiles, соответственно.

Также существует возможность определить индивидуальные настройки для различных протоколов, используя **FETCHCOMMAND\_HTTP**, **FETCHCOMMAND\_FTP**, **RESUMECOMMAND\_HTTP**, **RESUMECOMMAND\_FTP**, и т.п.

### Настройки rsync

Вы не можете заменить команду rsync, которую Portage использует для обновления дерева портежей, но можно установить несколько переменных, определяющих ее поведение:

- **RSYNC\_EXCLUDEFROM** указывает на файл, где перечислены пакеты и/или категории, которые rsync должна игнорировать во время обновления.
- **RSYNC\_RETRIES** определяет, сколько раз rsync должна пытаться соединиться с зеркалом, на которое указывает переменная SYNC. По умолчанию равна 3.
- **RSYNC\_TIMEOUT** определяет количество секунд, в течение которого rsync соединение может бездействовать, перед тем как rsync сочтет его превысившим время ожидания. По умолчанию равна 180, но если вы используете соединение по модему или у вас медленный компьютер, возможно, следует установить значение этой переменной равным 300 или большим.

## 2.e. Настройка Gentoo

### Выбор ветви

Используемую ветвь можно изменить переменной **ACCEPT\_KEYWORDS**. По умолчанию используется стабильная ветвь для вашей архитектуры. Дополнительная информация о ветвях Gentoo находится в следующей главе.

### Возможности Portage



Вы можете включить отдельные функции Portage с помощью переменной **FEATURES**. Возможности Portage рассматривались в предыдущих главах.

## 2.f. Поведение Portage

### Распределение ресурсов

С помощью переменной **PORTAGE\_NICENESS** можно увеличивать или уменьшать значение **nice**, с которым выполняется Portage. Значение **PORTAGE\_NICENESS** прибавляется к текущему значению **nice**.

Более подробно о значениях **nice** написано в странице справки:

**Листинг 5:** Дополнительные сведения о **nice**  
`$ man nice`

### Настройки вывода

Переменная **NOCOLOR** (по умолчанию «false») определяет, следует ли Portage отключить цветовую раскраску своих сообщений.

## 3. Смешение ветвей программного обеспечения

### 3.a. Использование одной ветви

#### Стабильная ветвь

Переменная **ACCEPT\_KEYWORDS** определяет, какую из ветвей использовать в вашей системе. По умолчанию используется стабильная ветвь для вашей архитектуры, например **x86**

Мы рекомендуем использовать только стабильную ветвь. Однако, если для вас стабильность не критична и вы хотите помочь Gentoo, отсылая отчеты об ошибках на <http://bugs.gentoo.org>, читайте дальше.

#### Тестовая ветвь

Если вы желаете использовать наиболее свежее ПО, подумайте над использованием тестовой ветви. Чтобы Portage начала использовать тестовую ветвь, добавьте «~» перед названием вашей архитектуры.

Тестовая ветвь полностью соответствует своему названию: для тестирования. Если пакет находится в стадии тестирования, это означает, что разработчики считают, что пакет работоспособен, но тщательно он не протестирован. Вы можете оказаться первым, кто столкнется с какой-либо ошибкой. В этом случае вы можете создать отчет об ошибке, чтобы разработчики узнали о ней.

Однако будьте готовы к тому, что могут возникнуть проблемы со стабильностью, неудовлетворительной поддержкой пакетов (например

неправильные/отсутствующие зависимости), слишком частыми обновлениями (а в результате — частыми сборками) или невозможностью собрать пакет. Если вы не знаете, как работает Gentoo и как разрешать возникающие проблемы, мы рекомендуем не отходить от стабильной и оттестированной ветви.

К примеру, для выбора тестовой ветви на архитектуре x86, отредактируйте `/etc/make.conf` и укажите в нем:

**Листинг 1:** Установка значения переменной `ACCEPT_KEYWORDS`  
`ACCEPT_KEYWORDS="~x86"`

Если вы запустите обновление системы, то увидите, что многие пакеты нуждаются в обновлении. Обратите внимание, что после перехода на тестовую ветвь и обновления системы, как правило, нет простого пути назад к стабильной официальной ветви (конечно, кроме использования резервной копии).

### 3.b. Одновременное использование стабильной и тестовой ветвей

#### Местоположение `package.keywords`

Вы можете указать, чтобы Portage использовала тестовую ветвь только для определенных пакетов, а для остальной системы — стабильную ветвь. Для этого добавьте категорию и имя пакета, для которого вы желаете использовать тестовую ветвь, в файл `/etc/portage/package.keywords`. Вместо этого можно создать каталог (с таким же именем) и указывать пакеты в файлах, находящихся внутри этого каталога. Например, для использования тестовой ветви для **gnumeric**:

**Листинг 2:** Настройка `/etc/portage/package.keywords` для `gnumeric`, вся строка  
`app-office/gnumeric ~x86`

#### Тестирование определенных версий

Если вы желаете использовать конкретную версию ПО из тестовой ветви, но не хотите, чтобы Portage использовала тестовую ветвь для последующих версий этого ПО, можно указать в местоположении `package.keywords` номер необходимой версии. В этом случае вы обязаны использовать оператор `=`. Также можно указать диапазон версий, используя операторы `<=`, `<`, `>` или `>=`.

В любом случае, добавляя информацию о версии, вы должны использовать один из этих операторов. Если вы не указываете версию, эти операторы использовать нельзя.

В следующем примере мы просим Portage разрешить установку `gnumeric-1.2.13`:

**Листинг 3:** Использование конкретной тестовой версии `gnumeric`  
`=app-office/gnumeric-1.2.13 ~x86`

### 3.c. Использование заблокированных пакетов

## Расположение `package.unmask`

Разработчики Gentoo не поддерживают использование этого места расположения. Пожалуйста, используйте их на свой страх и риск. Просьбы о помощи, связанные с использованием **`package.unmask`** и/или **`package.mask`**, останутся без ответа. Вы предупреждены.

Если использование пакета было заблокировано разработчиками Gentoo, но вы желаете его использовать несмотря на причины блокировки, указанные в файле `package.mask` (по умолчанию он находится в `/usr/portage/profiles`), добавьте для него точно такую же строку в файл `/etc/portage/package.unmask` (или в файл в этом каталоге, если это каталог).

Например, если `=net-mail/hotwayd-0.8` заблокирован, то разблокировать его можно, прописав в `package.unmask` точно такую же строчку:

**Листинг 4:** `/etc/portage/package.unmask`  
`=net-mail/hotwayd-0.8`

## Местоположение `package.mask`

Если вы не хотите, чтобы Portage использовала какое-то конкретное ПО или конкретные версии ПО, вы можете его самостоятельно заблокировать, добавив соответствующую запись в `/etc/portage/package.mask` (в такой файл либо в файл внутри такого каталога).

Если, к примеру, вы не хотите, чтобы Portage устанавливала исходные коды ядра новее, чем `gentoo-sources-2.6.8.1`, добавьте такую строку в местоположение `package.mask`:

**Листинг 5:** Пример использования файла `/etc/portage/package.mask`  
`>sys-kernel/gentoo-sources-2.6.8.1`

## 4. Дополнительные средства Portage

### 4.a. `etc-update`

**`etc-update`** — это утилита, предназначенная для обновления в системе файлов `._cfg0000_<имя>`. Она обеспечивает интерактивную настройку установки и может также автоматически устанавливать тривиальные изменения. Файлы создаются `._cfg0000_<имя>` Portage, когда нужно заменить файл в каталоге, защищенном переменной **`CONFIG_PROTECT`**.

Выполнить **`etc-update`** довольно просто:

**Листинг 1:** Запуск `etc-update`  
`# etc-update`

После выполнения тривиальных обновлений, вы увидите запрос со списком защищенных файлов, ожидающих обновления. Внизу вам предложат следующие

варианты:

**Листинг 2:** Запрос etc-update

Please select a file to edit by entering the corresponding number.

(-1 to exit) (-3 to auto merge all remaining files)

(-5 to auto-merge AND not use 'mv -i'):

(Пожалуйста, выберите файл для правки, введя соответствующее число.

(-1 - выход) (-3 - автоустановка всех оставшихся файлов)

(-5 для автоустановки БЕЗ использования 'mv -i'): )

При вводе -1, etc-update выходит, прекращая последующие изменения. Если вы введете -3 или -5, все перечисленные файлы конфигурации заменяются более новыми версиями. Следовательно, очень важно сначала отобрать файлы, которые не следует автоматически обновлять. Для этого надо только вводить номер, указанный слева от файлов.

Например, выбираем файл конфигурации /etc/pear.conf:

**Листинг 3:** Обновление конкретного конфигурационного файла

Beginning of differences between /etc/pear.conf and /etc/.\_cfg0000\_pear.conf

[...]

End of differences between /etc/pear.conf and /etc/.\_cfg0000\_pear.conf

1) Replace original with update

2) Delete update, keeping original as is

3) Interactively merge original with update

4) Show differences again

Теперь можно увидеть различия между двумя файлами. Если вы считаете, что обновленный файл конфигурации можно использовать без проблем, введите 1. Если вы считаете, что обновленный файл конфигурации не нужен, или не содержит новую или полезную информацию, введите 2. Если вы хотите обновить текущий файл в интерактивном режиме, введите 3.

Нет никакого смысла в подробном описании интерактивного обновления. Для полноты изложения, мы перечислим возможные команды, которые можно использовать при интерактивном слиянии двух файлов. Вас встречают две строки (одна исходная, вторая измененная) и запрос, в ответ на который можно ввести одну из следующих команд:

**Листинг 4:** Команды, доступные при интерактивном слиянии

ed: редактировать и использовать оба варианта, каждый пометить заголовком

eb: редактировать и использовать оба варианта

el: редактировать и использовать левый вариант

er: редактировать и использовать правый вариант

e: редактировать новую версию

l: использовать левую версию

r: использовать правую версию

s: молча включить общие строки

v: включить общие строки, сообщив подробности

q: выход

Завершив обновление важных файлов конфигурации, вы можете автоматически обновить оставшиеся файлы конфигурации. etc-update выйдет, если не найдет других файлов, подлежащих обновлению.

## 4.b. dispatch-conf

С помощью **dispatch-conf** можно обновлять файлы конфигурации, сохраняя при этом историю изменений. **dispatch-conf** хранит различия между файлами конфигурации в виде заплаток или в системе управления версиями RCS.

Как и с **etc-update**, вы можете попросить сохранить файл конфигурации как есть, использовать новый файл конфигурации, редактировать текущий или объединить изменения интерактивно. Однако, у **dispatch-conf** также есть приятные дополнительные возможности:

- автоматическое обновление файлов, в которых обновились только комментарии;
- автоматическое обновление файлов, которые отличаются только количеством пробелов.

Убедитесь, что вы сначала отредактировали **/etc/dispatch-conf.conf** и создали каталог, прописанный в **archive-dir**.

За дополнительными сведениями обращайтесь к странице справки **dispatch-conf**:

**Листинг 5:** Чтение справки по **dispatch-conf**  
**\$ man dispatch-conf**

## 4.c. quickpkg

С **quickpkg** вы можете создавать архивы пакетов, уже установленных в системе. Эти архивы можно использовать в качестве двоичных пакетов. Запуск **quickpkg** прост: только укажите имена пакетов, которые нужно заархивировать.

Например, чтобы поместить в архив **curl**, **arts** и **procps**:

**Листинг 6:** Пример использования **quickpkg**  
**# quickpkg curl arts procps**

Двоичные пакеты будут храниться в **\$PKGDIR/All** (по умолчанию — **/usr/portage/packages/All**). Символьные ссылки, указывающие на эти пакеты, помещаются в **\$PKGDIR/<категория>**.

## 5. Отступление от официального дерева

### 5.a. Использование собственного дерева Portage

#### Исключение пакета/категории

Вы можете выборочно обновлять определенные категории/пакеты, игнорируя обновление других категорий/пакетов. Это достигается путем исключения таких категорий/пакетов программой **rsync** на этапе выполнения **emerge --sync**.

Вам потребуется определить имя файла, содержащего шаблоны исключаемых пакетов, в переменной **RSYNC\_EXCLUDEFROM** в своем файле `/etc/make.conf`.

**Листинг 1:** Указание файла исключаемых пакетов в `/etc/make.conf`  
**RSYNC\_EXCLUDEFROM=/etc/portage/rsync\_excludes**

**Листинг 2:** Исключение всех игр в файле `/etc/portage/rsync_excludes`  
**games-\*/\***

Заметьте, однако, что это может привести к проблемам с зависимостями, так как новые разрешенные пакеты могут зависеть от других новых, но исключенных из обновления пакетов.

## 5.b. Добавление неофициального сборочного файла ebuild

### Определение оверлейного каталога портежей

Вы можете указать Portage использовать сборочные файлы, не входящие в официальное дерево Portage. Создайте новый каталог (к примеру, `/usr/local/portage`), в котором будут находиться файлы **ebuild** сторонних разработчиков. Используйте в точности такую же структуру каталогов, как и в официальном дереве портежей!

Затем определите переменную **PORTDIR\_OVERLAY** в `/etc/make.conf`, чтобы она указывала на ранее созданный каталог. Теперь при использовании Portage, эти сборочные файлы будут рассматриваться как часть системы, и не будут удаляться/перезаписываться при последующих запусках **emerge --sync**.

### Работа с несколькими оверлейными каталогами

Для продвинутых пользователей, ведущих разработку в нескольких оверлейных каталогах, тестирующих пакеты перед включением в основное дерево портежей или просто желающих использовать неофициальные сборочные файлы ebuild из разных источников, в пакете `app-portage/gentoolkit-dev` есть утилита `gensync`, которая поможет поддерживать ваши оверлейные репозитории в актуальном состоянии.

Используя **gensync**, вы можете обновить сразу все репозитории или выбрать для обновления только некоторые из них. В каждом репозитории в каталоге `/etc/gensync/` должен находиться файл **.syncsource**, в котором содержится информация о местоположении репозитория, его имени, идентификаторе и т.д.

Предположим, что у вас есть два дополнительных репозитория с названиями `java` (для сборочных файлов разработок, ведущихся на `java`) и `entapps` (для внутренних приложений, разработанных на вашем предприятии). Вы можете обновить эти репозитории следующей командой:

**Листинг 3:** Запуск `gensync` для обновления нескольких репозиториях  
**# gensync java entapps**

## 5.с. Программы, поддерживаемые не Portage

### Использование Portage с пакетами самостоятельной сборки

Иногда вам может потребоваться сконфигурировать, установить и поддерживать программное обеспечение самостоятельно, без автоматизации со стороны Portage, не смотря на то, что оно поддерживается Portage. Наиболее известные случаи — это исходные коды ядра и драйверы от nVidia. Вы можете настроить Portage так, чтобы системе стало известно, что определенные пакеты установлены вручную. Этот процесс называется внедрение, и поддерживается Portage посредством файла `/etc/portage/profile/package.provided`.

Например, если вы захотите сообщить Portage, что пакет `vanilla-sources-2.6.11.6` установлен вручную, нужно добавить следующую строку в `/etc/portage/profile/package.provided`:

Листинг 4: Пример строки из файла `package.provided`  
`sys-kernel/vanilla-sources-2.6.11.6`

## 6. Использование *ebuild*

### 6.a. Emerge и Ebuild

Программа **ebuild** — это низкоуровневый интерфейс системы Portage. С ее помощью можно выполнять определенные действия над заданными сборками **ebuild**. Например, вы можете самостоятельно выполнить отдельные этапы установки.

Программа **ebuild** предназначена в основном для разработчиков, поэтому более подробная информация находится в настольной книге разработчика (англ.). Однако, мы расскажем, какие экземпляры **ebuild** вызываются системой Portage на разных этапах установки, и как выполнить пост-конфигурационные шаги, которые допускаются некоторыми пакетами.

### 6.b. Ручная установка программ

#### Извлечение исходных кодов и проверка контрольных сумм

Каждый раз, когда вы вызываете **ebuild** для какого-то **ebuild**-файла, проверяется совпадение контрольной суммы всех задействованных файлов с указанной в файлах Manifest или `files/digest-<имя>-<версия>`. Проверка выполняется после загрузки исходных кодов.

Чтобы загрузить исходные коды с помощью **ebuild**, запустите:

Листинг 1: Загрузка исходных кодов  
`# ebuild путь/к/файлу-ebuild fetch`

Если контрольная сумма md5 сборочного файла не совпадает с той, что указана в файле **Manifest**, или же один из загруженных файлов не совпадает с описанием в файле `files/digest<пакет>`, вы получите сообщение об ошибке, похожее на такое:

```
Листинг 2: Ошибка контрольной суммы ebuild
!!! File is corrupt or incomplete. (Digests do not match)
>>> our recorded digest: db20421ce35e8e54346e3ef19e60e4ee
>>> your file's digest: f10392b7c0b2bbc463ad09642606a7d6
```

(!!! Файл поврежден или усечен. (Контрольные суммы не совпадают) )

На следующей строке указывается проблемный файл.

Если вы абсолютно уверены, что загруженные исходные коды и сам сборочный файл ebuild именно те, что вам нужны, можете пересоздать файлы Manifest и digest-<пакете>, используя функцию digest программы ebuild:

```
Листинг 3: Создание новых файлов Manifest и digest
# ebuild путь/к/файлу-ebuild digest
```

## Распаковка исходных кодов

Чтобы распаковать исходные коды в `/var/tmp/portage` (или любой другой каталог, указанный в `/etc/make.conf`), запустите функцию unpack программы ebuild:

```
Листинг 4: Распаковка исходных кодов
# ebuild путь/к/файлу-ebuild unpack
```

Эта команда выполнит функцию `src_unpack()` программы ebuild (которая по умолчанию просто выполняет распаковку, если функция `src_unpack()` не определена). Все необходимые заплатки накладываются также на этом этапе.

## Компиляция исходных кодов

Следующий шаг в процессе установки — компиляция исходных кодов. Для этого выполняется функция `src_compile()` вашего сборочного файла. Если нужно, заодно выполняется конфигурация.

```
Листинг 5: Компиляция исходных кодов
# ebuild путь/к/файлу-ebuild compile
```

Если вы хотите изменить инструкции компиляции, советуем отредактировать функцию `src_compile()`. Однако, вы можете также обмануть Portage, заставив ее поверить, что программа ebuild уже завершила компиляцию. Запустите нужные команды самостоятельно и создайте пустой файл `.compile` в рабочем каталоге.

```
Листинг 6: Сообщение Portage о завершении задания компиляции
# touch .compiled
```

## Установка файлов во временное место



Следующий шаг — установка всех необходимых файлов во временный каталог. В него помещаются все файлы, подлежащие включению в рабочую файловую систему. Вы можете выполнить этот этап, запустив функцию установки программы **ebuild**, которая исполняет функцию `src_install()` сборочного файла.

**Листинг 7:** Установка файлов  
**# ebuild путь/к/файлу-ebuild install**

### Помещение файлов в рабочую файловую систему

Последний этап — перенос всех файлов в рабочую файловую систему и их регистрация в системе Portage. В **ebuild** этот этап называется «**qmerge**», и включает следующие действия:

- выполняется функция `pkg_preinst()`, если она определена;
- все файлы копируются в рабочую файловую систему;
- файлы регистрируются в системе Portage;
- выполняется функция `pkg_postinst()`, если она определена;

Запустите функцию **qmerge** программы **ebuild**, чтобы выполнить этот этап:

**Листинг 8:** Помещение файлов в рабочую файловую систему  
**# ebuild путь/к/файлу-ebuild qmerge**

### Очистка временного каталога

Наконец, можно очистить временный каталог, используя команду `clean` программы **ebuild**:

**Листинг 9:** Очистка временного каталога  
**# ebuild путь/к/файлу-ebuild clean**

## 6.с. Дополнительные возможности Ebuild

### Запуск всех команд установки

С помощью функции **merge** программы **ebuild**, можно запустить команды извлечения, распаковки, компиляции, установки и помещения за один раз:

**Листинг 10:** Установка программы  
**# ebuild путь/к/файлу-ebuild merge**

### Выполнение действий по настройке

В некоторых приложениях содержатся инструкции по дальнейшей настройке установленного пакета. Эти инструкции могут потребовать участия пользователя, и, следовательно, не выполняться автоматически. Для запуска шагов настройки, указанных в необязательной функции **config()** сборочного файла, используйте команду **config** программы **ebuild**:

**Листинг 11:** Настройка пакета  
**# ebuild путь/к/файлу-ebuild config**

## Сборка пакета (RPM)

Вы можете попросить Portage создать двоичный пакет или даже RPM из вашего сборочного файла, воспользовавшись командами `rpackage` и `rpm`, соответственно. Эти команды несколько различаются:

- команда `rpackage` во многом похожа на `merge`, выполняя все необходимые шаги (извлечение, распаковку, компиляцию, установку) перед созданием пакета;
- команда `rpm` собирает пакет RPM из файлов созданных после запуска окончания функции `install` программы `ebuild`.

**Листинг 12:** Создание пакетов  
(создание двоичного пакета, совместимого с Portage)  
**# ebuild путь/к/файлу-ebuild package**

(создание пакета RPM)  
**# ebuild путь/к/файлу-ebuild rpm**

Созданный RPM, однако, не будет содержать информацию о зависимостях из сборочного файла **ebuild**.

## 6.d. Дополнительная информация

За дополнительными сведениями о системе **Portage**, программе `ebuild` и сценариях `ebuild` обращайтесь к следующим страницам справки **man**:

**Листинг 13:** Страницы справки  
**\$ man portage** (сама система Portage)  
**\$ man emerge** (команда emerge)  
**\$ man ebuild** (команда ebuild)  
**\$ man 5 ebuild** (синтаксис файлов ebuild)

Кроме того, дополнительные сведения, относящиеся к разработке, находятся в настольной книге разработчика (англ.).

## D. Настройка сети в Gentoo

### 1. Начальная настройка

#### 1.a. Приступаем к настройке

**Примечание:** В документе предполагается, что вы правильно сконфигурировали свое ядро и модули для оборудования, и вам известно интерфейсное имя устройств. Мы также предполагаем, что вы настраиваете **eth0**, хотя на самом деле это может оказаться **eth1**, **wlan0** и т.д.

**Примечание:** Требуется, чтобы у вас использовался baselayout-1.11.11 или более свежий.

Для начала настройки своей сетевой платы, нужно рассказать о ней системе Gentoo RC. Это делается созданием символической ссылки с net.lo на net.eth0 в /etc/init.d.

**Листинг 1:** Создание символической ссылки с net.lo на net.eth0

```
# cd /etc/init.d
# ln -s net.lo net.eth0
```

Теперь система **Gentoo** RC знает об этом интерфейсе. Ей также нужно знать, как настраивать новый интерфейс. Конфигурация всех сетевых интерфейсов находится в **/etc/conf.d/net**. Вот простая настройка для использования DHCP или статического адреса.

**Листинг 2:** Примеры для /etc/conf.d/net

```
# использование DHCP
config_eth0=( "dhcp" )

# статический IP-адрес, используется запись CIDR
config_eth0=( "192.168.0.7/24" )
routes_eth0=( "default via 192.168.0.1" )

# статический IP-адрес, запись с маской подсети
config_eth0=( "192.168.0.7 netmask 255.255.255.0" )
routes_eth0=( "default gw 192.168.0.1" )
```

**Примечание:** Если конфигурация для интерфейса не указывается, предполагается использование DHCP.

**Примечание:** CIDR расшифровывается как Classless InterDomain Routing (бесклассовая междоменная маршрутизация). Первоначально, адреса IPv4 были разделены на классы А, В и С. Ранняя система классификации не была рассчитана на массовую популярность интернета, и попала под угрозу исчерпания новых уникальных адресов. CIDR — это схема адресации, позволяющая одному IP-адресу обозначать множество IP-адресов. IP-адрес CIDR выглядит как обычный IP-адрес с добавлением косой черты и числа; например, 192.168.0.0/16. CIDR описывается в RFC 1519.

Теперь, настроив интерфейс, мы можем запускать и останавливать его следующими командами:

**Листинг 3:** Сценарии запуска и остановки сети

```
# /etc/init.d/net.eth0 start
# /etc/init.d/net.eth0 stop
```

**Важно:** При поиске неисправностей сети рекомендуется установить RC\_VERBOSE="yes" в /etc/conf.d/rc для получения более подробной информации о происходящем.

Теперь, успешно запустив и остановив сетевой интерфейс, вы можете захотеть, чтобы он запускался при каждой загрузке Gentoo. Вот как это сделать. Последняя команда «rc» указывает Gentoo, что нужно запускать в текущем уровне запуска

любые еще не запущенные сценарии.

Листинг 4: Настройка запуска сетевого интерфейса при загрузке  
# rc-update add net.eth0 default  
# rc

## 2. Расширенная настройка

### 2.а. Расширенная настройка

Переменная **config\_eth0** служит основой конфигурации интерфейса. Она содержит список высокоуровневых инструкций по настройке интерфейса (в данном случае, eth0). Все команды списка выполняются последовательно. Интерфейс считается работоспособным, если хотя бы одна команда выполнена успешно.

Вот список встроенных инструкций:

Команда	Описание
<b>null</b>	Не выполнять никаких действий
<b>noop</b>	Если интерфейс включен и существует адрес, успешно завершить настройку.
<b>an IPv4 or IPv6 address</b>	Добавить адрес к интерфейсу
<b>dhcр, adsl or apipa(или команда запуска модуля стороннего изготовителя)</b>	Запустить модуль, реализующий команду. Например, dhcр запускает модуль, реализующий команду. Например, dhcр запускает модуль, реализующий DHCP, которым может быть dhcрcd, udhcрс, dhclient или rump.

На случай неудачного выполнения команды можно указать запасную команду. Запасной вариант должен строго соответствовать структуре конфигурации.

Команды можно сцеплять. Вот несколько практических примеров.

**Листинг 1:** Примеры настройки

# Задание трех адресов IPv4

```
config_eth0=(  
  "192.168.0.2/24"  
  "192.168.0.3/24"  
  "192.168.0.4/24"  
)
```

# Задание одного адреса IPv4 и двух адресов IPv6

```
config_eth0=(  
  "192.168.0.2/24"  
  "4321:0:1:2:3:4:567:89ab"  
  "4321:0:1:2:3:4:567:89ac"
```

```
)  
# Сохранять адрес, присвоенный ядром, до отключения интерфейса.  
# При этом назначить другой через DHCP. Если DHCP не работает,  
# задать статический адрес, определяемый APIPA  
config_eth0=(  
  "noop"  
  "dhcp"  
)  
fallback_eth0=(  
  "null"  
  "apiipa"  
)
```

**Примечание:** При использовании модуля `ifconfig` для назначения нескольких адресов, для каждого дополнительного адреса создаются псевдонимы интерфейса. Так, в двух примерах, приведенных выше, создаются интерфейсы `eth0`, `eth0:1` и `eth0:2`. С этими интерфейсами нельзя сделать ничего особенного, так как и ядро, и другие программы обрабатывают `eth0:1` и `eth0:2` просто как `eth0`.

**Важно:** Порядок настройки запасного режима имеет значение! Если бы мы не указали инструкцию `null`, то команда `apiipa` запускалась бы только при неудачном выполнении команды `noop`.

**Примечание:** APIPA и DHCP обсуждаются позже.

## 2.b. Сетевые зависимости

Сценарии инициализации в `/etc/init.d` могут находиться в зависимости от определенного сетевого интерфейса или просто от службы сети (`net`). Определив переменную `RC_NET_STRICT_CHECKING` в `/etc/conf.d/rc`, службе `net` можно придать различный смысл.

Значение	Описание
<b>none</b>	Служба <code>net</code> считается всегда работающей
<b>no</b>	В основном это означает, что по крайней мере одна служба <code>net.*</code> , кроме <code>net.lo</code> , должна работать. Это может пригодиться пользователям ноутбуков, у которых есть WIFI и статическое проводное подключение, когда нужно, чтобы при включении хотя бы одного интерфейса служба сети выглядела включенной.
<b>lo</b>	То же, что и <code>no</code> , но с учетом <code>net.lo</code> . Может быть полезно для тех, кого не волнует, чтобы определенный интерфейс включался при загрузке.
<b>yes</b>	В этом случае ВСЕ сетевые интерфейсы ДОЛЖНЫ работать, чтобы служба <code>net</code> считалась работающей.

Но как насчет `net.br0`, зависящего от `net.eth0` и `net.eth1`? `net.eth1` может быть беспроводным или PPP-устройством, требующим предварительной настройки для возможности включения в мост. Это невозможно сделать в `/etc/init.d/net.br0`, так как он является символьной ссылкой на `net.lo`.

Ответом является создание своей собственной функции `depend()` в `/etc/conf.d/net`.

**Листинг 2:** Зависимость `net.br0` в `/etc/conf.d/net`  
# Можно использовать любую зависимость (use, after, before),  
# как видно в текущих сценариях  
`depend_br0() {`  
    `need net.eth0 net.eth1`  
`}`

Более подробно зависимости обсуждаются в разделе Написание сценариев инициализации Настольной книги Gentoo.

## 2.c. Имена и значения переменных

Имена переменных являются динамическими. Обычно они следуют структуре `variable_${interface|mac|essid|apmac}`. Например, значение переменной `dhcpcd_eth0` хранит параметры `dhcpcd` для `eth0`, а переменной `dhcpcd_essid` — параметры `dhcpcd`, используемые при подключении любого интерфейса к ESSID «`essid`».

Однако, не существует твердого простого правила, устанавливающего, что интерфейсы должны называться `ethx`. На деле, имена многих беспроводных выглядят как `wlanx`, `gax` и `ethx`. Кроме того, некоторые пользовательские интерфейсы, например, мосты, можно называть как угодно, например, `foo`. Для

пущего разнообразия, в именах беспроводных точек доступа также допускаются знаки, не входящие в алфавитно-цифровые; это имеет значение, потому что есть возможность настройки сетевых параметров для отдельных ESSID.

Оборотная сторона всего этого в том, что для настройки сети в Gentoo используются переменные `bash`, а `bash` не в состоянии использовать что-либо кроме знаков английского алфавита и цифр. Чтобы обойти такое ограничение, мы заменяем каждый символ, не являющийся английским буквенно-цифровым, на знак подчеркивания: `_`.

Другая особенность `bash` — это значения переменных: некоторые символы требуют специальной записи, перед ними помещается знак `\`. Им необходимо предварять следующие символы: `"`, `'` и `\`.

В следующем примере мы используем беспроводные ESSID, так как в них может содержаться самое широкое множество символов. Мы воспользуемся ESSID `My "\ NET`:

**Листинг 3:** Пример имени переменной

```
# Этот пример работает, но домен не существует
dns_domain_My____NET="My \"\\ NET"
```

```
# Предыдущая строка устанавливает домен dns в My "\ NET при
# подключении беспроводной платы к точке доступа с ESSID My "\ NET.
```

### 3. Модульное построение сети

#### 3.а. Сетевые модули

Сейчас мы поддерживаем модульные сетевые сценарии. Это значит, что мы можем легко добавлять поддержку для новых типов интерфейсов и конфигурационных модулей, сохраняя совместимость с существующими.

По умолчанию, модули загружаются только если пакет, нужный им, установлен. Если отметить модуль, для которого не установлен пакет, будет выдана ошибка с указанием, какой пакет нужно установить. В идеале, настройка модулей требуется только тогда, когда установлено несколько пакетов, представляющих одну и ту же службу, а вам установить приоритет одного из них.

**Примечание:** Все обсуждаемые значения хранятся в `/etc/conf.d/net`, если явно не указано иное.

**Листинг 1:** Предпочтение модуля

```
# выбор iproute2, а не ifconfig
modules=( "iproute2" )
```

```
# можно также указать другие модули для отдельного интерфейса
# здесь мы выбираем udhcpc, а не dhcpcd
modules_eth0=( "udhcpc" )
```

```
# также можно указать, какие модули не надо использовать: например,
# возможно, вы используете supplicant или linux-wlan-ng для управления
# параметрами беспроводной сети, но при этом желаете настраивать сетевые
# параметры отдельно для каждого связанного ESSID
```

```
modules=( "liwconfig" )
```

### 3.b. Обработчики интерфейса

Сейчас мы предоставляем два обработчика интерфейса: **ifconfig** и **iproute2**. Для настройки сети вам нужен только один из них.

- **ifconfig** в текущем Gentoo используется по умолчанию, и включен в системный профиль.
- **iproute2** — более мощный и гибкий пакет, который не включен в системный профиль по умолчанию.

**Листинг 2:** Установка iproute2

```
# emerge sys-apps/iproute2
```

```
# выбор iproute2, а не ifconfig, когда установлены оба  
modules=( "iproute2" )
```

Так как и ifconfig и iproute2 делают очень сходные вещи, то мы сделали их базовую настройку взаимозаменяемой. Например, оба приведенных ниже примера работают не зависимо от того, какой модуль используется.

**Листинг 3:** Примеры ifconfig и iproute2

```
config_eth0=( "192.168.0.2/24" )
```

```
config_eth0=( "192.168.0.2 netmask 255.255.255.0" )
```

```
# также можно указать широковещательный адрес
```

```
config_eth0=( "192.168.0.2/24 brd 192.168.0.255" )
```

```
config_eth0=( "192.168.0.2 netmask 255.255.255.0 broadcast 192.168.0.255" )
```

### 3.c. DHCP

**DHCP** — это способ получения сетевой информации (адреса IP, сервера DNS, шлюза и т.д.) с сервера. Это значит, что если в сети запущен сервер DHCP, вам остается только сказать каждому клиенту, чтобы он использовал DHCP, и сеть настроится сама собой. Конечно, вам придется настраивать все остальное (беспроводную сеть, подключение точка-точка и т.д.), если они должны работать до использования DHCP.

Поддержка DHCP обеспечивается **dhclient**, **dhcpcd**, **pump** или **udhcpc**. У каждого модуля DHCP есть свои плюсы и минусы: здесь мы быстренько рассмотрим их.



Модуль DHCP	Пакет	Плюсы	Минусы
<b>dhclient</b>	net-misc/dhclient	Сделан ISC, теми же людьми, кто делает BIND DNS. Гибок в настройке.	Настройка чрезмерно сложна, программа довольно «распухшая», не может получать данные о серверах NTP с DHCP, по умолчанию не отправляет имя узла.
<b>dhcpcd</b>	net-misc/dhcpcd	Давно в Gentoo по умолчанию, не зависит от внешних утилит.	Более не поддерживается разработчиком, может быть временами медленным, не становится демоном при неограниченном сроке аренды адреса.
<b>pump</b>	net-misc/pump	Компактный, не зависит от внешних утилит.	Более не поддерживается разработчиком, ненадежен, особенно по модему, не может получать данные о серверах NIS по DHCP.
<b>udhcp</b>	net-misc/udhcp	Компактный; наименьший существующий клиент DHCP, сделан для встроенных систем.	Не зарекомендовал себя — ни в одном дистрибутиве не используется по умолчанию; не поддерживает длительность таймаута более 3 секунд.

Если у вас установлено больше одного DHCP клиента, вам нужно указать, какой использовать; иначе по умолчанию используется **dhcpcd**, если есть.

Чтобы передать определенные параметры модулю DHCP, используйте `модуль_eth0="..."` (замените модуль на имя используемого модуля DHCP, например, `dhcpcd_eth0`).

Мы попытались сделать DHCP относительно агностиком: по существу, мы поддерживаем следующие команды, с использованием переменной `dhcpcd_eth0`. По умолчанию не включена ни одна из них.

**release** — освобождать IP-адрес для повторного использования

**nodns** — не замещать `/etc/resolv.conf`

**nntp** — не замещать `/etc/ntp.conf`

**nonis** — не замещать `/etc/yp.conf`

**Листинг 4:** Простая настройка DHCP в `/etc/conf.d/net`  
# требуется только если у вас несколько модулей DHCP  
modules=( "dhcpcd" )

```
config_eth0=( "dhcp" )
dhcpcd_eth0="-t 10" # прекращение после 10 секунд
dhcpc_eth0="release nodns nontp nonis" # только получать адрес
```

**Примечание:** По умолчанию, dhcpcd, udhcpc и rump передают текущее узла на сервер DHCP, поэтому его больше не требуется указывать.

### 3.d. Модем ADSL

Сначала нужно установить программное обеспечение для ADSL.

**Листинг 5:** Установка пакета gr-pppoe  
# emerge net-dialup/gr-pppoe

**Предупреждение:** В baselayout-1.11.x поддерживается только PPPoE. Надеемся, что в будущих версиях появится поддержка PPPoA.

Сейчас нам нужно указать, что на eth0 будет ADSL-интерфейс, и ввести наше имя пользователя, обновив /etc/conf.d/net.

**Листинг 6:** Настройка eth0 для ADSL в /etc/conf.d/net

```
config_eth0=( "adsl" )
adsl_user_eth0="имя-пользователя"
```

Наконец, нужно указать ваше имя и пароль в /etc/ppp/pap-secrets.

**Листинг 7:** Пример /etc/ppp/pap-secrets  
# \* обязательна  
"пользователь" \* "пароль"

### 3.e. APIPA (автоматическая частная IP-адресация)

2APIPA пытается найти свободный адрес в диапазоне 169.254.0.0-169.254.255.255, проверяя отклик на интерфейсе произвольного адреса из этого диапазона по протоколу arp. Если отклика нет, адрес назначается интерфейсу.

Это полезно только в локальных сетях, где нет сервера DHCP, нет прямого подключения к интернету, и все другие компьютеры используют APIPA.

Для поддержки APIPA установите **net-misc/iputils** или **net-analyzer/arping**.

**Листинг 8:** Настройка APIPA в /etc/conf.d/net  
# сначала пробуем DHCP, при неудаче переходим на APIPA  
config\_eth0=( "dhcp" )  
fallback\_eth0=( "apipa" )

# использование только APIPA

```
config_eth0=( "apipa" )
```

### 3.f. Объединение интерфейсов

Для объединения каналов в ствол (bonding) установите **net-misc/ifenslave**.

Объединение используется для повышения пропускной способности сети. Если у вас есть две сетевых карты, выходящих в одну и ту же сеть, можно объединить их, так что ваши приложения увидят только один интерфейс, но реально будут пользоваться двумя сетевыми платами.

**Листинг 9:** Настройка объединения в /etc/conf.d/net

```
# объединение интерфейсов
slaves_bond0="eth0 eth1 eth2"
```

```
# вы можете не захотеть назначать адрес IP объединенному интерфейсу
config_bond0=( "null" )
```

```
# указание зависимости от eth0, eth1 и eth2, так как им может требоваться
# дополнительная настройка
depend_bond0() {
    need net.eth0 net.eth1 net.eth2
}
```

### 3.g. Образование моста (поддержка 802.1d)

Для поддержки мостов установите **net-misc/bridge-utils**.

Мосты используются для объединения сетей. Например, у вас может быть сервер, подсоединенный к интернету через ADSL-модем, и плата беспроводного доступа для предоставления доступа в интернет через ADSL модем другим компьютерам. Чтобы соединить оба интерфейса, можно создать «мост».

**Листинг 10:** Настройка моста в /etc/conf.d/net

```
# настройка моста: подробности в "man brctl"
brctl_br0=( "setfd 0" "sethello 0" "stp off" )
```

```
# включаем порты в мост br0
bridge_br0="eth0 eth1"
```

```
# устанавливаем порты в "null", чтобы не запускался dhcp
config_eth0=( "null" )
config_eth1=( "null" )
```

```
# наконец, даем мосту адрес; можно использовать и DHCP
config_br0=( "192.168.0.1/24" )
```

```
# указываем зависимость от eth0 и eth1, так как им может требоваться
# дополнительная настройка
depend_br0() {
    need net.eth0 net.eth1
}
```

**Важно:** Для использования некоторых вариантов моста вам может потребоваться

### 3.h. MAC-адрес

Для изменения MAC-адреса своего интерфейса вам не потребуется ничего устанавливать, если у вас **sys-apps/baselayout-1.11.14** или новее, и вы собираетесь сменить MAC-адрес на какой-то определенный. Однако, если вам нужно сменить MAC-адрес на случайный, или ваш baselayout старше указанной версии, для смены адреса потребуется установить пакет командой **emerge net-analyzer/macchanger**.

**Листинг 11:** Пример изменения MAC-адреса

```
# установка MAC-адреса интерфейса
mac_eth0="00:11:22:33:44:55"
```

```
# случайная установка последних 3 байт адреса
mac_eth0="random-ending"
```

```
# установка случайного адреса из диапазона для физического соединения
# того же типа (оптического, медного, беспроводного) любого изготовителя
mac_eth0="random-samekind"
```

```
# установка случайного адреса из диапазона для любого типа физического
# соединения (оптического, медного, беспроводного) любого изготовителя
mac_eth0="random-anykind"
```

```
# полностью случайный; ВНИМАНИЕ, некоторые MAC-адреса, сгенерированные
# таким образом, могут вести себя НЕ ТАК, как предполагается
mac_eth0="random-full"
```

### 3.i. Образование туннеля

Для образования туннеля вам не нужно ничего устанавливать, поскольку на это способен сам обработчик интерфейса.

**Листинг 12:** Настройка туннеля в /etc/conf.d/net

```
# для туннелей GRE
iptunnel_vpn0="mode gre remote 207.170.82.1 key 0xffffffff ttl 255"
```

```
# для туннелей IPIP
iptunnel_vpn0="mode ipip remote 207.170.82.2 ttl 255"
```

```
# для настройки интерфейса
config_vpn0=( "192.168.0.2 peer 192.168.1.1" )
```

### 3.j. Виртуальные сети (поддержка 802.1q)

Для поддержки VLAN, установите **net-misc/vconfig**.

Виртуальная локальная сеть (VLAN) — это группа сетевых устройств, которые ведут себя, как будто подключены к одному сегменту сети, даже когда это не так. Членам VLAN видны только члены той же VLAN даже если в той же физической сети

присутствуют другие.

**Листинг 13:** VLAN configuration in /etc/conf.d/net

```
# указание номеров VLAN для интерфейса
# пожалуйста, убедитесь, что ваши номера VLAN НЕ дополнены нулем
vlans_eth0="1 2"
```

```
# можно также настроить VLAN
# за подробностями обращайтесь к man vconfig
vconfig_eth0=( "set_name_type VLAN_PLUS VID_NO_PAD" )
vconfig_vlan1=( "set_flag 1" "set_egress_map 2 6" )
```

```
# настройка интерфейса как обычно
config_vlan1=( "172.16.3.1 netmask 255.255.254.0" )
config_vlan2=( "172.16.2.1 netmask 255.255.254.0" )
```

**Важно:** Для использования некоторых вариантов VLAN вам может потребоваться обращение к документации по именам переменных.

## 4. Беспроводная сеть

### 4.a. Введение

В настоящее время поддерживается подключение к беспроводной сети с помощью **wireless-tools** или **wpa\_supplicant**. Важно помнить, что подключение к беспроводным сетям настраивается глобально, а не для определённого интерфейса.

- **wpa\_supplicant** — лучший выбор, но он поддерживает не все драйверы. Список поддерживаемых драйверов находится на сайте [wpa\\_supplicant](http://wpa_supplicant.org). Кроме того, сейчас **wpa\_supplicant** может подключаться только к тем сетям, на SSID которых настроен.
- **wireless-tools** поддерживает практически все платы и драйверы, но не способен подключаться к точкам доступа, работающим исключительно с WPA.

**Предупреждение:** Драйвер **linux-wlan-ng** в данный момент не поддерживается в **baselayout**. Это из-за того, что в **linux-wlan-ng** своя собственная программа установки и настройки, которая ни на что не похожа. Разработчики **linux-wlan-ng**, по слухам, собираются перейти на установку как в **wireless-tools**; когда это произойдет, вы сможете использовать **linux-wlan-ng** с **baselayout**.

### 4.b. Запросчик WPA

Запросчик **WPA (WPA Supplicant)** — пакет, позволяющий подсоединяться к точкам доступа с протоколом WPA. Его настройка проходит достаточно гладко, и пакет работает достаточно стабильно, хотя находится на стадии бета-тестирования.

**Листинг 1:** Установка **wpa\_supplicant**  
**# emerge net-wireless/wpa\_supplicant**

**Важно:** Для работы **wpa\_supplicant** в ядре должен быть включен параметр

Теперь нам необходимо настроить `/etc/conf.d/net` для предпочтения **wpa\_supplicant** по отношению к **wireless-tools** (по умолчанию, если обе программы установлены, работает **wireless-tools**).

**Листинг 2:** Настройка `/etc/conf.d/net` для `wpa_supplicant`

```
# выбор wpa_supplicant
modules=( "wpa_supplicant" )

# важно указать wpa_supplicant, какой драйвер нужно использовать,
# так как программа пока не слишком хорошо угадывает сама

wpa_supplicant_eth0="-Дбезумный-wifi"
```

**Примечание:** Если вы используете драйвер **host-ap**, то вам потребуется перевести плату в ведомый режим (managed mode), прежде чем она сможет правильно работать с **wpa\_supplicant**. Для этого можно указать `iwconfig_eth0="mode managed"` в `/etc/conf.d/net`.

Это довольно просто, не так ли? Однако, нужно настроить саму программу `wpa_supplicant`, что значительно сложнее. Сложность зависит от степени защиты точек доступа, к которым вы собираетесь подключаться. Следующий упрощенный пример взят из `/etc/wpa_supplicant.conf.example`, поставляемого в составе `wpa_supplicant`.

**Листинг 3:** Пример `/etc/wpa_supplicant.conf`

```
# следующую строку нельзя изменять, иначе программа не сможет работать
ctrl_interface=/var/run/wpa_supplicant

# ограничим доступ к настройкам WPA только для root
ctrl_interface_group=0

# пусть wpa_supplicant заботится о сканировании и выборе точки доступа
ap_scan=1

# простой случай: WPA-PSK, согласованный ключ - текстовая строка,
# принимать любой допустимый шифр
network={
    ssid="просто"
    psk="очень тайный пароль"
    # чем выше приоритет, тем скорее выбор
    priority=5
}

# как в предыдущем, но с запросом сканирования по определенному SSID
# (для точек доступа, отклоняющих широковещательный SSID)
network={
    ssid="второй ssid"
    scan_ssid=1
    psk="очень тайный пароль"
    priority=2
}

# использовать только WPA-PSK; принимать любое допустимое сочетание шифров

network={
    ssid="пример"
    proto=WPA
```

```

key_mgmt=WPA-PSK
pairwise=CCMP TKIP
group=CCMP TKIP WEP104 WEP40
psk=06b4be19da289f475aa46a33cb793029d4ab3db7a23ee92382eb0106c72ac7bb
priority=2
}

```

# открытое подключение без шифрования (не WPA, не IEEE #802.1X)

```

network={
  ssid="тест-открытого-текста"
  key_mgmt=NONE
}

```

# подключение с общим ключом WEP (не WPA, не IEEE #802.1X)

```

network={
  ssid="тест-статического-wep"
  key_mgmt=NONE
  wep_key0="abcde"
  wep_key1=0102030405
  wep_key2="1234567890123"
  wep_tx_keyidx=0
  priority=5
}

```

# подключение с общим ключом WEP (не WPA, не IEEE #802.1X),  
# допуск с использованием общего ключа IEEE 802.11

```

network={
  ssid="тест2-статического-wep"
  key_mgmt=NONE
  wep_key0="abcde"
  wep_key1=0102030405
  wep_key2="1234567890123"
  wep_tx_keyidx=0
  priority=5
  auth_alg=SHARED
}

```

# сеть IBSS/ad-hoc с WPA-None/TKIP

```

network={
  ssid="тест adhoc"
  mode=1
  proto=WPA
  key_mgmt=WPA-NONE
  pairwise=NONE
  group=TKIP
  psk="тайный пароль"
}

```

## 4.с. Утилиты Wireless tools

### Начальная установка и режим ведомого

Wireless Tools обеспечивают общий способ настройки базовых беспроводных интерфейсов, вплоть до защиты **WEP**. Хотя **WEP** является слабым методом защиты, он наиболее распространен.

Для настройки **Wireless Tools** служат несколько основных переменных. В примере файла настроек, приведенном ниже, описано все, что вам потребуется. Нужно

помнить, что отсутствие настройки означает «подключаться к нешифрующей точке доступа с самым сильным сигналом». Программа будет всегда пытаться подключить вас к чему-нибудь.

**Листинг 4:** Установка wireless-tools  
# emerge net-wireless/wireless-tools

**Примечание:** Хотя вы можете хранить свои параметры настройки беспроводной сети в **/etc/conf.d/wireless**, это руководство рекомендует хранить их в **/etc/conf.d/net**.

**Важно:** Вам понадобится обратиться к документации по именам переменных.

**Листинг 5:** Пример настройки iwconfig /etc/conf.d/net  
# приоритет использования iwconfig над wpa\_supplicant  
modules=( "iwconfig" )

```
# Настройка ключей WEP для точек доступа ESSID1 и ESSID2
# Можно указывать до 4 ключей WEP, но только 1 может работать в каждый
# момент, поэтому мы указываем индекс по умолчанию [1], чтобы сделать ключ [1],
# а впоследствии снова, чтобы изменить активный ключ на [1].
# Это нужно, если вы настраиваете другие ESSID на использование WEP-ключей,
# отличающихся от [1].
#
# Приставка s: перед ключом означает, что ключ текстовый, иначе -
# шестнадцатичный
#
# enc open указывает открытую защиту (более безопасно)
# enc restricted указывает ограниченную защиту (менее безопасно)
key_ESSID1="[1] s:ваш-ключ-здесь key [1] enc open"
key_ESSID2="[1] aaaa-bbbb-cccc-dd key [1] enc restricted"
#
# Нижеследующее работает только при поиске доступных точек доступа.

# Иногда видны несколько точек доступа, и требуется задать
# предпочтительный порядок подключения
preferred_aps=( "ESSID1" "ESSID2" )
```

## Настройка порядка выбора точки доступа

Можно указать несколько дополнительных параметров для уточнения порядка выбора точки доступа, но обычно этого не требуется.

Вам решать, подключаться ли только к указанным точкам доступа, или нет. По умолчанию, если подключение ко всем настроенным точкам доступа не удалось, и есть возможность подключиться к не шифрующей точке доступа, такое соединение произойдет. Этот порядок зависит от переменной **associate\_order**. Ниже приводится таблица значений и с описанием их действия:



Значение	Описание
<b>any</b>	поведение по умолчанию
<b>preferredonly</b>	соединяться только с видимыми точками доступа из списка
<b>forcepreferred</b>	принудительно подключаться к точкам доступа в заданной последовательности, если они не обнаружены при сканировании
<b>forcepreferredonly</b>	не сканировать точки доступа, просто пытаться подключиться к каждой по списку
<b>forceany</b>	так же, как в forcepreferred + подключаться к любой доступной точке доступа

Наконец, мы можем указать blacklist\_aps и unique\_ap. blacklist\_aps работает подобно preferred\_aps. unique\_ap устанавливается в yes или no, указывая, можно ли подключать второй беспроводной интерфейс к той же точке доступа, что и первый.

**Листинг 6:** Пример blacklist\_aps и unique\_ap

```
# иногда требуется полностью исключить возможность подключения
# к определенным точкам доступа
blacklist_aps=( "ESSID3" "ESSID4" )

# если у вас несколько беспроводных плат, можно указать, можно ли им
# подключаться к одной и той же точке доступа
# значение - "yes" или "no"
# по умолчанию - "yes"
unique_ap="yes"
```

## Режим отдельного и ведущего узла

Если вы хотите становиться отдельным узлом (ad hoc), когда не удается подключиться ни к какой точке доступа в ведомом режиме, это тоже возможно.

**Листинг 7:** Откат к режиму отдельного узла

```
adhoc_essid_eth0="Этот отдельный узел"
```

Как насчет подключения к сетям Ad-Hoc или запуска в режиме ведущего (master), чтобы стать точкой доступа? Есть конфигурация и для такой работы! Вам может потребоваться определить WEP-ключи, как показано выше.

**Листинг 8:** Пример настройки ad-hoc/master

```
# установка режима: допускается managed (ведомый, по умолчанию),
# ad-hoc (отдельный) или master (ведущий). Не все драйверы поддерживают
# каждый режим
mode_eth0="ad-hoc"

# установка ESSID интерфейса
# в ведомом режиме заставляет интерфейс пытаться подключиться к указанному
```

```
# ESSID, и больше ничего
essid_eth0="Этот отдельный узел"
```

```
# если не указан, используется канал 3
channel_eth0="9"
```

**Важно:** Следующий текст взят дословно из документации BSD wavelan, входящей в документацию NetBSD. «Существуют 14 каналов. Нам сообщили, что использование каналов с 1 по 11 является законным в Северной Америке, каналов с 1 по 13 — в большинстве стран Европы, каналов с 10 по 13 — во Франции, и только канала 14 — в Японии. Если у вас есть сомнения, обратитесь к документации от вашей платы или точки доступа. Убедитесь что выбранный канал совпадает с каналом точки доступа (или другой платы в сети ad-hoc). По умолчанию на платах, продаваемых в Северной Америке и большинстве стран Европы, настроен канал 3; на платах, продаваемых во Франции — канал 11; на платах, продаваемых в Японии — канал 14.»

## Устранение неполадок в wireless tools

Существуют дополнительные переменные, которые можно использовать для запуска своего беспроводного оборудования и устранения неполадок, возникших из-за драйвера или проблем с сетевым окружением. Ниже приведена таблица прочих функций, которые можно перепробовать:

Переменная	Значение по умолчанию	Описание
<b>iwconfig_eth0</b>		За подробными сведениями о параметрах iwconfig обращайтесь к странице справки iwconfig.
<b>iwpriv_eth0</b>		За подробными сведениями о параметрах iwpriv обращайтесь к странице справки iwpriv
<b>sleep_scan_eth0</b>	0	Время задержки в секундах перед попыткой сканирования. Требуется, когда драйверу или прошивке нужно дополнительное время для перехода в рабочий режим.
<b>sleep_associate_eth0</b>	5	Время ожидания связи интерфейса с точкой доступа (в секундах) перед переходом к опросу следующей
<b>associate_test_eth0</b>	MAC	Некоторые драйверы не сбрасывают MAC-адрес, связанный с недоступной точкой доступа, при потере или попытке связи. Некоторые драйверы не сбрасывают значение качества сигнала при потере или попытке соединения. Допустимые значения: <b>MAC</b> , <b>quality</b> и <b>all</b> .
<b>scan_mode_eth0</b>		Некоторым драйверам необходимо сканировать в режиме ad-hoc. Если сканирование не удастся, попробуйте указать здесь ad-hoc.
<b>iwpriv_scan_pre_eth0</b>		Посылать интерфейсу некоторые команды

		iwpriv перед сканированием. За дополнительными сведениями обращайтесь к странице справки iwpriv.
iwpriv_scan_post_eth 0		Посылать интерфейсу некоторые команды iwpriv после сканирования. За дополнительными сведениями обращайтесь к странице справки iwpriv.

#### 4.d. Раздельная настройка сети по ESSID

Иногда необходим статический IP при соединении с ESSID1, и DHCP при соединении с ESSID2. На деле, большинство переменных модуля можно определять раздельно по ESSID. Вот как это сделать:

**Примечание:** Это работает при использовании WPA Supplicant или Wireless Tools.

**Важно:** Вам потребуется свериться с документацией по именам переменных.

**Листинг 9:** Назначение сетевых настроек для ESSID

```
config_ESSID1=( "192.168.0.3/24 brd 192.168.0.255" )
routes_ESSID1=( "default via 192.168.0.1" )

config_ESSID2=( "dhcp" )
fallback_ESSID2=( "192.168.3.4/24" )
fallback_route_ESSID2=( "default via 192.168.3.1" )

# можно также указать сервера имен и др.
# ПРЕДУПРЕЖДЕНИЕ: DHCP переопределит настройки, если не указано иное
dns_servers_ESSID1=( "192.168.0.1" "192.168.0.2" )
dns_domain_ESSID1="some.domain"
dns_search_domains_ESSID1="search.this.domain search.that.domain"

# перенастройка производится по MAC-адресу точки доступа;
# это полезно, когда в разных местах есть точки доступа с одинаковым ESSID
config_001122334455=( "dhcp" )
dhcpcd_001122334455="-t 10"
dns_servers_001122334455=( "192.168.0.1" "192.168.0.2" )
```

### 5. Дополнительные возможности

#### 5.a. Стандартные функции-обработчики

Можно определить четыре функции, которые вызываются при операциях запуска (start) и останова (stop). При вызове функциям передается название интерфейса, так что одна и та же функция может управлять несколькими адаптерами.

Для указания на то, что запуск или останов интерфейса может продолжаться, возвращаемое значение функций **preup()** и **predown()** должно быть нулевым (успешным). Если **preup()** возвращает ненулевое значение, запуск интерфейса прерывается. Если **predown()** возвращает ненулевое значение, не допускается

продолжение останова интерфейса.

Возвращаемые значение функций `postup()` и `postdown()` игнорируется, так как показываемая ими ошибка не обрабатывается.

- **`${IFACE}`** присваивается название запускаемого/останавливаемого интерфейса.
- **`${IFVAR}`** — это значение `${IFACE}`, преобразованное в имя переменной, разрешенное в `bash`.

**Листинг 1:** Примеры функций до/после запуска/останова

```
preup() {
# Проверка соединения интерфейса перед его запуском. Она
# работает лишь с некоторыми сетевыми адаптерами и требует наличия
# установленного пакета mii-diag.
if mii-tool ${IFACE} 2> /dev/null | grep -q 'no link'; then
    ewarn "Интерфейс ${IFACE} не подключен, прерывание запуска"
    return 1
fi

# Проверка соединения интерфейса перед его запуском. Она
# работает лишь с некоторыми сетевыми адаптерами и требует наличия
# установленного пакета ethtool.
if ethtool ${IFACE} | grep -q 'Link detected: no'; then
    ewarn "Интерфейс ${IFACE} не подключен, прерывание запуска"
    return 1
fi

# Не забываем вернуть 0 при успехе
return 0
}

predown() {
# Назначение этого сценария - проверить наличие корня NFS
# и в этом случае предотвратить останов интерфейсов. Заметьте, что
# определяя функцию predown(), вы отменяете существующую логику.
# Вот она, на случай если все же понадобится...
if is_net_fs /; then
    eerror "Корневая ФС смонтирована в сети - останов ${IFACE} невозможен"
    return 1
fi

# Не забываем вернуть 0 при успехе
return 0
}

postup() {
# Эту функцию можно использовать, например, для регистрации в
# службе динамического DNS. Другой пример - отправка/прием почты после
# запуска интерфейса.
return 0
}

postdown() {
# Эта функция приводится в основном для полноты... Я не придумал,
# что бы ценное в нее поместить ;-)
return 0
}
```

## 5.b. Функции-обработчики wireless tools

**Примечание:** Это не работает вместе с WPA Supplicant, но переменные `{ESSID}` и `{ESSIDVAR}` доступны в функции `postup()`.

Можно определить две функции, вызываемые до и после функции подключения (`associate`). При вызове им сначала передается название интерфейса, так что одна и та же функция может управлять несколькими адаптерами.

Для указания на то, что запуск или останов интерфейса можно продолжать, возвращаемое значение функции `preassociate()` должно быть нулевым (успешным). Если `preassociate()` возвращает ненулевое значение, запуск интерфейса прерывается.

Возвращаемое значение функции `postassociate()` игнорируется, так как показываемая ей ошибка не обрабатывается.

`{ESSID}` присваивается точный **ESSID** точки доступа, к которой вы подключаетесь. `{ESSIDVAR}` — это `{ESSID}`, преобразованный в имя переменной, разрешенное в `bash`.

**Листинг 2:** Функции до/после соединения

```
preassociate() {
# Ниже добавляются две конфигурационных переменных, leap_user_ESSID
# и leap_pass_ESSID. Когда они обе настроены на подключаемый ESSID,
# мы запускаем сценарий CISCO LEAP

local user pass
eval user="\${leap_user_${ESSIDVAR}}\"
eval pass="\${leap_pass_${ESSIDVAR}}\"

if [[ -n ${user} && -n ${pass} ]]; then
if [[ ! -x /opt/cisco/bin/leapscript ]]; then
eend "Для поддержки LEAP, выполните emerge net-misc/cisco-aironet-client-utils"
return 1
fi
einfo "Ожидание допуска LEAP на \"${ESSID}/\${ESSIDVAR}\""
if /opt/cisco/bin/leapscript ${user} ${pass} | grep -q 'Login incorrect'; then
warn "Вход пользователя ${user} не удался"
return 1
fi
fi

return 0
}

postassociate() {
# Эта функция приводится в основном для полноты... Я не придумал,
# что бы ценное в нее поместить ;- )

return 0
}
```

**Примечание:** `{ESSID}` и `{ESSIDVAR}` недоступны в функциях `predown()` и `postdown()`.

## 6. Управление сетью

### 6.a. Управление сетью

Если вы часто берете компьютер в поездки, то у вас не всегда может быть возможность подключения к сети Ethernet или к беспроводной точке доступа. Но мы можем захотеть, чтобы сеть заработала автоматически, как только к компьютеру подключен кабель Ethernet или найдена беспроводная точка доступа.

Здесь вы найдете некоторые инструменты, которые помогут это организовать.

**Примечание:** В этом документе рассказывается только о **ifplugd**, но есть и альтернативные решения, например, **quickswitch**.

### 6.b. ifplugd

**ifplugd** — это программа, которая запускает и останавливает интерфейс при подключении или отключении кабеля к сети Ethernet. Также она может обработать подключение к беспроводной точке доступа или появление новых точек доступа.

**Листинг 1:** Установка ifplugd  
**# emerge sys-apps/ifplugd**

Настройка **ifplugd** — довольно простая задача. Файл конфигурации расположен по адресу: **/etc/conf.d/ifplugd**. Для просмотра подробного описания переменных запустите **man ifplugd**.

**Листинг 2:** Пример конфигурации ifplug  
# назначение интерфейса для слежения  
**INTERFACES="eth0"**

```
AUTO="no"
BEEP="yes"
IGNORE_FAIL="yes"
IGNORE_FAIL_POSITIVE="no"
IGNORE_RETVAL="yes"
POLL_TIME="1"
DELAY_UP="0"
DELAY_DOWN="0"
API_MODE="auto"
SHUTDOWN="no"
WAIT_ON_FORK="no"
MONITOR="no"
ARGS=""
```

```
# дополнительные параметры ifplugd для указанного интерфейса.
# учтите, установки глобальных переменных игнорируются, если указаны значения
# для конкретного интерфейса
MONITOR_wlan0="yes"
DELAY_UP_wlan0="5"
DELAY_DOWN_wlan0="5"
```

# Настольная книга по безопасности Gentoo

Ссылка на оригинал: <http://www.gentoo.org/doc/ru/security/>

С версии: 1.3

## Введение

Эта настольная книга предназначена для людей, использующих Gentoo Linux в качестве сервера или чувствующих, что они нуждаются в повышении безопасности системы.

Если после прочтения этой книги вы заинтересуетесь дальнейшим усилением защищенности системы Gentoo, то обратите внимание на проект укреплённой системы Gentoo — [Hardened Gentoo Project](#) (англ.).

## А. Безопасность системы

### 1. Соображения перед началом установки

#### 1.a. Физическая безопасность

Сколько преград не установи, злоумышленник, имея физический доступ к компьютеру, сможет легко их обойти. Несмотря на это, кое-какие меры для относительной защиты от злоумышленника с физическим доступом к вашей машине принять можно. Поместив оборудование в кладовку под замок, вы помешаете злоумышленнику попросту отключить его и унести с собой. Стоит запереть корпус компьютера на замок, чтобы не дать злоумышленнику просто вынуть из него жесткий диск. Чтобы предотвратить загрузку с альтернативного диска, благодаря чему можно просто обойти порядок входа в систему и права доступа, попробуйте установить в BIOS жесткий диск в качестве первого загрузочного устройства и запаролить BIOS. Также важно установить пароль для загрузчика LILO или GRUB, чтобы не дать недобросовестному пользователю загрузить систему в однопользовательском режиме, получив к ней полный доступ. Это подробно описано в третьей главе, в разделах [защита GRUB паролем](#) и [защита LILO паролем](#).

#### 1.b. Планирование демонов/служб

Для начала выпишите службы, которые должны работать на машине. Это поможет вам выбрать наилучшую схему разбивки разделов в системе и позволит лучше спланировать меры безопасности. Конечно же, в этом нет необходимости, если ваш компьютер служит для одной определенной задачи, например в качестве рабочей станции или выделенного межсетевого экрана. В таких случаях не должны запускаться никакие службы, за исключением, возможно, sshd.

Составленный список также может пригодиться для администрирования системы. Вы увидите, что, поддерживая перечень текущих версий, значительно легче удерживать все в актуальном состоянии, когда в каких-либо из ваших демонов обнаруживаются уязвимости к удаленному доступу.

## 1.с. Схемы создания разделов

Правила разбиения разделов:

- любое дерево каталогов, в которое пользователь должен иметь возможность записи (например, /home, /tmp), должно размещаться в отдельном разделе с использованием дисковых квот. Это снижает риск переполнения всей файловой системы каким-либо пользователем. Portage использует /var/tmp для компиляции, поэтому этот раздел должен быть большим
- любое дерево каталогов, в которое вы планируете устанавливать программное обеспечение не средствами управления пакетами дистрибутива, следует размещать в отдельном разделе. Согласно [стандарту иерархии файлов](#) (англ.), это /opt или /usr/local. Являясь отдельными разделами, они не удаляются при переустановке системы
- для лишней защиты можно поместить статические данные в отдельный раздел, монтируемый только для чтения. Если вы настоящий параноик, можете пользоваться непerezаписываемым носителем, например, компакт-диском

### 1.d. Пользователь root

Администратор системы (пользователь root) является наиболее важным пользователем, и его права не следует использовать без крайней необходимости. Если злоумышленник получит доступ с правами администратора, единственным способом восстановить доверие к системе будет переустановить ее.

#### Золотые правила для root

- всегда создавайте пользователя для каждодневных работ, а если этому пользователю понадобятся права администратора, добавьте его в группу «wheel». Это дает возможность обычным пользователям выполнять команды с правами администратора с помощью su
- никогда не запускайте X или другое пользовательское приложение под root. Права администратора следует использовать только при крайней необходимости: если уязвимость есть в приложении, запущенном от имени пользователя, злоумышленник может получить права этого пользователя. А если приложение запущено с правами администратора, то злоумышленник получит их.
- войдя под учетной записью администратора, всегда указывайте абсолютные пути (или всегда используйте команду su -, которая заменяет переменные среды пользователя на администраторские, обеспечивая наличие в PATH администратора только защищенных каталогов, например /bin и /sbin). Иначе можно обмануть администратора и заставить его выполнить совершенно другое приложение. Если PATH администратора защищен или он использует только абсолютные пути, то можно быть уверенным, что это не произойдет
- если пользователю нужно выполнять лишь несколько команд с правами администратора, то вместо использования учетной записи root, рекомендуется использовать команду sudo. Просто также задумывайтесь, кому вы даете доступ к этой команде!
- никогда не оставляйте без присмотра терминал, в котором вы зарегистрированы в качестве root

В Gentoo по умолчанию есть некоторая защита от обычных пользователей, пытающихся с помощью su получить права администратора. По умолчанию настройки PAM требуют, чтобы пользователь был членом группы «wheel» для того,



чтобы запускать команду su.

### 1.е. Правила безопасности

Есть ряд причин набросать правила безопасности для своей системы или сети.

- хорошие правила безопасности позволяют вам наметить подход к безопасности системно, а не сваливать в кучу разрозненные меры. Например, без правил администратор может решить отключить telnet, так как в нем пароли передаются незашифрованными, оставив доступ по FTP, обладающему таким же недостатком. Хорошие правила безопасности позволяют выявить, какие меры безопасности имеют смысл, а какие — нет
- чтобы выявлять проблемы, проводить аудит или отслеживать злоумышленников, может потребоваться перехват данных, передаваемых по сети, проверка истории входа пользователей и выполнявшихся ими команд, а также просмотр их домашних каталогов. Без указания об этом в документации и явного предупреждения пользователей такие действия могут оказаться незаконными и привести к привлечению вас к ответственности по закону
- похищенные учетные записи пользователей являются одной из самых распространенных угроз безопасности системы. Не объясняя пользователям, почему безопасность важна, и как ее поддерживать (например, не записывать пароли на клочках бумаги на рабочем месте), можно и не надеяться на защищенность пользовательских данных
- хорошо задокументированная схема сети и системы поможет как вам, так и, при необходимости, суду и следствию, отслеживать проникновение злоумышленника и выявлять слабые места по факту взлома. Уведомление правил безопасности о том, что ваша система является частной сетью и несанкционированный доступ запрещен, также способствует надлежащему преследованию нарушителя по закону после его поимки

Надеемся, что теперь необходимость в хороших правилах безопасности более чем ясна.

Сами по себе правила — это документ или набор документов, раскрывающих функции сети или системы (например, какие именно услуги предоставляются), допустимые и запрещенные действия, применение «передового опыта» обеспечения безопасности, и т. д. Все пользователи должны быть знакомы как с правилами безопасности, так и с изменениями, которые вы вносите для поддержания их актуальности. Важно уделять время, чтобы добиваться понимания правил пользователями, объяснять, зачем знакомится под роспись, и что будет, если пользователь явно нарушит их требования (в правилах безопасности об этом должно быть ясно указано). Необходимо повторять ознакомление по крайней мере раз в год, так как правила подвержены изменениям (а также для напоминания пользователям).

**Примечание:** Создавайте правила, понятные для чтения и не допускающие разночтений в любой ситуации.

Правила безопасности должны охватывать по крайней мере следующие темы:

- допустимые действия
  - экранные заставки
  - обращение с паролями
  - скачивание и установка программ
  - предупреждение, если действия пользователей отслеживаются
  - использование антивирусных программ
- обращение с ценными сведениями (в любом письменном виде, на

- электронном или бумажном носителе)
  - чистый рабочий стол и хранение сведений ДСП под замком
  - отключение компьютера перед уходом
  - использование средств шифрования
  - передача ключей доверенным сотрудникам
  - обращение с конфиденциальными данными в поездках
- обращение с компьютерным оборудованием в поездках
  - обращение с ноутбуком в поездках и в гостиницах

Для разных пользователей могут требоваться разные типы или уровни доступа, и содержание ваших правил может отличаться, чтобы охватить их все.

В распухших правилах безопасности легко упустить важные моменты. В правилах для ИТ-персонала могут содержаться сведения, закрытые для обычных пользователей. Поэтому есть смысл разделить их на несколько небольших разделов, например, «допустимые действия», «использование паролей», «правила использования электронной почты», «правила удаленного доступа».

Примеры правил безопасности приведены на сайте [проекта правил безопасности SANS](#) (англ.). Если у вас небольшая сеть, и вы считаете, что эти примеры слишком велики, можете обратиться к [руководству по объектовой безопасности \(Site Security Handbook\)](#) (англ.).

## 2. Закручивание гаек

### 2.a. USE-флаги

В Gentoo Linux файл `make.conf` содержит USE-флаги, определенные пользователем, а `/etc/make.profile/make.defaults` — USE-флаги по умолчанию. Для данного руководства важны флаги `ram` (Pluggable Authentication Modules — подключаемые модули опознания), `tcpd` (Упаковщики TCP) и `ssl` (Secure Socket Layer). Все они включены в USE-флаги по умолчанию.

### 2.b. Защита GRUB паролем

В GRUB можно защитить загрузчик паролем двумя различными способами. В первом используется открытый пароль, а во втором — шифрование с использованием `md5+salt`.

#### Листинг 1: `/boot/grub/grub.conf`

```
timeout 5
password changeme
```

Благодаря этим строкам был добавлен пароль `changeme`. Если во время загрузки не ввести пароль, то GRUB просто загружает вариант по умолчанию.

При добавлении пароля MD5 вы должны преобразовать открытый пароль в зашифрованный в том же формате, что и в файле `/etc/shadow`. За дополнительными сведениями обращайтесь к `man crypt`. Зашифрованный пароль, например `changeme`, может выглядеть вот так: `$1$T7/dgdIJ$dJM.n2wZ8RG.oEilOwJUs`.

Зашифровать пароль можно прямо в оболочке GRUB:

#### Листинг 2: `md5crypt` в оболочке `grub`

```
#!/sbin/grub

GRUB version 0.92 (640K lower / 3072K upper memory)
```

```
[ Minimal BASH-like line editing is supported. For the first word, TAB lists possible command completions. Anywhere else TAB lists the possible completions of a device/filename. ]
```

```
grub> md5crypt
```

```
Password: *****
```

```
(было набрано слово changeme)
```

```
Encrypted: $1$T7/dgdIJ$dJM.n2wZ8RG.oEiIOwJUs.
```

```
grub> quit
```

Затем скопируйте полученный пароль в /boot/grub/grub.conf.

#### Листинг 3: /boot/grub/grub.conf

```
timeout 5
password --md5 $1$T7/dgdIJ$dJM.n2wZ8RG.oEiIOwJUs.
```

Задержка в 5 секунд пригодится в случае, когда система физически недоступна, и нужна возможность перезагрузки без использования клавиатуры. Более подробно о паролях в GRUB вы можете узнать, набрав info grub.

### 2.c. Защита LILO паролем

В LILO также поддерживается два способа защиты с помощью пароля: общий и для отдельного образа, в обоих случаях пароль хранится в открытом виде.

Общий пароль устанавливается в начале файла конфигурации и относится к каждому загрузочному образу:

#### Листинг 4: /etc/lilo.conf

```
password=changeme
restricted
delay=3
```

Пароль для отдельного образа устанавливается следующим образом:

#### Листинг 5: /etc/lilo.conf

```
image=/boot/bzImage
    read-only
    password=changeme
    restricted
```

Если параметр restricted не введен, то пароль будет запрашиваться при каждой загрузке.

Чтобы сохранить новые сведения в файле lilo.conf, нужно запустить /sbin/lilo.

### 2.d. Ограничение доступа к консоли

С помощью файла /etc/securetty можно указать, с каких терминальных устройств (tty) можно входить в систему администратору.

Рекомендуется закомментировать все строки, кроме vc/1 (если у вас devfs), или кроме tty1 (если у вас udev). Благодаря этому суперпользователь сможет входить в систему одновременно только один раз и только с одного терминала.

**Примечание:** Пользователи, входящие в группу «wheel» по-прежнему смогут

воспользоваться su -, чтобы стать администратором, работая с других терминалов.

#### Листинг 6: /etc/securetty

```
(для devfs)
vc/1
(для udev)
tty1
```

## 3. Журналирование

### 3.a. Введение

Для перехвата предупреждений и ошибок, которые могут свидетельствовать о происходящей атаке или успешном взломе, следует подключать дополнительное журналирование. Часто злоумышленники сканируют систему и «проверяют ее на зуб» перед атакой.

Также немаловажно, чтобы файлы журналов были легко читаемыми и обозримыми. В Gentoo Linux при установке на выбор предлагается 3 разных средства журналирования.

### 3.b. Журналирование: Syslogd

Syslogd является самым распространенным средством журналирования для Linux и Unix. Он способен делать ротацию журналов, однако использование /usr/sbin/logrotate с вызовом по расписанию (logrotate настраивается с помощью файла /etc/logrotate.conf) может оказаться полезней, так как у logrotate есть много дополнительных возможностей. Частота ротации подбирается в зависимости от загрузки системы.

Ниже приведен стандартный файл syslog.conf с некоторыми добавлениями. Мы раскомментировали строки cron и tty, и добавили удаленный сервер журналирования. Для дальнейшего усиления журналирования можно настроить ведение журнала в двух местах.

#### Листинг 1: /etc/syslog.conf

```
# /etc/syslog.conf      Файл настройки syslogd.
#
#                       За дополнительной информацией обращайтесь к
#                       странице справки syslog.conf(5) .
#                       Взято из Debian, пока пользуемся им
#                       Daniel Robbins, 5/15/99
#
# Сначала стандартные файлы журналов.  Журналирование по подсистемам.
#
auth,authpriv.*        /var/log/auth.log
*.*;auth,authpriv.none -/var/log/syslog
cron.*                 /var/log/cron.log
daemon.*               -/var/log/daemon.log
kern.*                 -/var/log/kern.log
lpr.*                  -/var/log/lpr.log
mail.*                 /var/log/mail.log
user.*                 -/var/log/user.log
uucp.*                 -/var/log/uucp.log
local6.debug           /var/log/imapd.log
```

```

#
# Журналирование почтовой системы. Разбито на части, чтобы
# легко писать сценарии для разбора этих файлов.
#
mail.info                -/var/log/mail.info
mail.warn                -/var/log/mail.warn
mail.err                 /var/log/mail.err

# Журналирование новостной системы INN
#
news.crit                /var/log/news/news.crit
news.err                 /var/log/news/news.err
news.notice              -/var/log/news/news.notice

#
# Журналы, куда может попадать "все подряд".
#
*.=debug;\
    auth,authpriv.none;\
    news.none;mail.none    -/var/log/debug
*.=info;*.=notice;*.=warn;\
    auth,authpriv.none;\
    cron,daemon.none;\
    mail,news.none         -/var/log/messages

#
# Предупреждения и чрезвычайные сообщения, посылаемые всем вошедшим.
#
*.emerg                  *
*.=alert                  *

#
# Мне нравится вывод сообщений на консоль, но только на ту виртуальную
# консоль, что я обычно оставляю простаивать.
#
daemon,mail.*;\
    news.=crit;news.=err;news.=notice;\
    *.=debug;*.=info;\
    *.=notice;*.=warn      /dev/tty8

# Настройка сервера удаленного журналирования
*.*                      @logserver

# Именованный канал /dev/xconsole - для утилиты `xconsole'. Чтобы использовать,
# следует запускать `xconsole' с параметром `-file':
#
# $ xconsole -file /dev/xconsole [...]
#
# ПРИМЕЧАНИЕ: измените список, данный ниже, или вы с ума сойдете, если
# у вас относительно загруженная площадка..
#
#daemon.*;mail.*;\
#    news.crit;news.err;news.notice;\
#    *.=debug;*.=info;\
#    *.=notice;*.=warn      | /dev/xconsole

local2.*                  --/var/log/ppp.log

```

Злоумышленники, скорее всего, будут стараться уничтожать следы своего пребывания, исправляя или удаляя файлы журналов. Можно затруднить им работу, ведя журналирование на одном или нескольких удаленных серверах журнала, расположенных на других компьютерах. Узнать больше о syslogd можно, запустив `man syslog`.

### 3.c. Metalog

[Metalog](#), написанный Frank Dennis, не может отправлять журнал на удаленный сервер, но дает преимущество, когда речь о скорости и гибкости журналирования. Он может вести журналы по названию программы, срочности, подсистеме (как и syslogd), и комплектуется разборщиком регулярных выражений, с помощью которых можно запускать внешние сценарии при обнаружении заданных шаблонов. При необходимости это может оказаться очень полезным.

Стандартной настройки обычно достаточно. Если необходимо, чтобы вам отправлялось письмо при любом неверно введенном пароле, используйте один из следующих сценариев.

Для postfix:

Листинг 2: /usr/local/sbin/mail\_pwd\_failures.sh для postfix

```
#!/bin/sh
echo "$3" | mail -s "Warning (program : $2)" root
```

Для netqmail:

Листинг 3: /usr/local/sbin/mail\_pwd\_failures.sh for netqmail

```
#!/bin/sh
echo "To: root
Subject:Failure (Warning: $2)
$3
" | /var/qmail/bin/qmail-inject -f root
```

Не забудьте сделать сценарий исполняемым командой `/bin/chmod +x /usr/local/sbin/mail_pwd_failures.sh`.

Затем в файле `/etc/metalog/metalog.conf` раскомментируйте строку под «Password failures»:

Листинг 4: /etc/metalog/metalog.conf

```
command = "/usr/local/sbin/mail_pwd_failures.sh"
```

### 3.d. Syslog-ng

Syslog-ng, за некоторыми отличиями, обладает теми же возможностями, что и syslog и metalog. Он может фильтровать сообщения по уровню и содержанию (как metalog), вести удаленное журналирование, как syslog, обрабатывать журналы, получаемые от syslogd (даже потоки от Solaris), записывать в ТТУ, запускать программы, и служить сервером журналирования. То есть, он сочетает лучшие черты обоих средств журналирования с возможностью расширенной настройки.

Ниже приведен слегка подправленный классический конфигурационный файл.

Листинг 5: /etc/syslog-ng/syslog-ng.conf

```
options { chain_hostnames(off); sync(0); };

#источник, откуда читать журнал
source src { unix-stream("/dev/log"); internal(); };
source kernsrc { file("/proc/kmsg"); };

#получатели
```

```

destination authlog { file("/var/log/auth.log"); };
destination syslog { file("/var/log/syslog"); };
destination cron { file("/var/log/cron.log"); };
destination daemon { file("/var/log/daemon.log"); };
destination kern { file("/var/log/kern.log"); };
destination lpr { file("/var/log/lpr.log"); };
destination user { file("/var/log/user.log"); };
destination mail { file("/var/log/mail.log"); };

destination mailinfo { file("/var/log/mail.info"); };
destination mailwarn { file("/var/log/mail.warn"); };
destination mailerr { file("/var/log/mail.err"); };

destination newscrit { file("/var/log/news/news.crit"); };
destination newserr { file("/var/log/news/news.err"); };
destination newsnotice { file("/var/log/news/news.notice"); };

destination debug { file("/var/log/debug"); };
destination messages { file("/var/log/messages"); };
destination console { usertty("root"); };
destination console_all { file("/dev/tty12"); };
destination xconsole { pipe("/dev/xconsole"); };

#создание фильтров
filter f_authpriv { facility(auth, authpriv); };
filter f_syslog { not facility(authpriv, mail); };
filter f_cron { facility(cron); };
filter f_daemon { facility(daemon); };
filter f_kern { facility(kern); };
filter f_lpr { facility(lpr); };
filter f_mail { facility(mail); };
filter f_user { facility(user); };
filter f_debug { not facility(auth, authpriv, news, mail); };
filter f_messages { level(info..warn)
    and not facility(auth, authpriv, mail, news); };
filter f_emergency { level(emerg); };

filter f_info { level(info); };
filter f_notice { level(notice); };
filter f_warn { level(warn); };
filter f_crit { level(crit); };
filter f_err { level(err); };
filter f_failed { match("failed"); };
filter f_denied { match("denied"); };

#связывание фильтров с получателями
log { source(src); filter(f_authpriv); destination(authlog); };
log { source(src); filter(f_syslog); destination(syslog); };
log { source(src); filter(f_cron); destination(cron); };
log { source(src); filter(f_daemon); destination(daemon); };
log { source(kernsrc); filter(f_kern); destination(kern); };
log { source(src); filter(f_lpr); destination(lpr); };
log { source(src); filter(f_mail); destination(mail); };
log { source(src); filter(f_user); destination(user); };
log { source(src); filter(f_mail); filter(f_info); destination(mailinfo); };
log { source(src); filter(f_mail); filter(f_warn); destination(mailwarn); };
log { source(src); filter(f_mail); filter(f_err); destination(mailerr); };

log { source(src); filter(f_debug); destination(debug); };
log { source(src); filter(f_messages); destination(messages); };
log { source(src); filter(f_emergency); destination(console); };

#журнал по умолчанию

```

```
log { source(src); destination(console_all); };
```

Syslog-ng очень легко настраивается, но с той же легкостью можно что-либо упустить в файле настройки, так как он очень велик. Автор также обещает дополнительные возможности, например, шифрование, опознание, сжатие и обязательную проверку прав доступа (MAC — mandatory access control). С такими возможностями программа станет незаменимой для журналирования в сети, так как злоумышленник не сможет перехватывать журнал.

К тому же у syslog-ng есть еще одно достоинство: он не требует прав администратора для запуска!

### 3.е. Анализ журналов с помощью Logcheck

Конечно же, ведение журналов — это только полдела. Приложение наподобие Logcheck может значительно облегчить регулярный анализ журналов. Logcheck — это сценарий, который вместе с двоичным файлом logtail, запускается службой cron и сверяет журналы с заданными правилами на предмет подозрительной деятельности. При совпадении он отправляет электронное письмо с соответствующими записями на почтовый ящик администратора.

Logcheck и logtail входят в состав пакета app-admin/logsentry.

Logcheck использует четыре файла для выделения важных событий из обычных сообщений. Этими файлами являются logcheck.hacking, содержащий известные сообщения, встречающиеся при взломе, logcheck.violations, содержащий шаблоны, сигнализирующие о нарушениях безопасности, logcheck.violations.ignore, содержащий ключевые слова, совпадающие с шаблонами в файле нарушений, но которые можно игнорировать, и logcheck.ignore, содержащий записи, которые можно игнорировать.

**Предупреждение:** Не оставляйте logcheck.violations.ignore пустым. Для обработки журналов Logcheck использует grep, некоторые версии которого считают пустой файл знаком подстановки (wildcard). В таком случае все нарушения будут игнорироваться.

## 4. Монтирование разделов

### 4.а. Монтирование разделов

При монтировании разделов с файловой системой ext2, ext3 или reiserfs можно указать различные параметры в файле /etc/fstab. Среди них:

- nosuid — игнорировать установленный для файла бит SUID, уподобляя его обычному файлу
- noexec — предотвращать запуск файлов с данного раздела
- nodev — игнорировать файлы устройств

К сожалению, эти настройки можно легко обойти, задействовав косвенные пути. Тем не менее, установкой noexec для /tmp можно воспрепятствовать использованию большинства вредоносных программ, разработанных для запуска напрямую из каталога /tmp.

#### Листинг 1: /etc/fstab

```
/dev/sda1 /boot ext2 noauto,noatime 1 1
/dev/sda2 none swap sw 0 0
/dev/sda3 / reiserfs notail,noatime 0 0
/dev/sda4 /tmp reiserfs notail,noatime,nodev,nosuid,noexec 0 0
```



```
/dev/sda5 /var reiserfs notail,noatime,nodev 0 0
/dev/sda6 /home reiserfs notail,noatime,nodev,nosuid 0 0
/dev/sda7 /usr reiserfs notail,noatime,nodev,ro 0 0
/dev/cdroms/cdrom0 /mnt/cdrom iso9660 noauto,ro 0 0
proc /proc proc defaults 0 0
```

**Предупреждение:** Установка `/tmp` в режим `noexec` может помешать нормальной работе различных сценариев.

**Примечание:** За сведениями о дисковых квотах обращайтесь к разделу [КВОТЫ](#).

**Примечание:** Я не устанавливаю для `/var` параметры `noexec` или `nosuid`, даже если обычно файлы никогда не запускаются из этого каталога, так как `netqmail` устанавливается в `/var/qmail` и требует разрешения на исполнение и доступ к одному файлу с SUID. Я устанавливаю `/usr` в режим только для чтения, так как я никогда ничего не записываю туда, кроме случая обновления Gentoo. Тогда я перемонтирую файловую систему в режиме чтения-записи, обновляю систему и перемонтирую обратно в режиме только для чтения.

**Примечание:** Даже если вы не используете `netqmail`, Gentoo все же нужен установленный бит исполнения для `/var/tmp`, так поскольку сборка `ebuild` выполняется именно здесь. Но если у вас есть насущная необходимость монтировать каталог `/var` с параметром `noexec`, можно указать для этого альтернативный путь.

## 5. Ограничения пользователей/групп

### 5.a. `/etc/security/limits.conf`

Контроль использования ресурсов может быть очень эффективным при попытке предотвращения локальной атаки на отказ от обслуживания или ограничения максимального количества входов в систему для группы или пользователя. Однако излишне строгие настройки будут затормаживать работу вашей системы и станут причиной сбоев программ, поэтому проверьте каждый из параметров.

#### Листинг 1: `/etc/security/limits.conf`

```
*      soft core 0
*      hard core 0
*      hard nproc 15
*      hard rss 10000
*      -    maxlogins 2
@dev   hard core 100000
@dev   soft nproc 20
@dev   hard nproc 35
@dev   -    maxlogins 10
```

Если вы пытаетесь установить `nproc` или `maxlogins` в 0, то подумайте, может, лучше вместо этого удалить пользователя. В примере выше установлены настройки группы `dev` для процессов, основных файлов и `maxlogins`. Все остальное установлено в значения по умолчанию.

**Примечание:** `/etc/security/limits.conf` является частью пакета `PAM` и применима только для пакетов, использующих `PAM`.

### 5.b. `/etc/limits`

`/etc/limits` очень схож с файлом ограничений `/etc/security/limits.conf`. Единственное отличие — формат, который применим лишь для пользователей или заполнителей

(но не для групп). Вот пример конфигурации:

#### Листинг 2: /etc/limits

```
*      L2 C0 U15 R10000
kn L10 C100000 U35
```

Здесь устанавливаются настройки по умолчанию и специфичные настройки для пользователя kn. limits являются частью пакета sys-apps/shadow. Если вы включили pam в make.conf, то нет необходимости устанавливать какие-либо ограничения в этом файле.

### 5.с. Квоты

**Предупреждение:** Проверьте, что рабочая файловая система поддерживает квоты. Чтобы задействовать квоты для ReiserFS, необходимо наложить заплатку на ядро, которая доступна на сайте [Namesys](http://Namesys). Пользовательские утилиты можно загрузить с сайта проекта [Linux DiskQuota project](http://Linux DiskQuota project). При совместной работе квот с ReiserFS вы можете встретить проблемы. Мы вас предупредили!

Установка квот на файловые разделы ограничивает использование дискового пространства для каждого пользователя или каждой группы. Квоты должны быть включены в ядре и добавлены к точке монтирования в /etc/fstab. Параметр ядра включается в конфигурации ядра в разделе File systems->Quota support. Установите следующие настройки, пересоберите ядро и перезагрузитесь в систему с новым ядром.

Начните с установки квот, запустив emerge quota. Затем измените /etc/fstab и добавьте usrquota и grpquota к разделам, использование которых вы хотите ограничить, так, как показано в примере ниже.

#### Листинг 3: /etc/fstab

```
/dev/sda1 /boot ext2 noauto,noatime 1 1
/dev/sda2 none swap sw 0 0
/dev/sda3 / reiserfs notail,noatime 0 0
/dev/sda4 /tmp ext3 noatime,nodev,nosuid,noexec,usrquota,grpquota 0 0
/dev/sda5 /var ext3 noatime,nodev,usrquota,grpquota 0 0
/dev/sda6 /home ext3 noatime,nodev,nosuid,usrquota,grpquota 0 0
/dev/sda7 /usr reiserfs notail,noatime,nodev,ro 0 0
/dev/cdroms/cdrom0 /mnt/cdrom iso9660 noauto,ro 0 0
proc /proc proc defaults 0 0
```

Создайте для каждого раздела, для которого включены квоты, файлы квот (aquota.user и aquota.group) и поместите их в корень каждого из разделов.

#### Листинг 4: Создание файлов квот

```
# touch /tmp/aquota.user
# touch /tmp/aquota.group
# chmod 600 /tmp/aquota.user
# chmod 600 /tmp/aquota.group
```

Это шаг выполняется для любого раздела, для которого включены квоты. После добавления и настройки файлов квот вам понадобится добавить сценарий запуска quota в загрузочный уровень запуска.

**Важно:** XFS проверяет все квоты самостоятельно, поэтому для нее не требуется добавления сценария quota в начальный уровень загрузки. Есть ряд файловых систем, не указанных в этом документе, которые функционируют также, поэтому обратитесь к странице руководства вашей файловой системы, для того, чтобы

узнать, как она обрабатывает проверки квот.

#### Листинг 5: Добавление квот в загрузочный уровень исполнения

```
# rc-update add quota boot
```

Теперь настроим систему для проверки квот раз в неделю, добавив следующую строку в `/etc/crontab`:

#### Листинг 6: Добавление проверки квот в crontab

```
0 3 * * 0 /usr/sbin/quotacheck -avug.
```

Перезагрузив компьютер, время установить квоты для пользователей и групп. С помощью команды `edquota -u kn` запустится редактор, определенный переменной `$EDITOR` (по умолчанию `nano`) для правки квот для пользователя `kn`. `edquota -g` сделает то же самое для групп.

#### Листинг 7: Установка квот для пользователя kn

```
Quotas for user kn:
/dev/sda4: blocks in use: 2594, limits (soft = 5000, hard = 6500)
          inodes in use: 356, limits (soft = 1000, hard = 1500)
```

Для большей информации обратитесь к `man edquota` или [Quota miniHOWTO](#).

### 5.d. `/etc/login.defs`

Если правилами определено, что пользователи должны менять свои пароли каждые две недели, измените значения переменной `PASS_MAX_DAYS` на 14 и `PASS_WARN_AGE` на 7. Это рекомендуется для того, чтобы предотвратить старение пароля, так как полный перебор всех значений может взломать любой пароль, если есть достаточно времени. Также рекомендуется установить `LOG_OK_LOGINS` в значение `yes`.

### 5.e. `/etc/login.access`

Файл `login.access` также является частью пакета `sys-apps/shadow`, который управляет процессом входа в систему. Он определяет на основании имени пользователя, группы или узла, кто сможет войти в систему, а кто не сможет. Так как по умолчанию этот файл содержит лишь примеры и комментарии, то все пользователи системы могут войти в систему. В зависимости от того, что вы защищаете — сервер или рабочую станцию, рекомендуется настроить этот файл так, чтобы никто, кроме вас (как администратора), не мог получить доступ к консоли.

**Примечание:** Эти настройки не влияют на суперпользователя.

#### Листинг 8: `/etc/login.access`

```
 -:ALL EXCEPT wheel sync:console
 -:wheel:ALL EXCEPT LOCAL .gentoo.org
```

**Важно:** Настраивая эти параметры, будьте осторожны, так как любая ошибка может привести к тому, что вы не сможете войти в собственную систему, не воспользовавшись правами суперпользователя.

**Примечание:** Эти настройки не применимы для SSH, так как по умолчанию для SSH не запускается `/bin/login`. Это может быть включено с помощью параметра `UseLogin yes` в файле `/etc/ssh/sshd_config`.

Это устанавливает доступ на вход в систему так, что только члены группы `wheel`

могут входить в систему локально или из домена gentoo.org. Возможно, попахивает паранойей, но лучше быть в безопасности, чем сожалеть о потерянном.

## 6. Права доступа к файлам

### 6.a. Файлы, доступные по чтению для всех

Обычные пользователи не должны иметь доступ к конфигурационным файлам или паролям. Атакующий может украсть пароли к базе данных или веб-сайту и использовать их для дефейса или даже хуже — для удаления файлов. Вот почему важно правильно установить разрешения на файлы. Если вы уверены, что определенный файл может использоваться только суперпользователем, установите для него права 0600 и назначьте ему правильного владельца с помощью `chown`.

### 6.b. Файлы, доступные по записи для группы или для всех

#### Листинг 1: Поиск файлов, доступных по записи всем

```
# find / -type f \( -perm -2 -o -perm -20 \) -exec ls -lg {} \; 2>/dev/null >writable.txt
# find / -type d \( -perm -2 -o -perm -20 \) -exec ls -ldg {} \; 2>/dev/null >>writable.txt
```

Эти команды создадут огромный файл, содержащий имена файлов, которые имеют права на запись для группы и всех остальных. Проверьте их и уберите ненужные права, запустив для файлов команду `/bin/chmod o-w`.

### 6.c. Файлы SUID/SGID

Файлы с битами SUID или SGID будут запущены с привелегиями владельца или группы владельца, а не с привелегиями запустившего их пользователя. Обычно это используется для файлов, которые должны запускаться с правами суперпользователя для выполнения необходимых задач. Эти файлы, если содержат уязвимости, могут стать источником повышения привелегий локальным пользователем. Это может быть опасным, поэтому с подобных файлов необходимо любой ценой удалить установленный бит SUID или SGID. Если вы не используете эти файлы, запустите для них команду `chmod 0` или удалите пакет, установивший этот файл (проверить, какому пакету принадлежит определенный файл, можно с помощью команды `equery`; если у вас ее еще нет, то, чтобы установить ее, просто наберите `emerge gentoolkit`). Иначе просто удалите бит SUID с помощью `chmod -s`.

#### Листинг 2: Поиск файлов с setuid

```
# find / -type f \( -perm -004000 -o -perm -002000 \) -exec ls -lg {} \; 2>/dev/null >suicid
```

Эта команда создаст файл, содержащий список всех SUID/SGID-файлов.

#### Листинг 3: Список файлов с setuid

```
/bin/su
/bin/ping
/bin/mount
/bin/umount
/var/qmail/bin/qmail-queue
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/crontab
/usr/bin/chage
/usr/bin/expiry
/usr/bin/sperl5.6.1
```

```
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/procmail
/usr/bin/suidperl
/usr/lib/misc/pt_chown
/usr/sbin/unix_chkpwd
/usr/sbin/traceroute
/usr/sbin/pwdb_chkpwd
```

По умолчанию в Gentoo Linux не так много файлов с битом SUID (хотя это зависит от типа вашей установки), но у вас может получиться список, представленный выше. Большинство команд используется только суперпользователем. Удалите SUID-бит с `ping`, `mount`, `umount`, `chfn`, `chsh`, `newgrp`, `suidperl`, `pt_chown` и `traceroute`, запустив для каждого из них `chmod -s`. Не удаляйте бит с `su`, `qmail-queue` или `unix_chkpwd`. Удалив `setuid` с этих файлов, вы не сможете стать суперпользователем и получать почту. Удаляя бит (там, где это безопасно), вы исключаете возможность обычного пользователя (или злоумышленника) получить права суперпользователя с помощью этих файлов.

В моей системе есть только несколько SUID-файлов: `su`, `passwd`, `gpasswd`, `qmail-queue`, `unix_chkpwd` и `pwdb_chkpwd`. Но если у вас запущен X-сервер, то в вашей системе их будет больше, так как X нужны повышенные привелегии, которые могут быть предоставлены посредством SUID.

#### 6.d. SUID/SGID-файлы и жесткие ссылки

Файл может считаться удаленным лишь в том случае, когда нет больше ссылок, указывающих на него. Это может звучать странно, но просто следует понять, что файл (например, `/usr/bin/perl`) на самом деле является ссылкой на `inode`, в котором сохранены данные. На файл может указывать любое число ссылок, и пока каждая из них не будет удалена, файл будет существовать.

Если у ваших пользователей есть доступ к разделам, которые не смонтированы с параметрами `nosuid` или `noexec` (например, если `/tmp`, `/home`, `/var/tmp` не являются отдельными разделами), вы должны удостовериться, что они не смогут создать жесткие ссылки на файлы с SUID- или SGID-битами, благодаря чему они могут иметь доступ к устаревшим версиям файлов.

**Предупреждение:** Если вы получаете предупреждения об оставшихся жестких ссылках, а ваши пользователи имеют доступ по записи к разделам, позволяющим запускать файлы с SUID/SGID-битами, вы должны прочитать этот раздел очень внимательно. Любой из ваших пользователей может разрушить ваши обновления, сохранив устаревшую версию программы. Если ваши пользователи не могут создавать собственные файлы с битом SUID или могут только запускать программы с помощью динамического загрузчика (разделы, смонтированные с параметром `noexec`), то вам не о чем беспокоиться.

**Примечание:** Пользователям не нужен доступ по чтению для файла, чтобы создать на него ссылку, им нужен всего лишь доступ по чтению к каталогу, содержащему этот файл.

Чтобы проверить, сколько ссылок имеет файл, вы можете использовать команду `stat`.

#### Листинг 4: Команда stat

```
$ stat /bin/su
File: `/bin/su'
Size: 29350          Blocks: 64          IO Block: 131072 regular file
```

```
Device: 900h/2304d      Inode: 2057419      Links: 1
Access: (4711/-rws--x--x)  Uid: (    0/      root)  Gid: (    0/      root)
Access: 2005-02-07 01:59:35.000000000 +0000
Modify: 2004-11-04 01:46:17.000000000 +0000
Change: 2004-11-04 01:46:17.000000000 +0000
```

Чтобы найти файлы с SUID и SGID со множеством ссылок, вы можете задействовать команду `find`.

#### Листинг 5: Поиск suid/sgid-файлов с несколькими ссылками

```
$ find / -type f \( -perm -004000 -o -perm -002000 \) -links +1 -ls
```

## 7. PAM

### 7.a. PAM

PAM — это набор библиотек, предоставляющих альтернативный способ аутентификации пользователя в программах. USE-флаг `pam` уже включен по умолчанию. Хотя настройки PAM в Gentoo Linux довольно разумны, всегда есть возможность что-нибудь улучшить. Для начала установите `cracklib`.

#### Листинг 1: Установка cracklib

```
# emerge cracklib
```

#### Листинг 2: /etc/pam.d/passwd

```
auth      required pam_unix.so shadow nullok
account   required pam_unix.so
password  required pam_cracklib.so difok=3 retry=3 minlen=8 dcredit=2 ocredit=2
password  required pam_unix.so md5 use_authtok
session   required pam_unix.so
```

После этого с помощью `cracklib` будет проверяться пароль пользователя, имеет ли он длину по крайней мере 8 символов, содержит ли он по крайней мере 2 цифры, 2 других символа и не менее 3 символов, отличающихся от символов прошлого пароля. Благодаря этому пользователь будет выбирать стойкие пароли. Проверьте [документацию PAM](#) для других параметров.

#### Листинг 3: /etc/pam.d/sshd

```
auth      required pam_unix.so nullok
auth      required pam_shells.so
auth      required pam_nologin.so
auth      required pam_env.so
account   required pam_unix.so
password  required pam_cracklib.so difok=3 retry=3 minlen=8 dcredit=2 ocredit=2 use_authtok
password  required pam_unix.so shadow md5
session   required pam_unix.so
session   required pam_limits.so
```

Любая служба, не описанная в одном из файлов PAM из каталога `/etc/pam.d`, будет использовать правила, взятые из `/etc/pam.d/other`. По умолчанию установлена политика `deny`, как и должно быть. Но нам нужна регистрация, поэтому добавим `pam_warn.so`. Последняя конфигурация — это `pam_limits`, которая управляется с помощью `/etc/security/limits.conf`. См. раздел [/etc/security/limits.conf](#) для подробного описания этих настроек.

#### Листинг 4: /etc/pam.d/other

```
auth      required pam_deny.so
auth      required pam_warn.so
account   required pam_deny.so
account   required pam_warn.so
password  required pam_deny.so
password  required pam_warn.so
session   required pam_deny.so
session   required pam_warn.so
```

## 8. Упаковщики TCP

### 8.a. Упаковщики TCP

Обычно для контроля доступом к службам используется `inetd` (которого в Gentoo нет), но так же можно воспользоваться `xinetd` или другими сервисами.

**Примечание:** Служба должна быть запущена с помощью `tcpd` в качестве аргумента сервера (для случая `xinetd`). См. раздел о `xinetd` для большей информации.

#### Листинг 1: `/etc/hosts.deny`

```
ALL:PARANOID
```

#### Листинг 2: `/etc/hosts.allow`

```
ALL: LOCAL @wheel
time: LOCAL, .gentoo.org
```

Как вы видите, формат записи очень похож на содержимое файла `/etc/login.access`. `Tcpd` поддерживает отдельная служба; она никак не связана с `/etc/login.access`. Эти настройки применимы только для служб, использующих упаковщики TCP.

Также возможно запускать команды при разрешенной службе (это может быть полезным для создания трансляций для коммутируемых пользователей), но это не рекомендуется, так как обычно, решая одну проблему, люди создают еще больше. Допустим, что вы настроили сценарий, отправляющий электронное письмо в том случае, когда кто-то подпал под запрещающее правило, но тогда злоумышленник может запустить DoS-атаку, попадающую под правило запрещения. Из-за этого создастся много ввода-вывода и ненужно электронной почты, поэтому не делайте так! Прочтите `man 5 hosts_access` для дальнейшей информации.

## 9. Безопасность ядра

### 9.a. Удаление функциональности

Основное правило при конфигурации ядра — удалять все ненужное. Это не только создаст маленькое ядро, но и также исключит вероятные уязвимости, которые могут присутствовать в драйверах и других возможностях.

Также рекомендуется отключить поддержку загружаемых модулей. Хотя возможность внедрения руткитов сохраняется, это делает систему более неприступной для обычного злоумышленника, пытающегося установить руткит в виде модуля ядра.



## 9.b. Файловая система *proc*

Множество параметров ядра может быть изменено с помощью файловой системы */proc* или *sysctl*.

Чтобы иметь возможность на лету изменять параметры и переменные ядра, вам понадобится определить в ядре *CONFIG\_SYSCTL*. Она включена по умолчанию в ядрах серии 2.4.

### Листинг 1: Отключения IP-форвардинга

```
# /bin/echo "0" > /proc/sys/net/ipv4/ip_forward
```

Проверьте, что проброс IP-пакетов отключен. Эта возможность необходима лишь для узла, имеющего доступ к нескольким сетям. Рекомендуется устанавливать и отключать этот флаг до включения или отключения других флагов.

### Листинг 2: Отбрасывание пакетов ping

```
# /bin/echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

Это просто заставит ядро игнорировать все сообщения ping (ICMP-пакеты типа 0). IP-пакет, несущий ICMP-сообщение, может содержать также в нагрузку и другую информацию, о которой вы можете не подозревать, поэтому следует отключить прием. Администраторы используют ping как утилиту диагностики и часто выражают недовольство, если она отключена, но нет причины позволять чужакам пинговать узел. Тем не менее, если необходимо разрешить внутренним пользователям использовать ping, то можно отключить сообщения ICMP типа 0 в межсетевом экране (тем самым позволив локальным администраторам использовать эту утилиту).

### Листинг 3: Игнорирование широковещательных запросов ping

```
# /bin/echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

Это отключает ответ на широковещательные ICMP-запросы, предотвращая реализацию Smurf-атаки. Smurf-атака основана на отправке ICMP-пакета типа 0 (ping) по широковещательному адресу сети. Обычно для этого злоумышленник использует поддельный адрес. Все компьютеры сети ответят на сообщение ping, что может вызвать наводнение трафика на узел, адрес которого был подделан.

### Листинг 4: Отключение исходящих маршрутизированных пакетов.

```
# /bin/echo "0" > /proc/sys/net/ipv4/conf/all/accept_source_route
```

Не принимать исходящие маршрутизированные пакеты. Злоумышленники могут использовать исходящую маршрутизацию для генерации трафика, имеющего адрес внутренней сети, но на самом деле возвращаемого назад злоумышленнику, что позволит ему компрометировать сеть. Исходящая маршрутизация очень редко используется по назначению, так что безопаснее ее отключить.

### Листинг 5: Отключение приема перенаправлений

```
# /bin/echo "0" > /proc/sys/net/ipv4/conf/all/accept_redirects  
# /bin/echo "0" > /proc/sys/net/ipv4/conf/all/secure_redirects
```

Не принимать ICMP-пакеты перенаправления. ICMP-перенаправления могут быть использованы злоумышленником для изменения таблиц маршрутизации.



#### Листинг 6: Защита против неправильных сообщений об ошибках

```
# /bin/echo "1" > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
```

Включить защиту против приходящих неправильных сообщений об ошибках.

#### Листинг 7: Включение фильтрации обратного пути

```
# for i in /proc/sys/net/ipv4/conf/*; do
    /bin/echo "1" > $i/rp_filter
done
```

Включает фильтрацию обратного пути. Это поможет быть уверенным, что в пакетах используются правомерные адреса, при этом входящие пакеты будут автоматически отброшены, если запись в таблице маршрутизации об адресе этих пакетов не соответствует сетевому интерфейсу, с которого они пришли. Это позволит предотвратить подделку IP-адресов. Необходимо включить эту возможность для каждого `net/ipv4/conf/*`, иначе проверка исходящих пакетов не будет полностью функционировать.

**Предупреждение:** Тем не менее включение фильтрации обратного пути может стать проблемой при использовании асимметричной маршрутизации (исходящие пакеты идут отличным от входящих пакетов путем) или при использовании немаршрутизируемом узле с несколькими IP-адресами на различных интерфейсах.

#### Листинг 8: Регистрация всех поддельных, исходящих маршрутизированных и перенаправленных пакетов

```
# /bin/echo "1" > /proc/sys/net/ipv4/conf/all/log_martians
```

Регистрировать поддельные пакеты, пакеты с исходящей маршрутизацией и пакеты перенаправлений.

Все настройки будут сброшены после перезагрузки компьютера. Рекомендуется добавлять их в `/etc/sysctl.conf`, который будет автоматически загружен сценарием инициализации `/etc/init.d/bootmisc`.

Синтаксис `/etc/sysctl.conf` достаточно понятный. Удалите из указанных путей `/proc/sys/` и замените `/` на `..`

#### Листинг 9: Переход к файлу `sysctl.conf`

```
(Вручную с помощью команды echo:)
/bin/echo "0" > /proc/sys/net/ipv4/ip_forward
```

```
(Автоматически в файле sysctl.conf:)
net.ipv4.ip_forward = 0
```

### 9.c. Grsecurity

Заплата от [Grsecurity](#) входит в `sys-kernel/hardened-sources`, но по умолчанию отключена. Сначала сконфигурируйте ядро, а затем настройте параметры Grsecurity. Подробное описание доступных параметров Grsecurity можно найти на странице проекта [укрепленного Gentoo](#).

Последние версии `hardened-sources` содержат Grsecurity версии 2.\*. Для дополнительной информации о наборе патчей Grsecurity обратитесь к документации, доступной на [домашнем сайте Grsecurity](#).

### 9.d. Kerneli

[Kerneli](#) — это заплатка, которая добавляет функции шифрования к существующему ядру. Наложив эту заплатку на ядро, вы получите такие новые параметры, как алгоритмы шифрования, дайджеста и фильтры к зашифрованным образам.

**Предупреждение:** В настоящий момент заплатка kerneli для последнего ядра нестабильна, поэтому будьте осторожны при ее использовании.

### 9.e. Другие заплатки к ядру

- [Проект OpenWall](#)
- [Linux Intrusion Detection System](#)
- [Rule Set Based Access Control](#)
- [NSA's security enhanced kernel](#)
- [Wolk](#)

И множество других.

## 10. Безопасность служб

### 10.a. Apache

Apache поставляется с достаточно полным конфигурационным файлом по умолчанию, но опять же необходимо улучшить некоторые аспекты, например, использовать для Apache только один адрес. Ниже приведены параметры, которые вы должны внести в конфигурационный файл.

Если вы не отключали поддержку ssl в /etc/make.conf перед установкой Apache, то теперь вы должны иметь доступ к серверу с включенным SSL. Чтобы включить ее, просто добавьте следующую строку.

#### Листинг 1: /etc/conf.d/apache

```
HTTPD_OPTS="-D SSL"
```

#### Листинг 2: /etc/apache/conf/apache.conf

```
#Добавьте свой IP для прослушивания
Listen 127.0.0.1
BindAddress 127.0.0.1
#Не следует использовать nobody или nogroup для каждого сервиса, запущенного
#без административных привелегий
#(просто добавляем пользователя apache с группой apache)
User apache
Group apache
#Не сообщать информацию о версии сервера
ServerSignature Off
ServerTokens Prod
```

Apache собран с параметрами --enable-shared=max и --enable-module=all. По умолчанию это включает все модули, поэтому вы должны закомментировать все неиспользуемые модули в разделе LoadModule (LoadModule и AddModule). Затем перезапустите сервис, выполнив /etc/init.d/apache restart.

Документация доступна по адресу <http://www.apache.org>.

### 10.b. Bind

Документацию можно найти на сайте [Internet Software Consortium](#). Руководство

администратора BIND 9 также доступно каталоге doc/arm.

Новые сборочные файлы BIND поддерживают изменение корневого каталога из коробки. После установки bind следуйте этим простым инструкциям:

### Листинг 3: Изолирование среды BIND

```
# emerge --config bind
(Перед запуском следующей команды вам может потребоваться изменить каталог chroot в файле /etc/conf.d/named. Иначе будет использован /chroot/dns.)
```

### 10.c. Djbdns

Djbdns — это реализация DNS, за обнаружение уязвимостей в которой автор готов выплачивать деньги. Принцип работы сильно отличается от Bind 9, однако он работает. Дополнительная информация может быть получена на сайте <http://www.djbdns.org>.

### 10.d. FTP

В общем случае использование FTP (File Transfer Protocol, протокол передачи файлов) является плохой идеей. Этот протокол отправляет данные незашифрованными (в том числе и пароли), прослушивает 2 порта (обычно это порты 20 и 21), и часто злоумышленники ищут анонимный доступ для обмена вarezом. Так как протокол содержит ряд проблем безопасности, то следует использовать sftp или HTTP. Если это невозможно, то максимально обезопасьте свои сервисы и будьте готовы.

### 10.e. Mysql

Если вам необходимо предоставить доступ к базе данных mysql локальным приложениям, раскомментируйте следующую строку в /etc/mysql/my.cnf.

### Листинг 4: Отключение доступа к сети

```
skip-networking
```

Затем мы отключаем использование команды LOAD DATA LOCAL INFILE. Это предотвратит несанкционированное чтение локальных файлов. Это также подходит против SQL-инъекций в PHP-сценариях.

### Листинг 5: Отключение LOAD DATA LOCAL INFILE в разделе [mysqld]

```
set-variable=local-infile=0
```

Затем необходимо удалить тестовую базу данных (test) и все учетные записи за исключением root.

### Листинг 6: Удаление тестовой базы данных и всех ненужных пользователей

```
mysql> drop database test;
mysql> use mysql;
mysql> delete from db;
mysql> delete from user where not (host="localhost" and user="root");
mysql> flush privileges;
```

**Предупреждение:** Если вас уже есть настроенные учетные записи, то будьте осторожны с этими командами.

**Примечание:** Если вы изменяли пароли в командной строке MySQL, то всегда

очищайте ~/.mysql\_history и /var/log/mysql/mysql.log, так как они сохраняют список выполненных SQL-команд с открытыми паролями.

### 10.f. Proftpd

У proftpd было несколько проблем с безопасностью, но большинство из них было исправлено. Тем не менее, неплохо внести некоторые улучшения:

#### Листинг 7: /etc/proftpd/proftpd.conf

```
ServerName "My ftp daemon"
# Не показывать ident сервера
ServerIdent on "Go away"

# Упрощаем создание виртуальных пользователей
RequireValidShell off

# Использовать альтернативные файлы паролей и групп (в формате passwd)
AuthUserFile "/etc/proftpd/passwd"
AuthGroupFile "/etc/proftpd/group"

# Разрешения
Umask 077

# Таймауты и ограничения
MaxInstances 30
MaxClients 10 "Only 10 connections allowed"
MaxClientsPerHost 1 "You have already logged on once"
MaxClientsPerUser 1 "You have already logged on once"
TimeoutStalled 10
TimeoutNoTransfer 20
TimeoutLogin 20

# Все входят в изолированную оболочку
DefaultRoot ~

# Не запускать с правами администратора
User nobody
Group nogroup

# Регистрировать любые передачи
TransferLog /var/log/transferlog

# Проблемы с универсализацией имен файлов
DenyFilter \*.*/*
```

Дополнительная информация может быть найдена на <http://www.proftpd.org>.

### 10.g. Pure-ftpd

Pure-ftpd является ответвлением оригинального trolldpd с модификациями Фрэнка Денниса по части безопасности и функциональности.

Используйте виртуальных пользователей (и никогда не применяйте системные учетные записи), включив параметр AUTH. Установите его для -lpuredb:/etc/pureftpd.pdb и создайте пользователей при помощи команды /usr/bin/pure-pw.

#### Листинг 8: /etc/conf.d/pure-ftpd

```
AUTH="-lpuredb:/etc/pureftpd.pdb"
```

```
## Misc. Others ##  
MISC_OTHER="-A -E -X -U 177:077 -d -4 -L100:5 -I 15"
```

Добавьте к переменной MISC\_OTHER параметры запрета на подключение анонимных пользователей (-E), изменения корневого каталога для всех (-A), запрета записи и чтения файлов, начинающихся с точки (-X), максимального периода бездействия (-I), ограничения рекурсии (-L) и подходящего umask.

**Предупреждение:** Не используйте параметры -w или -W! Если же вы хотите, чтобы у вас был сайт с врезом, то перестаньте читать это руководство!

Дополнительная информация может быть найдена на <http://www.pureftpd.org>.

### 10.h. Vsftpd

Vsftpd (сокращение от very secure ftp) — это небольшой FTP-демон, который может запускаться с настройками по умолчанию. Он прост и не имеет множества возможностей, присущих pureftpd и proftpd.

#### Листинг 9: /etc/vsftpd

```
anonymous_enable=NO  
local_enable=YES  
  
#read only  
write_enable=NO  
  
#enable logging of transfers  
xferlog_std_format=YES  
  
idle_session_timeout=20  
data_connection_timeout=20  
nopriv_user=nobody  
  
chroot_list_enable=YES  
chroot_list_file=/etc/vsftpd/chrootlist  
  
ls_recurse_enable=NO
```

Как вы можете видеть, нет возможности установить для данного сервиса индивидуальные разрешения, но если используется настройки для анонимного доступа, то это он не так уж плох. Иногда неплохо иметь анонимный FTP-сервер (для доступа к открытому ПО), и vsftpd неплохо подходит для этой работы.

### 10.i. Netqmail

Netqmail часто признается наиболее безопасным почтовым сервером. Он написан с прицелом на безопасность (и паранойю). По умолчанию он не разрешает релей и, начиная с 1996 года, не имеет никаких уязвимостей безопасности. Просто наберите emerge netqmail и настройте его!

### 10.j. Samba

Samba — это протокол, предоставляющий файлы в сетях Microsoft/Novell, и он не должен использоваться в интернете. Его необходимо обезопасить.

#### Листинг 10: /etc/samba/smb.conf

```
[global]  
#Прослушивать определенный интерфейс  
interfaces = eth0 10.0.0.1/32
```

```

#Использовать зашифрованные пароли
encrypt passwords = yes
directory security mask = 0700

#Разрешить трафик из подсети 10.0.0.*
hosts allow = 10.0.0.

#Включить аутентификацию
#(не используйте режим share)
security = user

#Запретить привелегированные учетные записи
invalid users = root @wheel

#Максимальный отображаемый размер ресурса (не является ограничителем)
max disk size = 102400

#Ужесточить политики паролей
min password length = 8
null passwords = no

#Использовать PAM (если была добавлена поддержка)
obey pam restrictions = yes
pam password change = yes

```

Проверьте, что для каждого ресурса выставлены правильные разрешения и не забудьте прочитать [документацию](#).

Теперь перезапустите сервер и добавьте пользователей, которым необходим доступ к этому сервису. Это можно сделать с помощью команды `/usr/bin/smbpasswd` с параметром `-a`.

## 10.k. ssh

Для OpenSSH необходимо лишь одно улучшение — использование более надежного механизма аутентификации, основанного на шифровании с использованием открытого ключа. Очень много сайтов (например, <http://www.sourceforge.net>, <http://www.php.net> и <http://www.apache.org>) было подвержено несанкционированному доступу из-за пустых или слабых паролей.

### Листинг 11: `/etc/ssh/sshd_config`

```

#Включаем только версию 2
Protocol 2

#Отключаем вход администратора. Пользователи могут использовать su для входа
PermitRootLogin no

#Включаем аутентификацию по открытому ключу
PubkeyAuthentication yes
AuthorizedKeysFile      .ssh/authorized_keys

#Отключаем .rhost и обычную аутентификацию по паролю
HostbasedAuthentication no
PasswordAuthentication no
PermitEmptyPasswords no

#Разрешаем подключаться только пользователям из групп wheel или admin
AllowGroups wheel admin

#Из этих групп разрешаем подключаться только следующим пользователям

```

```
#@<domainname> не обязательна, но заменяет старую директиву
#AllowHosts
AllowUsers kn@gentoo.org bs@gentoo.org

#Вход в систему
SyslogFacility AUTH
LogLevel INFO

(Измените адрес на свой)
ListenAddress 127.0.0.1
```

Также проверьте, что в конфигурационном файле нет параметра `UsePAM yes`, который перекрывает механизм аутентификации с помощью открытого ключа.

Теперь создайте для каждого пользователя ключ (на компьютере, с которого они будут регистрироваться) с помощью следующей команды:

#### Листинг 12: Создание пар ключей DSA

```
# /usr/bin/ssh-keygen -t dsa
```

И введите пароль.

#### Листинг 13: Вывод ssh-keygen

```
Generating public/private dsa key pair.
Enter file in which to save the key (/home/kn/.ssh/id_dsa): [Press enter]
Created directory '/home/kn/.ssh'.
Enter passphrase (empty for no passphrase): [Enter passphrase]
Enter same passphrase again: [Enter passphrase again]
Your identification has been saved in /home/kn/.ssh/id_dsa.
Your public key has been saved in /home/kn/.ssh/id_dsa.pub.
The key fingerprint is:
07:24:a9:12:7f:83:7e:af:b8:1f:89:a3:48:29:e2:a4 kn@knielsen
```

После этого в каталоге `~/.ssh/` появятся два файла с названием `id_dsa` и `id_dsa.pub`. Файл `id_dsa` является секретным ключом и должен оберегаться от других. Другой файл, `id_dsa.pub`, должен быть размещен на каждом сервере, к которому необходим доступ. Добавьте этот ключ в каталог `~/.ssh/authorized_keys` в домашнем каталоге пользователя. Теперь пользователь должен иметь доступ к подключению:

#### Листинг 14: Adding the id\_dsa.pub file to the authorized\_keys file

```
$ scp id_dsa.pub other-host:/var/tmp/currenthostname.pub
$ ssh other-host
password:
$ cat /var/tmp/currenthostname.pub >> ~/.ssh/authorized_keys
```

Теперь ваши пользователи должны тщательно оберегать свои секретные ключи. Поместите их на сменный носитель, который они будут всегда носить с собой, или спрячьте их на рабочих станциях пользователей (поместите их под [правила паролей](#)).

Для дальнейшей информации посетите веб-сайт [OpenSSH](#).

### 10.1. Использование xinetd

`xinetd` является заменой `inetd` (которого в Gentoo нет), демона сетевых сервисов. Он поддерживает контроль доступа, основанный на адресе удаленного узла и времени доступа. Он также предоставляет расширенные возможности регистрации событий, включая время запуска сервера, адрес удаленного узла, имя удаленного

пользователя, время работы сервера и выполненные запросы.

Как в случае с другими сервисами, важно создать хорошую конфигурацию по умолчанию. Но так как xinetd запускается с правами администратора и поддерживает протоколы, о работе которых вы можете не знать, рекомендуется не использовать его. Но если вы все равно хотите использовать его, то вы можете укрепить его безопасность:

#### Листинг 15: Установка xinetd

```
# emerge xinetd tcp-wrappers
```

И отредактируйте конфигурационный файл:

#### Листинг 16: /etc/xinetd.conf

```
defaults
{
    only_from = localhost
    instances = 10
    log_type = SYSLOG authpriv info
    log_on_success = HOST PID
    log_on_failure = HOST
    cps = 25 30
}

# Это установит работу pserver (cvs) через xinetd со следующими:
# настройками:
# максимум 10 обработчиков (10 подключений одновременно)
# использовать только TCP
# использовать пользователя cvs для запуска сервиса
# использовать только 1 IP
# разрешать доступ с 10.0.0.*
# разработчики могут пользоваться cvs только с 8 до 17 часов
# использовать упаковщики tpcd (контроль доступа через
# /etc/hosts.allow и /etc/hosts.deny)
# max_load для компьютера 1.0
# Отключающий флаг для настроек по умолчанию, я предпочитаю,
# чтобы он был отключен
service cvspserver
{
    socket_type = stream
    protocol = tcp
    instances = 10
    protocol = tcp
    wait = no
    user = cvs
    bind = 10.0.0.2
    only_from = 10.0.0.0
    access_times = 8:00-17:00
    server = /usr/sbin/tcpd
    server_args = /usr/bin/cvs --allow-root=/mnt/cvsdisk/cvsroot pserver
    max_load = 1.0
    log_on_failure += RECORD
    disable = no
}
```

Для большей информации см. `man 5 xinetd.conf`.

## 10.m. X

По умолчанию Xorg настроен функционировать в качестве Xserver. Это может быть опасным, так как X использует незашифрованные TCP-соединения и прослушивает



подключения xclient.

**Важно:** Если вам не нужен этот сервис, отключите его!

Но если вам требуется использовать рабочую станцию в качестве X-сервера, используйте команду `/usr/X11R6/bin/xhost` с большой осторожностью. Эта команда позволит клиентам других узлов подключаться к вашему экрану. Это может быть полезно при необходимости запуска приложения X с другого компьютера, и это возможно лишь через сеть, но это может быть использовано и злоумышленником. Синтаксис этой команды — `/usr/X11R6/bin/xhost +hostname`.

**Предупреждение:** Не используйте возможность `xhost +!` Эта возможность позволит любому клиенту подключиться к X-серверу и взять контроль над ним. Если злоумышленник может получить доступ к X, то сможет регистрировать нажатия на клавиши, что позволит ему взять контроль над вашим компьютером. Если вы все же используете ее, то всегда указывайте узел.

Более безопасным решением является полное отключение этой возможности при старте X-сервера с помощью `startx -- -nolisten tcp` или отключение ее навсегда в файлах конфигурации.

Листинг 17: `/usr/X11R6/bin/startx`

```
defaultserverargs="-nolisten tcp"
```

Вы должны защитить `startx`, чтобы не допустить его перезаписи при установке новой версии Xorg. Добавьте следующую строку в `/etc/make.conf`:

Листинг 18: `/etc/make.conf`

```
CONFIG_PROTECT_MASK="/usr/X11R6/bin/startx"
```

Если вы используете графический диспетчер входа в систему, вам понадобится другой подход.

Для gdm (Gnome Display Manager)

Листинг 19: `/etc/X11/gdm/gdm.conf`

```
[server-Standard]
command=/usr/X11R6/bin/X -nolisten tcp
```

Для xdm (X Display Manager) и kdm (Kde Display Manager)

Листинг 20: `/etc/X11/xdm/Xservers`

```
:0 local /usr/bin/X11/X -nolisten tcp
```

## 11. Изменение корневого каталога и виртуальные серверы

### 11.а. Изменение корневого каталога

Изменение корневого каталога для службы является способом ограничения среды службы (или пользователя), в которой она имеет доступ лишь к необходимым ресурсам. Запущенная служба под пользователем, отличным от суперпользователя (`nobody`, `apache`, `named`) может предоставить злоумышленнику доступ лишь к тем файлам, к которым имеет доступ пользователь, от имени которого запущена служба. Это означает, что злоумышленник не сможет получить права `root`, даже если служба подвержена различным уязвимостям.

Некоторые службы, например `pure-ftpd` и `bind`, могут быть заключены в `chroot`. Если

служба поддерживает эту возможность, используйте ее, иначе вы можете можете создать среду собственноручно. Давайте рассмотрим пример создания chroot. Чтобы изучить работу механизма chroot, мы будем экспериментировать с bash (простейший для изучения случай).

Создайте каталог /chroot, выполнив команду `mkdir /chroot`. Затем определите, какие динамические библиотеки необходимы для работы bash (если он собран с параметром `-static`, этот шаг можно пропустить):

Следующая команда создаст список библиотек, необходимых для bash.

#### Листинг 1: Получение списка используемых библиотек

```
# ldd /bin/bash
libncurses.so.5 => /lib/libncurses.so.5 (0x4001b000)
libdl.so.2 => /lib/libdl.so.2 (0x40060000)
libc.so.6 => /lib/libc.so.6 (0x40063000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Теперь создадим изолированную среду для bash.

#### Листинг 2: Создание изолированной среды для bash

```
# mkdir /chroot/bash
# mkdir /chroot/bash/bin
# mkdir /chroot/bash/lib
```

Затем скопируем файлы, используемые bash (/lib) в изолированный lib и скопируем файл bash в изолированный каталог bin. Этим мы создадим ту же самую среду, но с ограниченными возможностями. После этого попробуйте выполнить команду `chroot /chroot/bash /bin/bash`. Если вы получите строку приглашения, гласящую `/`, то у вас все получилось! Иначе вам сообщат, какого файла не хватает. Некоторые разделяемые библиотеки могут использовать другие файлы.

Как вы можете заметить, внутри изолированной среды ничего, кроме `echo`, не работает. Это потому что внутри среды chroot кроме bash нет никаких других команд, а `echo` является встроенной командой.

Подобным образом вы можете создать службу в изолированной среде. Единственное различие в том, что служба может время от времени обращаться к устройствам и файлам настроек в /etc. Просто скопируйте их (файлы устройств можно скопировать с помощью команды `cp -a`) в изолированную среду, отредактируйте сценарий инициализации перед тем, как заключить службу в среду. Может быть сложным определить, какие устройства и файлы конфигурации могут понадобиться. Здесь может пригодиться команда `strace`. Запустите службу вместе с `/usr/bin/strace` и отследите все вызовы `open`, `read`, `stat` и, возможно, `connect`. Это подскажет вам, что нужно копировать. Но в большинстве случаев вам понадобится скопировать файл `passwd` (отредактируйте копию, удалив из него всех пользователей, не нужных для запуска службы), `/dev/zero`, `/dev/log` и `/dev/random`.

### 11.b. Пользовательский режим Linux

Другим способом создания безопасной среды является запуск виртуальной машины. Виртуальная машина — это процесс, выглядящий как отдельная ОС и работающий в реальной операционной системе, которая предоставляет ему необходимые системные ресурсы. Безопасность достигается в том, что если сервер, запущенный внутри виртуальной машины, будет взломан, то будет затронут только виртуальный сервер, а не родительская установка.

Для дальнейшей информации по установке пользовательского режима Linux,

обратитесь к [руководству по пользовательскому режиму Linux](#).

## 12. Межсетевые экраны

### 12.a. Межсетевой экран

Люди часто думают, что межсетевой экран окончательно защитит их систему, но они ошибаются. В большинстве случаев плохо настроенный межсетевой экран предоставляет еще меньше безопасен, чем его отсутствие. Он тоже является программой, и с ним нужно обращаться так же, как и с остальными программами, так как он может содержать уязвимости.

А теперь подумайте перед реализацией межсетевого экрана! Так ли он вам нужен? Если да, то вам нужно написать правила, определяющие его работу, тип межсетевого экрана и кто должен им управлять. Но сначала прочитайте это руководство.

Межсетевые экраны используются для двух целей:

- Для защиты извне (черви/злоумышленники)
- Для защиты изнутри (сотрудники/дети)

В основном есть три типа межсетевых экранов:

- Фильтрация пакетов
- Прокси
- Программный шлюз

Межсетевой экран должен работать на выделенном компьютере без лишних запущенных сервисов (или же только с sshd) и безопасен теми способами, которые описаны в этом руководстве.

### 12.b. Фильтрация пакетов

Весь сетевой трафик передается в виде пакетов. Непрерывный поток разделяется на маленькие, легко управляемые пакеты, которые собираются в точке назначения. В заголовке каждого пакета содержится информация о способе и месте его назначения. Также эта информация используется работы межсетевого экрана. Фильтрация основана на:

- разрешении или запрещении пакетов на основании IP-адреса отправителя/получателя
- разрешении или запрещении пакетов на основании порта отправителя/получателя
- разрешении или запрещении пакетов на основании протокола
- разрешении или запрещении пакетов на основании параметров, специфичных для каждого из протоколов

Другими словами, фильтрация основана на содержимом заголовка пакета, а не самого пакета.

Недостатки:

- адрес в пакете может быть ненастоящим адресом IP (или подделанным отправителем)
- данные или запросы в пропущенном пакете могут содержать непредвиденные данные, которые злоумышленник может использовать для эксплуатации известных уязвимостей в сервисах, расположенным на межсетевом экране или позади него

- критично к сбоям

Преимущества:

- простая реализация
- возможность отправки предупреждений о возможных атаках до того, как они произойдут (то есть регистрация сканирования портов)
- хорош против предотвращения SYN-атак

Вот примеры свободных фильтров пакетов для Linux:

- [Iptables](#)
- [Ipchains](#)
- [SmoothWall](#)

**Примечание:** Рекомендуется использовать iptables. ipchains устарел.

### **12.c. Прокси**

Шлюз сеансового уровня является межсетевым экраном, проверяющим соединения перед разрешением обмена данных. Это значит, что он не только разрешает или запрещает пересылку пакетов на основании их заголовка, но и определяет перед открытием сессии на основании настраиваемых правил, что оба адресата реально существуют. Фильтрация основывается на:

- IP-адресе отправителя/получателя
- порте отправителя/получателя
- времени
- протоколе
- пользователе
- пароле

Весь трафик прослушивается и проверяется, и непрошенный трафик может быть отброшен.

Недостатки:

- Взаимодействует на транспортном уровне и может потребовать изменения программ, предоставляющих транспортные функции.

### **12.d. Программный шлюз**

Шлюзы уровня приложений являются прокси-серверами для приложений, обменивающимися вместо клиентов данными с удаленными системами. Это позволяет находиться вне доступа извне за ДМЗ (демилитаризованной зоной, частью частной сети, видимой сквозь межсетевой экран) или разрешать межсетевому экрану предотвращать соединения извне. Фильтрация основывается на:

- разрешении или запрещении на основании IP-адреса источника/назначения
- на содержимом пакетов
- ограничения доступа к файлу на основе типа файла и расширения

Преимущества:

- Может кэшировать файлы, увеличивая производительность сети
- Детализированная регистрация всех подключений
- Хорошая расширяемость (некоторые прокси могут распределять кэшированные данные между собой)
- Нет прямого доступа извне
- Может изменять содержимое пакета на лету

Недостатки:

- Конфигурация является комплексной

Программные шлюзы принято считать наиболее безопасным решением, так как они не запускаются с правами администратора, и все узлы за ними будут недоступными из интернета.

Пример свободного шлюза:

- [Squid](#)

## 12.e. *Iptables*

Для нормального функционирования iptables должен быть включен в ядро. Я включил поддержку iptables модулей (команда iptables загрузит их по мере необходимости) и пересобрал ядро (но вы можете включить их в само ядро, если намереваетесь отказаться от загружаемых модулей ядра, как сказано выше). Для более детальной информации по настройке iptables в ядре обратитесь к странице [iptables Tutorial Chapter 5: Preparations](#). После пересборки нового ядра (или во время компиляции) вы должны добавить команду iptables. Для этого просто наберите `emerge iptables`.

Теперь проверим свою работу, запустив `iptables -L`. Если команда завершилась ошибкой, значит, что-то не так, поэтому проверьте еще раз настройки.

iptables — новый и весьма улучшенный межсетевой экран в ядрах Linux 2.4.x. Он является наследником ipchains для ядер Linux 2.2.x. Одним из значительных нововведений является то, что iptables способен совершать полноценную фильтрацию пакетов. С ее помощью стало возможным следить за каждым установленным TCP-соединением.

TCP-соединение содержит в себе серию пакетов, содержащих информацию IP-адресе и порте отправителя и получателя, а также последовательное число, поэтому пакеты могут быть воссозданы без потери данных. TCP является протоколом, ориентированном на подключение, в отличие от UDP, который не гарантирует доставку.

Изучая заголовок TCP-пакета, межсетевой экран может определить, является ли полученный пакет частью уже установленного соединения или нет, и принять решение, принять или отбросить этот пакет.

При использовании межсетевого экрана, не отслеживающего соединения, возможно провести его и заставить с помощью манипулирования заголовками TCP-пакета принимать пакеты, которые необходимо отбросить. Это может быть сделано с помощью установки SYN или других флагов заголовка TCP и создания поддельного пакета, который будет считаться частью уже установленного соединения (ведь межсетевой экран не будет отслеживать состояние соединения). При использовании фильтрации пакетов, отслеживающих соединения, можно отбрасывать подобные пакеты, если они не являются частью уже установленного соединения. Это также предотвратит «stealth-сканирование», разновидности сканирования, с помощью которого сканер отправляет пакеты с такой комбинацией флагов, при которых они не будут зарегистрированы межсетевым экраном, полагаящим, что это — обычные SYN-пакеты.

В iptables также есть различные возможности, как например NAT (Network Address Translation, сетевая трансляция адресов) и ограничения по частоте. Ограничение по частоте весьма полезно для предотвращения DoS-атак (Denial of Service, отказ от обслуживания), например SYN-наводнений.

TCP-соединения устанавливаются после так называемого троекратного рукопожатия. При установлении TCP-соединения клиент отправляет серверу пакет с установленным флагом SYN. При получении сервер возвращает клиенту пакет с установленными SYN+ACK. При его получении клиент отвечает третьим пакетом с ACK для подтверждения соединения.

SYN-наводнения основаны на отправке SYN-пакетов с одновременным запретом отправки пакетов SYN+ACK. Клиент может создать пакет с поддельным IP-адресом отправителя, так как ему не нужно на что-либо отвечать. Сервер заполнит очередь подключений полуоткрытыми соединениями, ожидающими окончательного пакета с ACK, до того, как удалит их из очереди. Очередь ограничена определенным числом, и когда она заполнится, то станет невозможным принимать новые подключения. Если пакет с ACK не будет получен по истечении определенного временного интервала, то он будет автоматически удален из очереди. Настройки таймаута могут быть различными, но обычно они находятся в пределах 30—60 секунд или даже больше. Клиент инициирует атаку, создавая множество SYN-пакетов с различных IP-адресов и отправляя их на адрес цели как можно чаще, заполняя тем самым очередь полуоткрытыми соединениями и не позволяя другим клиентам устанавливать легитимные подключения к серверу.

В данном случае может быть полезным ограничение отправки. Можно ограничить частоту отправки принятых SYN-пакетов с помощью `-m limit --limit 1/s`, тем самым ограничив число SYN-пакетов до одного в секунду и оградив свои ресурсы от SYN-наводнений.

**Примечание:** Другим решением, предотвращающим SYN-наводнения является использование [SYN cookies](#), которые позволяют вашему компьютеру отвечать на SYN-пакеты без заполнения очереди подключений. SYN cookies могут быть включены при конфигурировании ядра Linux, однако на данный момент эта поддержка считается экспериментальной.

А теперь немного практических занятий!

Будучи загруженным в ядро, iptables предоставляет 5 ловушек, в которые вы можете поместить свои правила. Они называются INPUT, OUTPUT, FORWARD, PREROUTING и POSTROUTING. Каждая из них называется цепочкой и содержит список правил. Каждое правило описывает, что если заголовок пакета выглядит как образец, то делать с этим пакетом нужно то-то и то-то. Если правило не подходит под пакет, то пакет переходит к следующему правилу в цепочке.

Вы можете поместить правила в 5 цепочках напрямую или создать новые цепочки и добавлять в них правила таким же способом, как и в обычную цепочку. Iptables поддерживает следующие параметры.

Параметр:	Описание:
-A	Добавление
-D	Удаление
-I	Вставка
-R	Замена
-L	Просмотр
-F	Удалить все правила в цепочке или все цепочки
-Z	Обнулить счетчики во всех цепочках



-C	Проверить этот пакет в цепочке
-N	Создать новую пользовательскую цепочку
-X	Удалить пользовательскую цепочку
-P	Изменить политику цепочки для цели
-E	Изменить имя цепочки
-p	протоколе
-s	Адрес/маска источника
-d	Адрес/маска назначения
-i	Входящее имя (имя ethernet)
-o	Исходящее имя (имя ethernet)
-j	Перейти (цель для правила)
-m	Расширенные сравнения (могут использовать внешние расширения)
-n	Числовой вывод адресов и портов
-t	Таблица для обработки
-v	Расширенный режим
-x	Расширить числа (вывести все значения)
-f	Проверять только второй и последующие фрагменты
-V	Версия пакета
--line-numbers	Указывать номера строк при выводе

Сначала попробуем заблокировать все ICMP-пакеты, идущие на наш компьютер, просто для того, чтобы познакомиться с iptables поближе.

#### Листинг 1: Блокировка всех пакетов ICMP

```
# iptables -A INPUT -p icmp -j DROP
```

Сначала мы указываем цепочку, в которую хотим поместить правило, затем протокол проверяемого пакета и в конце цель. Цель может быть именем пользовательской цепочки или одной из специальных целей — ACCEPT, DROP, REJECT, LOG, QUEUE или MASQUERADE. В случае, если мы укажем DROP, то пакет будет отброшен без уведомления клиента.

**Примечание:** LOG является «необрывающей» целью. Если пакет подпадает под правило с целью LOG, то он будет передан следующему правилу в цепочке, а не обработан только этим правилом. Это позволяет регистрировать пакеты и при этом их обрабатывать.

Теперь попробуйте выполнить `ping localhost`. Вы не должны получить ответа, так как iptables будет отбрасывать все входящие ICMP-сообщения. Вы также не сможете проверить и другие компьютеры, так как ответные ICMP-пакеты тоже будут отбрасываться. А теперь, чтобы вновь получать ICMP, очистим цепочку.

#### Листинг 2: Сбросить все правила

```
# iptables -F
```

Теперь взглянем на полноценную фильтрацию пакетов с помощью iptables. Если нужно полноценное исследование входящих пакетов на интерфейсе eth0, то выполним следующую команду:

### Листинг 3: Принимать пакеты от уже установленных соединений

```
# iptables -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Это разрешит принимать любой пакет из уже установленного соединения или связанного в цепочке INPUT. Вы можете отбросить любой пакет, не соответствующий состоянию таблицы, выполнив команду `iptables -A INPUT -i eth0 -m state --state INVALID -j DROP` сразу же после предыдущей. Эта команда включит фильтрацию пакетов на основании состояния, подключив внешнее расширение «state». Если вам необходимо разрешить подключаться клиентам к компьютеру, то вы можете использовать флаг `--state NEW`. Iptables содержит несколько модулей различного назначения. Вот некоторые из них:

Модуль/сравнение	Описание	Расширенные параметры
mac	Расширение для проверки MAC-адресов пакетов.	--mac-source
state	Включение проверки состояния соединения	--state (состоянием может быть ESTABLISHED,RELATED, INVALID, NEW)
limit	Частотное ограничение	--limit, --limit-burst
owner	Попытка проверить различные характеристики создателя пакета	--uid-owner userid --gid-owner groupid --pid-owner processid --sid-owner sessionid
unclean	Различные случайны проверки пакетов на правильность	

Теперь попробуем создать пользовательскую цепочку и применить ее для одного из существующих:

### Листинг 4: Создание пользовательской цепочки

```
(Создаем новую цепочку с одним правилом)
# iptables -X mychain
# iptables -N mychain
# iptables -A mychain -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
(Разрешающая политика по умолчанию для всего исходящего трафика. Входящий будет отброшен.)
# iptables -P OUTPUT ACCEPT
# iptables -P INPUT DROP
(И добавление ее в цепочку INPUT)
# iptables -A INPUT -j mychain
```

Применив правило к входящей цепочке, мы получим следующую политику: все исходящие пакеты будут пропущены, а все входящие — отброшены.

Документация может быть найдена по адресу [Netfilter/iptables documentation](http://Netfilter/iptables documentation).

Теперь взглянем на полный пример. Мои правила для межсетевого экрана/шлюза будут следующими:



- Локальная сеть должна иметь доступ к HTTP, HTTPS и SSH (также должен быть разрешен DNS)
- ICMP может содержать постороннюю информацию и не должен быть разрешен. Конечно же, мы разрешаем некоторые ICMP-сообщения.
- Сканирование портов должно быть определено и зарегистрировано
- SYN-атаки должны быть пресечены
- Весь остальной трафик должен быть отброшен и запротоколирован

#### Листинг 5: /etc/init.d/firewall

```
#!/sbin/runscript
IPTABLES=/sbin/iptables
IPTABLESSAVE=/sbin/iptables-save
IPTABLESRESTORE=/sbin/iptables-restore
FIREWALL=/etc/firewall.rules
DNS1=212.242.40.3
DNS2=212.242.40.51
#inside
IIP=10.0.0.2
IINTERFACE=eth0
LOCAL_NETWORK=10.0.0.0/24
#outside
OIP=217.157.156.144
OINTERFACE=eth1

opts="{opts} showstatus panic save restore showoptions rules"

depend() {
    need net
}

rules() {
    stop
    ebegin "Setting internal rules"

    einfo "Setting default rule to drop"
    $IPTABLES -P FORWARD DROP
    $IPTABLES -P INPUT DROP
    $IPTABLES -P OUTPUT DROP

    #default rule
    einfo "Creating states chain"
    $IPTABLES -N allowed-connection
    $IPTABLES -F allowed-connection
    $IPTABLES -A allowed-connection -m state --state ESTABLISHED,RELATED -j ACCEPT
    $IPTABLES -A allowed-connection -i $IINTERFACE -m limit -j LOG --log-prefix \
        "Bad packet from ${IINTERFACE}:"
    $IPTABLES -A allowed-connection -j DROP

    #ICMP traffic
    einfo "Creating icmp chain"
    $IPTABLES -N icmp_allowed
    $IPTABLES -F icmp_allowed
    $IPTABLES -A icmp_allowed -m state --state NEW -p icmp --icmp-type \
        time-exceeded -j ACCEPT
    $IPTABLES -A icmp_allowed -m state --state NEW -p icmp --icmp-type \
        destination-unreachable -j ACCEPT
    $IPTABLES -A icmp_allowed -p icmp -j LOG --log-prefix "Bad ICMP traffic:"
    $IPTABLES -A icmp_allowed -p icmp -j DROP

    #Incoming traffic
    einfo "Creating incoming ssh traffic chain"
    $IPTABLES -N allow-ssh-traffic-in
```

```

$IPTABLES -F allow-ssh-traffic-in
#Flood protection
$IPTABLES -A allow-ssh-traffic-in -m limit --limit 1/second -p tcp --tcp-flags \
    ALL RST --dport ssh -j ACCEPT
$IPTABLES -A allow-ssh-traffic-in -m limit --limit 1/second -p tcp --tcp-flags \
    ALL FIN --dport ssh -j ACCEPT
$IPTABLES -A allow-ssh-traffic-in -m limit --limit 1/second -p tcp --tcp-flags \
    ALL SYN --dport ssh -j ACCEPT
$IPTABLES -A allow-ssh-traffic-in -m state --state RELATED,ESTABLISHED -p tcp --dport ss

#outgoing traffic
einfo "Creating outgoing ssh traffic chain"
$IPTABLES -N allow-ssh-traffic-out
$IPTABLES -F allow-ssh-traffic-out
$IPTABLES -A allow-ssh-traffic-out -p tcp --dport ssh -j ACCEPT

einfo "Creating outgoing dns traffic chain"
$IPTABLES -N allow-dns-traffic-out
$IPTABLES -F allow-dns-traffic-out
$IPTABLES -A allow-dns-traffic-out -p udp -d $DNS1 --dport domain \
    -j ACCEPT
$IPTABLES -A allow-dns-traffic-out -p udp -d $DNS2 --dport domain \
    -j ACCEPT

einfo "Creating outgoing http/https traffic chain"
$IPTABLES -N allow-www-traffic-out
$IPTABLES -F allow-www-traffic-out
$IPTABLES -A allow-www-traffic-out -p tcp --dport www -j ACCEPT
$IPTABLES -A allow-www-traffic-out -p tcp --dport https -j ACCEPT

#Catch portscanners
einfo "Creating portscan detection chain"
$IPTABLES -N check-flags
$IPTABLES -F check-flags
$IPTABLES -A check-flags -p tcp --tcp-flags ALL FIN,URG,PSH -m limit \
    --limit 5/minute -j LOG --log-level alert --log-prefix "NMAP-XMAS:"
$IPTABLES -A check-flags -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP
$IPTABLES -A check-flags -p tcp --tcp-flags ALL ALL -m limit --limit \
    5/minute -j LOG --log-level 1 --log-prefix "XMAS:"
$IPTABLES -A check-flags -p tcp --tcp-flags ALL ALL -j DROP
$IPTABLES -A check-flags -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG \
    -m limit --limit 5/minute -j LOG --log-level 1 --log-prefix "XMAS-PSH:"
$IPTABLES -A check-flags -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP
$IPTABLES -A check-flags -p tcp --tcp-flags ALL NONE -m limit \
    --limit 5/minute -j LOG --log-level 1 --log-prefix "NULL_SCAN:"
$IPTABLES -A check-flags -p tcp --tcp-flags ALL NONE -j DROP
$IPTABLES -A check-flags -p tcp --tcp-flags SYN,RST SYN,RST -m limit \
    --limit 5/minute -j LOG --log-level 5 --log-prefix "SYN/RST:"
$IPTABLES -A check-flags -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
$IPTABLES -A check-flags -p tcp --tcp-flags SYN,FIN SYN,FIN -m limit \
    --limit 5/minute -j LOG --log-level 5 --log-prefix "SYN/FIN:"
$IPTABLES -A check-flags -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP

# Apply and add invalid states to the chains
einfo "Applying chains to INPUT"
$IPTABLES -A INPUT -m state --state INVALID -j DROP
$IPTABLES -A INPUT -p icmp -j icmp_allowed
$IPTABLES -A INPUT -j check-flags
$IPTABLES -A INPUT -i lo -j ACCEPT
$IPTABLES -A INPUT -j allow-ssh-traffic-in
$IPTABLES -A INPUT -j allowed-connection

einfo "Applying chains to FORWARD"
$IPTABLES -A FORWARD -m state --state INVALID -j DROP

```

```

$IPTABLES -A FORWARD -p icmp -j icmp_allowed
$IPTABLES -A FORWARD -j check-flags
$IPTABLES -A FORWARD -o lo -j ACCEPT
$IPTABLES -A FORWARD -j allow-ssh-traffic-in
$IPTABLES -A FORWARD -j allow-www-traffic-out
$IPTABLES -A FORWARD -j allowed-connection

einfo "Applying chains to OUTPUT"
$IPTABLES -A OUTPUT -m state --state INVALID -j DROP
$IPTABLES -A OUTPUT -p icmp -j icmp_allowed
$IPTABLES -A OUTPUT -j check-flags
$IPTABLES -A OUTPUT -o lo -j ACCEPT
$IPTABLES -A OUTPUT -j allow-ssh-traffic-out
$IPTABLES -A OUTPUT -j allow-dns-traffic-out
$IPTABLES -A OUTPUT -j allow-www-traffic-out
$IPTABLES -A OUTPUT -j allowed-connection

#Allow client to route through via NAT (Network Address Translation)
$IPTABLES -t nat -A POSTROUTING -o $OINTERFACE -j MASQUERADE
eend $?
}

start() {
  ebegin "Starting firewall"
  if [ -e "${FIREWALL}" ]; then
    restore
  else
    einfo "${FIREWALL} does not exists. Using default rules."
    rules
  fi
  eend $?
}

stop() {
  ebegin "Stopping firewall"
  $IPTABLES -F
  $IPTABLES -t nat -F
  $IPTABLES -X
  $IPTABLES -P FORWARD ACCEPT
  $IPTABLES -P INPUT ACCEPT
  $IPTABLES -P OUTPUT ACCEPT
  eend $?
}

showstatus() {
  ebegin "Status"
  $IPTABLES -L -n -v --line-numbers
  einfo "NAT status"
  $IPTABLES -L -n -v --line-numbers -t nat
  eend $?
}

panic() {
  ebegin "Setting panic rules"
  $IPTABLES -F
  $IPTABLES -X
  $IPTABLES -t nat -F
  $IPTABLES -P FORWARD DROP
  $IPTABLES -P INPUT DROP
  $IPTABLES -P OUTPUT DROP
  $IPTABLES -A INPUT -i lo -j ACCEPT
  $IPTABLES -A OUTPUT -o lo -j ACCEPT
  eend $?
}

```

```

save() {
    ebegin "Saving Firewall rules"
    $IPTABLESSAVE > $FIREWALL
    eend $?
}

restore() {
    ebegin "Restoring Firewall rules"
    $IPTABLESRESTORE < $FIREWALL
    eend $?
}

restart() {
    svc_stop; svc_start
}

showoptions() {
    echo "Usage: $0 {start|save|restore|panic|stop|restart|showstatus}"
    echo "start)      will restore setting if exists else force rules"
    echo "stop)       delete all rules and set all to accept"
    echo "rules)      force settings of new rules"
    echo "save)       will store settings in ${FIREWALL}"
    echo "restore)    will restore settings from ${FIREWALL}"
    echo "showstatus) Shows the status"
}

```

Вот несколько советов при создании правил для межсетевого экрана:

1. Создайте политику межсетевого экрана до того, как ее реализуете
2. Сделайте ее простой
3. Знайте принцип работы каждого протокола (прочитайте подходящий [RFC](#))
4. Всегда помните, что межсетевой экран — это просто программа, запускаемая с правами администратора.
5. Проверьте свой межсетевой экран

Если вам кажется, что iptables труден для понимания или нужно слишком много времени для настройки межсетевого экрана, то вы можете попробовать [Shorewall](#). Для генерации правил межсетевого экрана он использует iptables, но акцентируется на правилах и не указывает протокол.

## 12.f. Squid

Squid является очень хорошим прокси-сервером. Он может фильтровать трафик на основании времени, регулярных выражений в пути/URI, IP-адреса получателя и отправителя, домена, браузера, имени зарегистрированного пользователя, типа MIME и номера порта (протокола). Некоторые возможности не указаны, но трудно перечислить их всех.

В следующем примере я добавил фильтр баннеров вместо фильтра, основанного на фильтрации порносайтов. По этой причине gentoo.org не должен быть перечислен в списке порносайтов. И я не собираюсь тратить свое время на поиски хороших сайтов для вас.

В данном случае, вот мои правила:

- Веб-серфинг (HTTP/HTTPS) разрешен только в рабочее время (с понедельника по пятницу с 8 до 17 часов и в субботу с 8 до 13 часов), но если служащие остаются, то они должны работать, а не сидеть в интернете.
- Скачивание файлов не разрешено (.exe, .com, .arj, .zip, .asf, .avi, .mpg, .mpeg и так далее)

- Нам не нужны баннеры, поэтому они будут отфильтрованы и заменены на прозрачный GIF (здесь пригодится ваша креативность!).
- Все остальные подключения в/из Интернета запрещены.

Все это реализуется в 4 простых шага.

#### Листинг 6: /etc/squid/squid.conf

```
# Указываем IP и порт
http_port 10.0.2.1:3128

# Стандартная конфигурация
hierarchy_stoplist cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY

# Добавляем основные списки контроля доступа
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255

# Добавляем тех, кто может пользоваться прокси
acl localnet src 10.0.0.0/255.255.0.0

# Добавляем порты
acl SSL_ports port 443
acl Safe_ports port 80
acl Safe_ports port 443
acl purge method PURGE

# Добавляем список контроля доступа, основанного на регулярных
# выражениях, встречаемых в URL
acl archives urlpath_regex "/etc/squid/files.acl"
acl url_ads url_regex "/etc/squid/banner-ads.acl"

# Добавляем список контроля доступа, основанного на времени и дате
acl restricted_weekdays time MTWHF 8:00-17:00
acl restricted_weekends time A 8:00-13:00

acl CONNECT method CONNECT

# Разрешаем менеджеру подключаться с localhost
http_access allow manager localhost
http_access deny manager

# Разрешать очищать запросы только с localhost
http_access allow purge localhost
http_access deny purge

# Запрещать запросы на неизвестные порты
http_access deny !Safe_ports

# Запрещать подключение к портам, не относящимся к SSL
http_access deny CONNECT !SSL_ports

# Мои собственные правила

# Добавляем страницу, отображаемую на месте
# удаленного баннера
deny_info NOTE_ADS_FILTERED url_ads

# Запрещаем баннеры
http_access deny url_ads
```

```
# Запрещаем любые архивы
http_access deny archives

# Ограничиваем доступ в рабочим временем
http_access allow localnet restricted_weekdays
http_access allow localnet restricted_weekends

# Запрещаем все остальное
http_access deny all
```

Далее перечислим типы запрещенных к скачиванию файлов. Я добавил zip, viv, exe, mp3, rar, ace, avi, mov, mpg, mpeg, au, ra, arj, tar, gz и z файлы.

#### Листинг 7: /etc/squid/files.acl

```
\.[Zz][Ii][pP]$
\.[Vv][Ii][Vv].*
\.[Ee][Xx][Ee]$
\.[Mm][Pp]3$
\.[Rr][Aa][Rr]$
\.[Aa][Cc][Ee]$
\.[Aa][Ss][Ff]$
\.[Aa][Vv][Ii]$
\.[Mm][Oo][Vv]$
\.[Mm][Pp][Gg]$
\.[Mm][Pp][Ee][Gg]$
\.[Aa][Uu]$
\.[Rr][Aa]$
\.[Aa][Rr][Jj]$
\.[Tt][Aa][Rr]$
\.[Gg][Zz]$
\.[Zz]$
```

**Примечание:** Обратите внимание на [] с заглавными и строчными буквами. Это сделано для того, чтобы никто не смог обмануть наш фильтр, пытаясь скачать файл с расширением AvI вместо avi.

Далее добавим регулярные выражения для определения баннеров. Вы, возможно, будете более изобретательнее меня:

#### Листинг 8: /etc/squid/banner-ads.acl

```
/adv/*.gif$
/[Aa]ds/*.gif$
/[Aa]d[Pp]ix/
/[Aa]d[Ss]erver
/[Aa][Dd]/*.gif$
/[Bb]annerads/
/adbanner/*.gif$
/images/ad/
/reklame/
/RealMedia/ads/*
^http://www\.submit-it.*
^http://www\.eads.*
^http://ads\.
^http://ad\.
^http://ads02\.
^http://adaver.*.
^http://adforce\.
adbot\.com
/ads/*.gif.*
_ad\.*cgi
/Banners/
/SmartBanner/
```

```
/Ads/Media/Images/  
^http://static\.wired\.com/advertising/  
^http://*\.dejanews\.com/ads/  
^http://adfu\.blockstackers\.com/  
^http://ads2\.zdnet\.com/adverts  
^http://www2\.burstnet\.com/gifs/  
^http://www\.valueclick\.com/cgi-bin/cycle  
^http://www\.altavista\.com/av/gifs/ie_horiz\.gif
```

И в заключении мы хотим, чтобы отображался следующий файл вместо удаленного баннера. Он основан половине HTML-файла с прозрачным GIF-изображением размером 4x4.

#### Листинг 9: /etc/squid/errors/NOTE\_ADS\_FILTERED

```
<HTML>  
<HEAD>  
<META HTTP-EQUIV="REFRESH" CONTENT="0; URL=http://localhost/images/4x4.gif">  
<TITLE>ERROR: The requested URL could not be retrieved</TITLE>  
</HEAD>  
<BODY>  
<H1>Add filtered!</H1>
```

**Примечание:** Не закрывайте теги <HTML> и <BODY>. Squid сделает это самостоятельно.

Как вы видите, у Squid есть множество возможностей для очень эффективной фильтрации и кэширования. Можно даже использовать альтернативные прокси-сервера Squid для сегментирования очень больших сетей. Приведенная конфигурация подходит для небольшой сети с 1—20 пользователями.

Однако комбинация межсетевого экрана (iptables) и программного шлюза (Squid), возможно, является наилучшей, особенно если Squid находится где-нибудь в безопасном месте, где никто не может иметь к нему доступ извне. Нам все же стоит позаботиться об атаках изнутри.

Теперь вам необходимо настроить клиентские браузеры для использования прокси-сервера. Шлюз предотвратит попытки пользователей общаться с внешним миром без использования прокси.

**Примечание:** В Mozilla это может быть сделано через Edit->Preferences->Advanced->Proxies.

Также можно настроить прозрачное использование прокси, указав iptables перенаправлять весь исходящий трафик на вход Squid. Это можно сделать, добавив следующее правило на шлюзе:

#### Листинг 10: Разрешить проброс портов для нашего прокси-сервера

```
# iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to proxyhost:3128  
# iptables -t nat -A PREROUTING -p tcp --dport 443 -j DNAT --to proxyhost:3128
```

**Примечание:** Если прокси-сервер работает на том же узле, что и межсетевой экран (хотя это не рекомендуется, но может понадобиться из-за отсутствия свободных компьютеров), то используйте цель REDIRECT вместо DNAT (REDIRECT направит пакеты на localhost).

## 12.g. Изучено

Мы узнали:

1. Межсетевой экран является опасным сам по себе. Плохо настроенный межсетевой экран хуже, чем его отсутствие.
2. Как установить простой шлюз и прозрачный прокси-сервер.
3. Залогом хорошей работы межсетевого экрана заключается в знании протоколов, которые вы собираетесь разрешить.
4. IP-трафик не всегда содержит законные данные, например пакеты ICMP, которые могут содержать вредоносное содержимое.
5. Как предотвратить SYN-атаку.
6. Фильтрация HTTP-трафика предотвратит загрузку нежелательных изображений и вирусов.
7. Сочетание пакетных фильтров и программных шлюзов дает лучший контроль.

Теперь, если вам действительно это нужно, идите и создайте правила межсетевого экрана, отвечающего вашим требованиям.

## 13. Обнаружение вторжения

### 13.a. AIDE (Advanced Intrusion Detection Environment)

AIDE — это система обнаружения атак, основанная на узле (Host-Based Intrusion Detection System, HIDS), свободная альтернатива Tripwire (если вы уже работали с Tripwire, то не должны испытывать сложности с конфигурационным файлом AIDE). HIDS используется для обнаружения изменений в важных системных файлах настроек и двоичных файлах с помощью сверки уникальных криптографических контрольных сумм, созданных каждого из файлов и сохраненных в надежном месте. Принцип действия таков: заранее сохраненный и достоверный хеш сравнивается с сгенерированным с текущей копии каждого файла, чтобы определить какой из файлов был изменен. HIDS является великолепным решением для регистрации подозрительных изменений в системе, однако для того, чтобы она заработала, нужно приложить небольшие усилия.

Конфигурационный файл основан на применении регулярных выражений, макросов и правил для файлов и каталогов. Есть следующие макросы:

Макрос	Описание	Синтаксис
ifdef	если определен	@@ifdef "name"
ifndef	если не определен	@ @ i f n d e f "name"
define	определить переменную	@ @ d e f i n e "name" "value"
undef	удалить переменную	@ @ u n d e f "name"
ifhost	если "hostname"	@ @ i f h o s t "hostname"
ifnhost	если не "hostname"	@ @ i f n h o s t "hostname"



endif должен использоваться каждый раз, когда  
endif используется любой из вышеприведенных макросов, за @@endif  
исключением define и undef

Если у вас есть много компьютеров с Gentoo и вы хотите использовать на них AIDE, то макросы будут весьма полезными. Но не на всех компьютерах запускаются одни и те же сервисы или есть одни и те же пользователи.

Теперь необходимо установить флаги проверок для файлов и каталогов. Возможны сочетания прав доступа, свойств файлов и криптографических хешей (контрольных сумм).

Флаг	Описание
------	----------

p	права доступа
i	inode
n	количество ссылок
u	пользователь
g	группа
s	размер
b	число блоков
m	время последней модификации
a	время последнего доступа
c	время изменения
S	проверка растущего размера
md5	контрольная сумма MD5
sha1	контрольная сумма SHA1
rmd160	контрольная сумма RMD160
tiger	контрольная сумма Tiger
R	p+i+n+u+g+s+m+c+md5
L	p+i+n+u+g
E	Пустая группа
>	растущий лог-файл p+u+g+i+n+S

А если AIDE собран с поддержкой mhash, то доступны следующие возможности:

Флаг	Описание
------	----------

haval	контрольная сумма HAVAL
gost	контрольная сумма ГОСТ
crc32	контрольная сумма CRC32

Теперь вы можете создать собственные правила, основанные на сочетании вышеперечисленных флагов, например:

## Листинг 1: Создание набора правил для AIDE

```
All=R+a+sha1+rmcl160
Norm=s+n+b+md5+sha1+rmcl160
```

И последнее, что нам понадобится для создания собственного конфигурационного файла, — это умение добавлять правило для файла или каталога. Чтобы ввести правило, скомбинируйте файл или каталог с правилом. AIDE добавляет все файлы рекурсивно, если вы не указали обратное в другом правиле.

### Флаг Описание

! Не добавлять этот файл или каталог.

= Добавлять этот каталог, но не рекурсивно.

А теперь взглянем на полный пример:

## Листинг 2: /etc/aide/aide.conf

```
@@ifndef TOPDIR
@@define TOPDIR /
@@endif

@@ifndef AIDEDIR
@@define AIDEDIR /etc/aide
@@endif

@@ifhost smbserve
@@define smbactive
@@endif

# Расположение базы данных для чтения.
database=file: @@{AIDEDIR}/aide.db

# Расположение базы данных для записи.
database_out=file:aide.db.new

verbose=20
report_url=stdout

# Определение правил
All=R+a+sha1+rmcl160
Norm=s+n+b+md5+sha1+rmcl160

@@{TOPDIR} Norm
!@@{TOPDIR}etc/aide
!@@{TOPDIR}dev
!@@{TOPDIR}media
!@@{TOPDIR}mnt
!@@{TOPDIR}proc
!@@{TOPDIR}root
!@@{TOPDIR}sys
!@@{TOPDIR}tmp
!@@{TOPDIR}var/log
!@@{TOPDIR}var/run
!@@{TOPDIR}usr/portage
@@ifdef smbactive
!@@{TOPDIR}etc/smb/private/secrets.tdb
@@endif
=@@{TOPDIR}home Norm
```

В приведенном примере мы указали некоторые макросы, корневой каталог, с

которого будет начинать свою работу AIDE, и каталог, в котором находится AIDE. AIDE сверяется с /etc/aide/aide.db при проверке на целостность. Но при обновлении или создании нового файла она сохраняет информацию в /etc/aide/aide.db.new. Это делается для того, чтобы не переписать старую базу данных. Параметр report\_URL пока не реализован, но, согласно автору, он должен отправлять электронную почту или даже запуск сценариев.

Теперь по умолчанию пакет AIDE поставляется вместе с рабочим конфигурационным файлом, сценарием-помощником и сценарием планировщика заданий crontab. Сценарий-помощник содержит множество задач для вас и предоставляет более дружелюбный интерфейс. Чтобы просмотреть все доступные параметры, попробуйте aide --help. Все что вам нужно, чтобы начать, — запустить aide -i, и теперь сценарий crontab должен найти базу данных и ежедневно отправлять отчеты по электронной почте. Рекомендуется все же пересмотреть файл /etc/aide/aide.conf и удостовериться, что конфигурация подходит для данного компьютера.

**Примечание:** В зависимости от процессора, скорости чтения диска и установленных для файлов флагов, это может занять много времени.

**Примечание:** Не забудьте установить алиасы, чтобы получать почту, адресуемую суперпользователю. Иначе вы никогда не узнаете, что пишет для вас в отчетах AIDE.

Все же есть небольшой риск, заключающийся в том, что злоумышленник может (если узнает об установленном AIDE) изменить локальную базу данных файлов, обновить ее или изменить /usr/bin/aide. Поэтому вы должны создать компакт-диск или иной носитель и сохранить на нем копию файла .db и двоичных файлов AIDE.

Дополнительную информацию вы можете найти на сайте проекта [AIDE](#)

### 13.b. Snort

Snort является сетевой системой обнаружения вторжения (Network Intrusion Detection System, NIDS). Чтобы установить и настроить его, воспользуйтесь следующими примерами.

#### Листинг 3: /etc/conf.d/snort

```
PIDFILE=/var/run/snort_eth0.pid
MODE="full"
NETWORK="10.0.0.0/24"
LOGDIR="/var/log/snort"
CONF=/etc/snort/snort.conf
SNORT_OPTS="-D -s -u snort -dev -l $LOGDIR -h $NETWORK -c $CONF"
```

#### Листинг 4: /etc/snort/snort.conf

```
(#var 1)
var HOME_NET 10.0.0.0/24
var EXTERNAL_NET any
var SMTP $HOME_NET
var HTTP_SERVERS $HOME_NET
var SQL_SERVERS $HOME_NET
var DNS_SERVERS [10.0.0.2/32,212.242.40.51/32]
var RULE_PATH ./

(#var 2)
preprocessor frag2
preprocessor stream4: detect_scans detect_state_problems detect_scans disable_evasion_aler
preprocessor stream4_reassemble: ports all
```

```

preprocessor http_decode: 80 8080 unicode iis_alt_unicode double_encode iis_flip_slash full
preprocessor rpc_decode: 111 32771
preprocessor bo: -nobrute
preprocessor telnet_decode

(## 3)
include classification.config

(## 4)
include $RULE_PATH/bad-traffic.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/telnet.rules
include $RULE_PATH/smtp.rules
include $RULE_PATH/rpc.rules
include $RULE_PATH/rservices.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/tftp.rules
include $RULE_PATH/web-cgi.rules
include $RULE_PATH/web-coldfusion.rules
include $RULE_PATH/web-iis.rules
include $RULE_PATH/web-frontpage.rules
include $RULE_PATH/web-misc.rules
include $RULE_PATH/web-attacks.rules
include $RULE_PATH/sql.rules
include $RULE_PATH/x11.rules
include $RULE_PATH/icmp.rules
include $RULE_PATH/netbios.rules
include $RULE_PATH/misc.rules
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/shellcode.rules
include $RULE_PATH/policy.rules
include $RULE_PATH/porn.rules
include $RULE_PATH/info.rules
include $RULE_PATH/icmp-info.rules
include $RULE_PATH/virus.rules
# include $RULE_PATH/experimental.rules
include $RULE_PATH/local.rules

```

#### Листинг 5: /etc/snort/classification.config

```

config classification: not-suspicious,Not Suspicious Traffic,3
config classification: unknown,Unknown Traffic,3
config classification: bad-unknown,Potentially Bad Traffic, 2
config classification: attempted-recon,Attempted Information Leak,2
config classification: successful-recon-limited,Information Leak,2
config classification: successful-recon-largescale,Large Scale Information Leak,2
config classification: attempted-dos,Attempted Denial of Service,2
config classification: successful-dos,Denial of Service,2
config classification: attempted-user,Attempted User Privilege Gain,1
config classification: unsuccessful-user,Unsuccessful User Privilege Gain,1
config classification: successful-user,Successful User Privilege Gain,1
config classification: attempted-admin,Attempted Administrator Privilege Gain,1
config classification: successful-admin,Successful Administrator Privilege Gain,1

# NEW CLASSIFICATIONS
config classification: rpc-portmap-decode,Decode of an RPC Query,2
config classification: shellcode-detect,Executable code was detected,1
config classification: string-detect,A suspicious string was detected,3

```

```

config classification: suspicious-filename-detect,A suspicious filename was detected,2
config classification: suspicious-login,An attempted login using a suspicious username was
config classification: system-call-detect,A system call was detected,2
config classification: tcp-connection,A TCP connection was detected,4
config classification: trojan-activity,A Network Trojan was detected, 1
config classification: unusual-client-port-connection,A client was using an unusual port,2
config classification: network-scan,Detection of a Network Scan,3
config classification: denial-of-service,Detection of a Denial of Service Attack,2
config classification: non-standard-protocol,Detection of a non-standard protocol or event
config classification: protocol-command-decode,Generic Protocol Command Decode,3
config classification: web-application-activity,access to a potentially vulnerable web app
config classification: web-application-attack,Web Application Attack,1
config classification: misc-activity,Misc activity,3
config classification: misc-attack,Misc Attack,2
config classification: icmp-event,Generic ICMP event,3
config classification: kickass-porn,SCORE! Get the lotion!,1

```

Дополнительной информация может быть найдена на сайте [Snort](#).

### 13.c. Обнаружение вредоносных программ с помощью *chkrootkit*

Система HIDS, как AIDE, является лучшим средством для обнаружения изменений в вашей системе, но хуже не будет, если будет установлен еще один рубеж обороны. *chkrootkit* — это утилита, которая сканирует системные файлы на наличие руткитов — программ, разработанных для сокрытия присутствия взломщика и сохранения доступа к системе, а также на наличие следов кейлоггеров и прочих вредоносных программ. Хотя *chkrootkit* (и прочие аналоги, например *rkhunter*) являются полезными утилитами, как контроля за системой, так и для обнаружения следов вторжения, но они не могут гарантировать, что ваша система в безопасности.

Наилучший способ использовать *chkrootkit* для обнаружения внедрения — запускать его с помощью *cron*. Сначала установите *app-admin/chkrootkit*. *chkrootkit* может запускаться из командной строки или же с помощью записи в *cron* следующего вида:

#### Листинг 6: Назначение задания *chkrootkit* в *crontab*

```
0 3 * * * /usr/sbin/chkrootkit
```

## 14. Регулярные обновления

### 14.a. Регулярные обновления

После успешной установки и настройки ее безопасности систему, тем не менее, нельзя называть завершенной. Процесс обеспечения безопасности является постоянным по времени, ведь большинство известных взломов является результатом использования известных уязвимостей в устаревших системах. Постоянное обновление системы — наиболее значимый шаг, который вы можете совершить для обеспечения наилучшей безопасности.

Если у вас установлена последняя версия *portage*, то, предварительно обновив дерево с помощью команды *emerge --sync*, вы можете запустить команду *glsa-check --list*, которая проверит вашу систему на наличие известных уязвимостей. *glsa-check* является частью пакета *app-portage/gentoolkit*.

#### Листинг 1: Пример вывода *glsa-check -l*

```

# glsa-check -l
WARNING: This tool is completely new and not very tested, so it should not be
used on production systems. It's mainly a test tool for the new GLSA release

```

and distribution system, it's functionality will later be merged into emerge and equery.

Please read <http://www.gentoo.org/proj/en/portage/glsa-integration.xml> before using this tool AND before reporting a bug.

[A] means this GLSA was already applied,  
[U] means the system is not affected and  
[N] indicates that the system might be affected.

200406-03 [N] sitecopy: Multiple vulnerabilities in included libneon ( net-misc/sitecopy )  
200406-04 [U] Mailman: Member password disclosure vulnerability ( net-mail/mailman )  
.....

**Предупреждение:** Утилита glsa-check все еще является экспериментальной, поэтому, если вы действительно ставите безопасность превыше всего, то лучше также свериться с другими источниками.

Все строки, содержащие [A] и [U], могут быть игнорированы, так как они неприменимы для данной системы.

**Важно:** Не забывайте, что использование emerge -vpuD world не устанавливает все необходимые обновления пакетов. Используйте glsa-check, если хотите быть уверенным, что все GLSA исправлены в вашей системе.

## Листинг 2: Проверка всех GLSA

```
(Подвержена ли ваша система GLSA?)
# glsa-check -t all
WARNING: This tool is completely new and not very tested, so it should not be
used on production systems. It's mainly a test tool for the new GLSA release
and distribution system, it's functionality will later be merged into emerge
and equery.
Please read http://www.gentoo.org/proj/en/portage/glsa-integration.xml
before using this tool AND before reporting a bug.

This system is affected by the following GLSA:
200504-06
200510-08
200506-14
200501-35
200508-12
200507-16

(Просмотр всех пакетов, подверженных переустановке)
# glsa-check -p $(glsa-check -t all)
(частичный вывод)
Checking GLSA 200504-06
The following updates will be performed for this GLSA:
  app-arch/sharutils-4.2.1-r11 (4.2.1-r10)

*****

Checking GLSA 200510-08
The following updates will be performed for this GLSA:
  media-libs/xine-lib-1.1.0-r5 (1.1.0-r4)

(применение необходимых исправлений)
# glsa-check -f $(glsa-check -t all)
```

Если вы обновили запущенный сервис, не забудьте перезапустить его.

Также рекомендуется регулярно [обновлять ядро](#).

Вы можете также подписаться на список рассылки gentoo-announce, чтобы получать

уведомления GLSA по электронной почте. Инструкции по подписке могут быть найдены на странице [Введение в почтовые рассылки Gentoo Linux](#).

Другим хорошим источником информации по безопасности является [список рассылки Bugtraq](#).

# Обновление Gentoo Linux

## Руководство по обновлению Gentoo

Ссылка на оригинал: <http://www.gentoo.org/doc/en/gentoo-upgrading.xml>

С версии: 1.2

Перевод: [vnp](#), [Михаил Ярмиш](#), [Алексей Чумаков](#), [Батогов Е.В.](#), а.к.а [JohnBat26](#)

Примечание: Из оригинала включён только самый **актуальный** материал !

## 1. Gentoo и обновления

### Философия

Здесь, на земле Gentoo, обновление понимается совсем не так, как во всем остальном мире Linux. Вы, вероятно, уже знаете, что в наших дистрибутивах мы никогда не следовали «классическому» способу обновления программ: дожидаться нового релиза, скачать его, прожечь, засунуть диск в дисковод и, наконец, следовать установочным инструкциям.

Вы знаете (вы же не зря выбрали Gentoo), как такой процесс раздражает опытного пользователя, стремящегося всегда быть на переднем крае. Учитывая популярность и распространенность инструментов типа **apt** или **apt-rpm**, облегчающих быстрые и частые обновления, мы полагаем, что и опытные пользователи других дистрибутивов испытывают те же чувства. Однако ни один дистрибутив не приспособлен для удовлетворения нужд требовательных пользователей лучше Gentoo, поскольку Gentoo с самого начала строился на идее быстрых последовательных обновлений.

В идеале, вы устанавливаете систему однажды, и более никогда не беспокоитесь о выпусках: просто следуйте инструкциям из введения в Portage в настольной книге Gentoo, где описывается, как поддерживать актуальность системы. Хотя обычно такого подхода достаточно, иногда в ядро системы вносятся изменения, требующие установки вручную.

### Выпуски и профили

О процессе выпуска версий Gentoo постоянно задают вопрос: «Зачем нужно часто делать новые выпуски, если они не предназначены для обновления программ у пользователей?». Тому есть ряд причин:

Новый выпуск — это новые исправленные и дополненные установочные диски.

Новый выпуск несет обновленный набор пакетов GRP, так что пользователи,



избирающие «быстрый способ» установки (stage3 + заранее скомпилированные пакеты), получают не устаревшую систему.

Наконец, в новых выпусках время от времени реализуются функции, несовместимые с предыдущими выпусками.

Если в выпуск входят возможности, несовместимые с предыдущими, или содержится набор фундаментальных пакетов и установок, сильно меняющих поведение системы, мы говорим, что он задает новый профиль.

**Профиль — это набор конфигурационных файлов, хранимый в подкаталоге `/usr/portage/profiles`, описывающий сценарии `ebuild` системных пакетов, флаги использования (USE) по умолчанию, отображение виртуальных пакетов по умолчанию, а также архитектуру, на которой запускается система.**

Используемый профиль определяется символьной ссылкой `/etc/make.profile`, которая указывает на подкаталог в `/usr/portage/profiles`, содержащий файлы профиля. К примеру, профиль `x86 2007.0` по умолчанию находится в `/usr/portage/profiles/default-linux/x86/2007.0`. Файлы в родительских каталогах также входят в состав профиля (и, таким образом, совместно используются различными подпрофилями). Поэтому мы называем такие профили каскадными профилями.

Профили, устаревающие с появлением новых, хранятся в `/usr/portage/profiles` наравне с актуальными, но помечены как нежелательные для использования. Для этого в каталог профиля помещается файл **deprecated**. Файл содержит имя профиля, на который рекомендуется обновиться. Portage использует эту информацию для автоматического предупреждения о необходимости перехода на новый профиль.

Существуют различные причины для создания нового профиля: выпуск новых несовместимых версий базовых пакетов (таких, как **baselayout**, **gcc**, или **glibc**), изменения в USE-флагах по умолчанию или отображении виртуальных пакетов, или, возможно, серьезные системные изменения.

## 2. Гонка за новыми выпусками

### *Выпуски без изменений в профиле*

Если объявлен новый выпуск Gentoo, в котором не появился новый профиль, то твердо можно притвориться, что вы его не заметили :).

Ваша система будет в точности соответствовать вновь установленной с нового выпуска, если вы обновите установленные пакеты согласно описанию из Настольной книги Gentoo.

### *Выпуски с изменениями в профиле*

Если в выпуске появился новый профиль (как, например, в 2007.0 для x86), у вас

есть возможность на него перейти.

Естественно, вы не обязаны это делать, и можете продолжать пользоваться старым профилем, просто обновляя свои пакеты, как описано в Настольной книге Gentoo.

Однако, Gentoo крайне рекомендует обновлять профиль, если он объявляется нежелательным для использования. Когда такое происходит, это означает, что разработчики больше не планируют его поддерживать. В приведенной ниже таблице можно быстро уточнить, какие профили поддерживаются в данный момент.

Переход на новый профиль придется осуществлять вручную. Необходимые действия могут сильно отличаться от выпуска к выпуску, в зависимости от глубины изменений, привносимых в новом профиле.

В простейшем случае, вам просто потребуется изменить символьную ссылку **/etc/make.profile**, а в худшем — придется перекомпилировать всю систему, тем временем изображая танец вуду. Процесс перехода обычно описывается в замечаниях к выпуску. Кроме того, есть инструкции в конце данного руководства.

### ***Поддерживаемые профили***

Вы можете увидеть список профилей на Вашей архитектуре, официально поддерживаемой Gentoo когда установите утилиту **eselect (emerge eselect)** и затем выполните следующую команду:

**Листинг 2.1** Просмотр поддерживаемых профилей

```
# eselect profile list
```

## **3. Инструкции по обновлению профилей**

### ***Общие инструкции.***

**Важно:** Удостоверьтесь в том, что Ваши порты обновлены перед выполнением любых изменений с профилями !

Первое, выполните **emerge eselect**. Утилита eselect позволяет Вам просматривать и выбирать профили без необходимости создания или удаления символических ссылок вручную.

**Листинг 3.1:** Выбор профиля с помощью утилиты eselect

(Просмотр доступных профилей)

```
# eselect profile list
```

(Выберите номер Вашего желаемого профиля из списка)

```
# eselect profile set <номер профиля>
```

Если Вы предпочитаете изменять профили вручную, тогда просто сделайте следующее:

**Листинг 3.2** Изменение профилей вручную.

```
# rm /etc/make.profile  
# ln -s ../usr/portage/profiles/<выбранный профиль> /etc/make.profile
```

**Примечание:** Для большинства архитектур существуют два типа подпрофилей: **desktop** и **server**. Внимательно изучите эти профили, так как они могут удовлетворять Вашим нуждам лучше, чем чрезвычайно минимальные профили по умолчанию.

### **Обновление до 2007.0 или 2006.1**

Если Вы намереваетесь обновиться до уровня этих профилей, то Вам следовало бы знать, что они ожидают что по умолчанию в системе установлена локаль Unicode. Или более точно, должна быть определена строка `UNICODE="yes"` в файле `/etc/rc.conf`. Для этоо Вы должны создать Unicode локаль для Вашей системы. Пожалуйста, прочитайте UTF-8 guide чтобы узнать как создать правильную локаль.

Альтернативно, если Вы не желаете устанавливать локаль, тогда Вы можете определить `UNICODE="no"` в файле `/etc/rc.conf`. И переустановить (re-emerge) пакет **baselayout** (или ждите следующего обновления этого пакета) с USE флагом: `-unicode`. Вы можете установить `-unicode` только для baselayout или Вы можете установить его глобально для всех пакетов, добавив Вашу переменную USE в файл `/etc/make.conf`.

**Листинг 3.3** Опционально: удаление поддержки Unicode.

(Для удаления поддержки Unicode только для baselayout)

```
# echo "sys-apps/baselayout -unicode" >> /etc/portage/package.use  
# emerge -a baselayout
```

(Для удаления поддержки Unicode для всей системы)

```
# nano -w /etc/make.conf  
USE="-unicode"  
# emerge -a baselayout
```

**Примечание:** Если Вы обновляетесь до 2007.0 на архитектуре SPARC, тогда вы должны следовать [GCC Upgrading Guide](#)

## HOWTO Полное обновление системы

Ссылка на оригинал:

[http://ru.gentoo-wiki.com/HOWTO\\_Полное\\_обновление\\_системы](http://ru.gentoo-wiki.com/HOWTO_Полное_обновление_системы)

С версии: 1.2

### Приведение в порядок /var/lib/portage/world

В **world** должен быть список программ, которые нужно доустановить к тем, которые уже входят в "system" (т.е. в текущий профайл).

- в world не должно быть никаких библиотек, и т.д., которые не нужны сами по себе, а нужны только для удовлетворения чьих-то зависимостей (чтобы не продолжать устанавливать/обновлять их, если они уже станут не нужны по какой-то причине) ;
- программ, которые уже входят в "**system**", не должно быть в **world**
- в world нельзя указывать определенную версию софта, это лучше делать в **/etc/portage/package.mask** ;
- скрипт **regenworld** может помочь восстановить world путем анализа **/var/log/merge.log** и генерации на его базе файла world (он перезапишет текущий world!) ;
- скрипт **dep -p -w** (входящий в состав пакета **udept**) поможет найти избыточные записи в **world**(которые всё-равно нужны другим записям в world или входят в **system**);
- перед серьёзными обновлениями желательно просмотреть **/etc/portage/\***, т.к. там могут быть уже не актуальные записи мешающие текущему обновлению.

### Обновление profile

- Не каждый Gentoo release включает в себя новый profile (например, 2004.1 был без profile).
- Даже если новый profile есть, то переходить на него не обязательно (если это будет обязательно, то старый профайл будет "протестующий" (deprecated) и emerge об этом должен будет громко кричать).
- Инструкции по обновлению profile будут выкладываться здесь: <http://www.gentoo.org/doc/ru/gentoo-upgrading.xml> и как правило сводиться к изменению симлинка **/etc/make.profile**

### USE-флаги

- Запустить **emerge -uDpv --newuse world** и проверить что USE-флаги для всех пакетов выставлены корректно, и при необходимости скорректировать

USE-флаги выставляются в **/etc/make.conf** и **/etc/portage/package.use**

### Запуск обновления системы (если не нужно обновлять toolchain)

- Если

```
emerge -puDpv --newuse world
```

показывает что будет обновляться пакет входящий в toolchain (linux-headers, glibc,

binutils или gcc), то крайне рекомендуется полностью перекомпилировать всю систему - см. следующий пункт - а иначе можно вместо следующего пункта просто запустить:

```
emerge -uDav --newuse world
```

## Некоторые причины не использовать emerge -U вместо -и

- Причина 1: Проблемы со **SLOT**

Это, к примеру, происходит потому, что некоторые люди хотели **gimp-2** вместо **gimp-1.2**. Представьте ситуацию, где **gimp-1.2** помечен stable и находится в SLOT 1, **gimp-2** помечен unstable и находится в SLOT 2. Теперь при выполнении **ACCEPT\_KEYWORDS=~x86 emerge gimp** получите **gimp-2**.

Позже, когда вы посчитаете, что наступило время обновить свою систему чем-либо похожим на "emerge -U world", эта команда установит gimp-1.2, потому, что gimp находится в world-файле, и флаг "-U" не обрабатывает SLOT должным образом.

- Причина 2: Проблемы, в случае удаления ebuild-ов с Portage-дерева.

Допустим, в Portage находятся 2 версии пакетов foo, foo-1.4 (помеченный как stable) и foo-1.6 (помеченный как unstable). Вы хотите вариант unstable и делаете emerge, как с вышеуказанным gimp. Позже обновляете world как было сказано выше, но в промежутке этого времени вышло критическое обновление для foo-1.6 - foo-1.6.1. Теперь появляется несколько возможностей обработки.

1. foo-1.6 был удален из Portage. Будет установлен foo-1.4, несмотря на "снижение" версии вместо флага "-U"
2. Ситуация будет еще хуже, если foo-1.6 не был удалён из Portage по какой-либо причине: foo-1.6 (тот, что с критической уязвимостью) будет оставаться на вашей системе до тех пор, пока не будет помечено stable что-либо выше чем foo-1.6.

**Предупреждение:** Не рекомендуется использовать **ACCEPT\_KEYWORDS=~x86 emerge foo** о чем можно почитать здесь <http://www.gentoo.org/doc/ru/gentoo-amd64-faq.xml#keyword>

## Обновление одного из пакетов входящих в toolchain

- Если обновляется хотя-бы один из **linux-headers**, **glibc**, **binutils** или **gcc**, то рекомендуется пересобрать их дважды, после чего весь **system**, после чего весь **world**.

**Примечание:** Цель двойной компиляции **toolchain** - получить гарантированно стабильный и корректный **toolchain** не зависящий от предыдущего. Перекомпилировать **system/world** после этого жёсткой необходимости нет, по крайней мере если остальной софт продолжает работать (возможно даже используя библиотеки из старого **toolchain** - см. предыдущие пункты об апгрейде).

Цель перекомпиляции **system/world** - чтобы весь софт получил потенциальное преимущество от установки нового **toolchain**. **system** перекомпилируется перед **world** из тех-же соображений, т.к. при компиляции программ из **world** используются утилиты из **system**.

- Если увеличивается первая или вторая цифра версии **gcc**, то перед второй сборкой нужно переключиться на новую версию через **gcc-config** - иначе новый **gcc** просто установится параллельно со старым в "новый слот", но по умолчанию использоваться будет старый.

- При сборке **system** после двойной перекомпиляции **toolchain** нет необходимости опять компилировать **toolchain** как часть **system**. Аналогично при сборке **world** после **system** нет необходимости опять компилировать пакеты из **system** как часть **world**. Это можно попробовать обойти либо вручную, либо используя скрипты [1], либо через бинарные пакеты и **`emerge -k`** (я предпочитаю последний вариант).
- Итак, рекомендованный набор команд:

#### Листинг 1. Рекомендованный набор команд

```
# для того, чтобы безопасно использовать `emerge -k` нужно очистить
# каталог с текущими бинарными пакетами
# (напр., переместить его в /tmp/portage-packages)
```

```
pkgdir=$(portageq pkgdir)
mv $pkgdir /tmp/portage-packages1
install -d -o portage -g portage $pkgdir
```

```
# первая сборка toolchain
emerge linux-headers glibc binutils gcc-config gcc
```

```
# выбрать новый gcc если он установился в новый слот
gcc-config имя_или_номер_нового_gcc
# см. `gcc-config -l`
source /etc/profile
```

```
# компиляция toolchain с созданием бинарных пакетов
emerge -b glibc binutils gcc portage
```

```
# не компилировать glibc, binutils и gcc
emerge -bke system
```

```
# не компилировать предыдущие пакеты (включая system)
emerge -bke world
```

**Примечание:** Чисто теоретически существует пакет **binutils-config**, который когда-нибудь может потребоваться использовать аналогично **gcc-config**.

#### Обновление безопасности.

**Примечание:** Даже после **`emerge -uDav --newuse world`** в системе могут оставаться устаревшие пакеты с дырами в безопасности - в слотах!

```
glsa-check -l | grep '[N\]'
emerge ... # если нужно
```

#### Удаление неиспользуемых пакетов.

После обновления системы в ней могут оказаться пакеты, которые никто не использует. Эти пакеты желательно удалить, т.к. они не будут в дальнейшем обновляться при **`emerge -uDav --newuse world`**.

```
emerge -a depclean # очень осторожно!!!
```

- После обновления библиотек может потребоваться перекомпилировать программы, которые эти библиотеки используют:

**Примечание:** Для **glsa-check**, **revdep-rebuild** необходимо установить пакет **gentoolkit**

```
rm /root/.revdep-rebuild*.*_*
revdep-rebuild -p
revdep-rebuild
```

## Обновление конфигов.

dispatch-conf

- Если используется **runit-init** и обновлялся пакет **baselayout**, то нужно восстановить /sbin/init:

```
ls -l /sbin/*init*
if (/sbin/init это бинарник, а не симлинк) {
mv /sbin/init /sbin/init-sysv
ln -s runit-init /sbin/init
}
```

- Отслеживание важных сообщений при установке пакетов.

В процессе **emerge world** выдаётся очень много сообщений, причём важные комментарии перемешаны с командами компиляции, и отследить их при сборке нескольких пакетов одновременно невозможно.

Но все эти сообщения можно получить из log-файлов после окончания установки **emerge world**. Для этого нужно использовать либо **enotice**, либо **portlog-info**.

# Руководство Gentoo Linux по обновлению GCC

Ссылка на оригинал: <http://www.gentoo.org/doc/ru/gcc-upgrading.xml>

С версии: 1.0

## 1. Введение

### Обновление GCC

Зачем нужно обновлять? Ну, GCC довольно похож на любой другой пакет в вашей системе, но чуточку более важен. GCC следует обновлять всякий раз, когда в новой версии исправляются какие-нибудь раздражающие вас ошибки, добавляется новые нужные функции, или если вы хотите держать свою систему обновленной. Если ни один из этих случаев к вам не относится, обновление можно спокойно откладывать, пока ваша версия GCC поддерживается разработчиками Gentoo.

Если вы устанавливаете новую версию GCC, система не переключается на ее использование автоматически. Вам необходимо явно запросить изменение, потому что в процессе перехода может потребоваться несколько дополнительных шагов. Если вы решите не переключаться, система Portage продолжит использовать более старую версию компилятора, пока вы не передумаете или пока не удалите старый компилятор из своей системы.

В этом руководстве описываются необходимые шаги, нужные для полноценного обновления компилятора, используемого вашей системой Gentoo. Отдельный раздел посвящен переходу с GCC 3.3 на 3.4 или более новые версии и проблемам с libstdc++. Другой частный раздел предназначен пользователям, впервые устанавливающим Gentoo из архива третьей стадии (stage3), после выхода новой версии GCC.

Примечание: Необходимо заметить, что обновление с GCC-3.4 до GCC-4.0 или более нового не требует существенных изменений от пользователя, так как GCC-3.4 и GCC-4.0 используют одинаковый двоичный прикладной интерфейс (ABI). Все что нужно, это использовать gcc-config, чтобы выбрать желаемый компилятор.

## 2. Общие указания по обновлению

### Введение

**Важно:** Если вы ищете подробные указания по обновлению с GCC-3.3 на GCC-3.4 или более новый, обратитесь к соответствующему разделу.

**Важно:** Если вы ищете подробные указания по обновлению GCC на вновь установленных системах, обратитесь к соответствующему разделу.

Вообще говоря, переход на версии с исправленными ошибками (bugfix release), как с 3.3.5 на 3.3.6, должен быть довольно безопасен: надо только установить новую версию, переключиться на нее и пересобрать единственный затрагиваемый пакет —



libtool. Однако, при некоторых обновлениях GCC нарушается двоичная совместимость, в таких случаях может потребоваться пересборка не только затрагиваемых пакетов, но и даже всего системного набора и пакетов, необходимых для компиляции.

Говоря о необходимости ручного переключения на новую версию компилятора, мы сказали, что оно не происходит автоматически. Тем не менее, есть одно исключение — переход на версию с исправленными ошибками (как с 3.3.5 на 3.3.6), если не используется режим «multislot», позволяющий обеим версиям сосуществовать в одной системе. По умолчанию этот режим выключен, ведь большинству пользователей он ничего не даст.

#### **Листинг 2.1: Обновление GCC**

```
# emerge -uav gcc
```

(Вместо "i686-pc-linux-gnu-3.4.5" укажите обновленную версию GCC и настройки CHOST)

```
# gcc-config i686-pc-linux-gnu-3.4.5
```

```
# source /etc/profile
```

(Пересборка libtool)

```
# emerge --oneshot -av libtool
```

Теперь пересоберите набор программ для компиляции, затем world, используя новый компилятор.

#### **Листинг 2.2: Пересборка системы**

```
# emerge -eav system
```

```
# emerge -eav world
```

Теперь можно без опасений удалить старую версию GCC. Если вы чувствуете такую необходимость, введите следующую команду (как обычно, вместо =sys-devel/gcc-3.3\* укажите версию, которую собираетесь удалить):

#### **Листинг 2.3: Удаление более старой версии GCC**

```
# emerge -aC =sys-devel/gcc-3.3*
```

## **3. Переход с GCC-3.3 на 3.4 или более новый**

### **Введение**

Переход с GCC-3.3 на 3.4 или более новую версию не так гладок, ведь между этими версиями изменился двоичный прикладной интерфейс C++ (ABI). Также придется позаботиться о существующей проблеме с библиотекой libstdc++.

### **Варианты**

**Важно:** Если вы обновляете GCC на машине SPARC, вам придется выбрать путь полной пересборки системы из-за некоторых внутренних изменений двоичного интерфейса (ABI) GCC в области передачи параметров функций.

У вас есть два варианта обновления системы. Первый способ быстрее и требует использования программы `revdep-rebuild` из пакета `gentoolkit`, а во втором вся система пересобирается с нуля, чтобы задействовать новые возможности GCC. Ваше дело, какой из способов выбрать. В большинстве случаев первого способа достаточно.

## Использование `revdep-rebuild`

Если вы выбрали этот способ, нужно сначала установить `gentoolkit`, если вы еще этого не сделали. Затем обновите GCC и переключите систему на новый компилятор. Также пересоберите пакет `libtool`, чтобы обеспечить пригодность программ для компиляции.

**Листинг 3.1:** Установка `gentoolkit` и обновление GCC

```
# emerge -an gentoolkit
# emerge -uav gcc
(вместо "i686-pc-linux-gnu-3.4.5" укажите обновленную
 версию GCC и настройки CHOST)
# gcc-config i686-pc-linux-gnu-3.4.5
# source /etc/profile
```

(пересборка `libtool`)

```
# emerge --oneshot -av libtool
```

Теперь посмотрите, какие пакеты собирается пересобирать `revdep-rebuild`. Затем запустите `revdep-rebuild` на собственно пересборку. Это займет некоторое время, так что потерпите.

**Листинг 3.2:** Использование `revdep-rebuild`

```
# revdep-rebuild --library libstdc++.so.5 -- -p -v
# revdep-rebuild --library libstdc++.so.5
```

**Примечание:** Возможно, у вас появятся проблемы с несуществующими версиями пакетов из-за того, что они устарели или замаскированы. В этом случае можно при запуске `revdep-rebuild` указать параметр `--package-names`. Это заставит пакеты пересобираются, основываясь на именах пакетов, вместо точного имени и версии.

Для обеспечения совместимости с более старыми двоичными приложениями C++ и любыми пакетами, которые `revdep-rebuild` мог пропустить, надо установить пакет `sys-libs/libstdc++-v3` до того, как удалять GCC 3.3 из своей системы.

**Листинг 3.3:** Установка `libstdc++-v3` и удаление GCC

```
# emerge --oneshot sys-libs/libstdc++-v3
# emerge -aC =sys-devel/gcc-3.3*
```

## Использование `emerge -e`

Этот гораздо более медленный способ пересобирает всю систему, гарантируя, что все пересобирается новым компилятором, и, таким образом, он безопаснее. Сперва потребуется обновить GCC и `libtool`, и переключить систему на новый компилятор.

**Листинг 3.4:** Обновление GCC

```
# emerge -uav gcc
(вместо "i686-pc-linux-gnu-3.4.5" укажите обновленную
```

```
версию GCC и настройки CHOST)  
# gcc-config i686-pc-linux-gnu-3.4.5  
# source /etc/profile
```

```
(пересборка libtool)  
# emerge --oneshot -av libtool
```

Для обеспечения совместимости с более старыми двоичными приложениями C++, надо установить в систему пакет `sys-libs/libstdc++-v3`.

**Листинг 3.5:** Установка `libstdc++-v3`  

```
# emerge --oneshot sys-libs/libstdc++-v3
```

Теперь займемся пересборкой сначала пакетов `system`, а затем `world`. Это займет очень длительное время, зависящее от количества установленных пакетов: ведь будут пересобиаться все средства компиляции и поддерживающие системные файлы, а затем каждый пакет, находящийся в вашей системе. Это необходимо, чтобы все пакеты были гарантированно скомпилированы уже с новыми средствами компиляции, включая сами эти средства.

**Листинг 3.6:** Пересборка `system` и `world`  

```
# emerge -e system  
# emerge -e world
```

Теперь можно удалить старые версии GCC, не вызывая проблем:

**Листинг 3.7:** Очистка  

```
# emerge -aC =sys-devel/gcc-3.3*
```

## 4. Обновление GCC на новой установке

### Введение

Обновление GCC в системе после установки из архива третьей стадии (`stage3`) — дело простое. Отсутствие изобилия установленных программ, которые ссылаются на старую версию GCC — это преимущество пользователей, только что установивших систему. В следующем примере показывается обновление с GCC-3.3 на 3.4 или более новые версии. При обновлении с других версий GCC будут кое-какие отличия. Например, имена библиотек, используемые ниже для `revdep-rebuild`, относятся к версии GCC 3.3, также, как и необходимость установки `libstdc++-v3`.

Пока пользователь не внес каких-либо изменений в систему, для получения системы с новой версией GCC нужно всего несколько шагов. Также, как и при обновлении с GCC-3.3 до 3.4, у пользователя есть две возможности. Но, в отличие от обновления с GCC-3.3 до 3.4, здесь обновление проще, так как различий между способами меньше. Первый способ быстрее, и задействует программу `revdep-rebuild` из пакета `gentoolkit`, подобно вышеописанной процедуре обновления. Использование `revdep-rebuild` предполагает пересборку только тех пакетов, которые действительно ссылаются на библиотеки GCC, тогда как при втором способе система полностью перекомпилируется с новой версией GCC, что занимает намного больше времени. Второй способ никогда не потребуется, и описан только для полноты картины.

Приведенные первые шаги одинаковы для обоих способов, и должны делаться в любом случае.

#### Листинг 4.1: Обновление GCC

```
# emerge -uav gcc
(вместо "i686-pc-linux-gnu-3.4.5" укажите обновленную
 версию GCC и настройки CHOST)
# gcc-config i686-pc-linux-gnu-3.4.5
# source /etc/profile
```

(пересборка libtool)

```
# emerge --oneshot -av libtool
```

Для обеспечения совместимости с более старыми двоичными приложениями C++, надо установить в систему пакет sys-libs/libstdc++-v3.

#### Листинг 4.2: Установка libstdc++-v3

```
# emerge --oneshot sys-libs/libstdc++-v3
```

### Использование revdep-rebuild

Этот способ требует, чтобы вы сначала установили пакет gentoolkit, если это еще не сделано. Затем запустите программу revdep-rebuild, чтобы просканировать установленные пакеты, найти и пересобрать нужные.

#### Листинг 4.3: Установка gentoolkit и запуск revdep-rebuild

```
# emerge -an gentoolkit
# revdep-rebuild --library libstdc++.so.5 -- -p -v
# revdep-rebuild --library libstdc++.so.5
```

**Примечание:** Возможно, у вас появятся проблемы с несуществующими версиями пакетов из-за того, что они устарели или замаскированы. В этом случае можно при запуске revdep-rebuild указать параметр --package-names. Это заставит пакеты пересобираться, основываясь на именах пакетов, вместо точного имени и версии.

### Использование emerge -e

Этот способ, будучи гораздо медленнее, пересобирает всю систему, гарантируя, что все пересобрано новым компилятором. Это необязательно, но допустимо, если вы также изменяете переменную среды CFLAGS или другие переменные make.conf, которые влияют на компиляцию системы.

Так как эти действия выполняются сразу после первоначальной установки, не потребуется перекомпилировать world, как мы поступили бы для обновления компилятора в ранее установленной системе. Однако, для пущей уверенности в том, что обновлены все пакеты, можно запустить обновление цели world вместо system.

#### Листинг 4.4: Пересборка system

```
# emerge -e system
```

### Очистка

Теперь можно удалить старые версии GCC, не вызывая проблем. Выражение ВАША-НОВАЯ-ВЕРСИЯ-GCC замените номером версии, на которую вы перешли:

**Листинг 4.5:** Очистка

```
# emerge -aC "<sys-devel/gcc-ВАША-НОВАЯ-ВЕРСИЯ-GCC"
```

## 5. Обычные грабли

Важно на время обновления отключить distcc. Смешение версий компилятора на разных узлах вызовет проблемы при сборке. Это не относится к ccache, так как объекты кэша будут в любом случае сделаны действительными.

Всегда используйте одну и ту же версию GCC для своего ядра и дополнительных модулей ядра. Как только вы пересоберете world с новым GCC, внешние модули (например, app-emulation/qemu-softmmu) не смогут загрузиться. Пожалуйста, чтобы это исправить, пересоберите свое ядро новой версией GCC.

Если вы обновляете GCC на машине SPARC, обязательно еще раз запустите silo -f после пересборки world, чтобы избежать возможных проблем.

Распространенные сообщения об ошибках

Если ваша система жалуется на что-то вроде libtool: link: `/usr/lib/gcc-lib/i686-pc-linux-gnu/3.3.6/libstdc++.la' is not a valid libtool archive, запустите /sbin/fix\_libtool\_files.sh 3.3.6 (замените «3.3.6» на номер версии из сообщения об ошибке).

Если вы увидели error: /usr/bin/gcc-config: line 632: /etc/env.d/gcc/i686-pc-linux-gnu-3.3.5: No such file or directory, тогда попробуйте удалить /etc/env.d/gcc/config-i686-pc-linux-gnu и снова запустить gcc-config, следом указав source /etc/profile. Делайте это только тогда, когда у вас не настроены никакие кросс-компиляторы.

Если при emerge -e system или emerge -e world не собирается пакет, выполнение можно продолжить командой emerge --resume. Если ошибка появляется снова и снова, пропустите пакет, указав emerge --resume --skipfirst. Не запускайте параллельно другие экземпляры emerge, чтобы не потерять информацию, нужную для возобновления.

Если во время обновления компиляторов встретится ошибка spec failure: unrecognized spec option, попробуйте откатиться на свой компилятор по умолчанию, убрать переменную GCC\_SPECS и снова обновить компилятор:

**Листинг 5.1:** Восстановление первичной настройки компилятора

```
# gcc-config 1
# source /etc/profile
# unset GCC_SPECS
# emerge -uav gcc
```

# Gentoo и X.Org

## Описание настройки МОНОЛИТНОГО X-сервера версий 6.X

Ссылка на оригинал (от 13.06.07):

<http://www.gentoo.org/doc/ru/xorg-config.xml>

С версии: 1.0

### 1. Что такое X Window Server?

#### Графический интерфейс против командной строки

Обычного пользователя могут испугать мысли о необходимости печатать в командной строке. Почему бы ему не прокладывать свой путь на территории свободы выбора Gentoo (и Linux вообще), просто «водя мышкой»? Ну, \*большая улыбка\*, конечно, вы сможете это сделать :-). В Linux предлагается широкое разнообразие пользовательских интерфейсов и сред, которые можно установить поверх установленной у вас системы.

Это один из самых больших сюрпризов, с которым сталкиваются новички: графический интерфейс пользователя — это всего лишь прикладная программа, выполняемая в системе. Он не является частью ядра Linux или каких-либо других компонентов системы. Это — мощный инструмент, позволяющий полностью задействовать графические возможности вашего компьютера.

Так как стандартизация очень важна, для организации прорисовки и перемещения окон по экрану, взаимодействия пользователя с мышью и клавиатурой, а также других важных базовых аспектов, был создан стандарт под названием X Window System, сокращённо X11 или просто X. Он используется в Unix, Linux и других Unix-подобных операционных системах во всем мире.

Приложение, которое дает пользователям Linux возможность запускать графические интерфейсы пользователя и использует стандарт X11, называется Xorg-X11, ответвление проекта XFree86. Проектом XFree86 было решено использовать лицензию, возможно, несовместимую с лицензией GPL, поэтому рекомендуется использовать Xorg. Пакет XFree86 больше не поддерживается в официальном дереве портежей.

#### Проект X.org

Проектом X.org создана и поддерживается бесплатно распространяемая реализация системы X11 с открытым исходным кодом. Это открытая инфраструктура рабочей станции, основанная на X11.

Xorg обеспечивает интерфейс между устройствами вашего компьютера и нужными

вам графическими программами. Кроме того, Xorg является полностью сетевым, то есть вы можете запускать программу на одной системе, а отображать ее — на другой.

## 2. Установка Xorg

### Использование emerge

Хватит болтовни, не перейти ли нам к делу? Чтобы установить **Xorg**, вам просто нужно запустить **emerge xorg-x11**. Установка **Xorg** займёт достаточно времени, так что вы можете перекусить, дожидаясь ее завершения.

Перед тем, как установить Xorg, в файле `/etc/make.conf` вам понадобится настроить две очень важные переменные.

Первая из них — **VIDEO\_CARDS**, которая используется для выбора драйвера для используемой вами видеокарты, и связана с ее типом и маркой. Одним из распространённых значений является **nvidia** для карт nVidia или **fglrx** для карт ATI Radeon. Это — закрытые драйверы, предоставленные nVidia и ATI. Если вы хотите использовать версии с открытым исходным кодом, используйте **nv** вместо **nvidia**, но помните, эта версия не обеспечивает 3D-ускорение. Аналогично, используйте **radeon** для видеокарт от ATI. В **VIDEO\_CARDS** можно указывать несколько драйверов, разделённых пробелами.

Вторая переменная, **INPUT\_DEVICES**, используется для указания драйверов устройств ввода. В большинстве случаев связки `keyboard mouse` достаточно.

Теперь нужно решить, какие драйверы использовать, и добавить нужные параметры в файл `/etc/make.conf`:

**Листинг 2.1:** Пример строк в `make.conf`  
(для поддержки мыши и клавиатуры)  
`INPUT_DEVICES="keyboard mouse"`  
(для видеокарт nVidia)  
`VIDEO_CARDS="nvidia"`  
(ИЛИ для видеокарт ATI Radeon)  
`VIDEO_CARDS="fglrx"`

Дополнительные сведения по настройке карт nVidia и ATI находятся в руководстве по видеокартам nVidia в Gentoo Linux и Gentoo Linux ATI FAQ. Если вы не знаете, какие драйверы выбрать, ознакомьтесь с этими руководствами.

Примечание: Если у вас рекомендуемые настройки не работают, следует запустить `emerge -pv xorg-x11`, проверить все доступные параметры и указать соответствующие вашей системе. Для различных версий Xorg на различных архитектурах эти переменные могут выводиться командой `emerge -pv xorg-server` вместо приведенной выше. Например, для архитектуры x86 и `xorg-x11-7.0`:

**Листинг 2.2:** Просмотр всех доступных параметров драйверов  
`# emerge -pv xorg-x11`

These are the packages that would be merged, in order:

Calculating dependencies... done!

```
[ebuild R ] x11-base/xorg-x11-7.0-r1 USE="-3dfx" INPUT_DEVICES="keyboard
mouse -acecad -aiptek -calcomp -citron -digitaledge -dmc -dynapro -elo2300
-elographics -evdev -fpit -hyperpen -jamstudio -joystick -magellan -magictouch
-microtouch -mutouch -palmax -penmount -spaceorb -summa -synaptics -tek4957
-ur98 -vmmouse -void -wacom" VIDEO_CARDS="nvidia -apm -ark -chips -cirrus -cyrix
-dummy -fbdev -fglrx -glint -i128 -i740 -i810 -imstt -mach64 -mga -neomagic -nsc
-nv -r128 -radeon -rendition -s3 -s3virge -savage -siliconmotion -sis -sisusb
-tdfx -tga -trident -tseng -v4l -vesa -vga -via -vmware -voodoo" 0 kB
```

После установки всех необходимых переменных можно перейти к установке пакета Xorg.

**Листинг 2.3:** Установка Xorg  
# emerge xorg-x11

После завершения установки, чтобы продолжить, вам наверняка придётся обновить некоторые переменные среды. Для установки значений просто запустите `env-update`, а затем — **source /etc/profile**.

**Листинг 2.4:** Обновление переменных среды  
# env-update  
# source /etc/profile

### 3. Настройка Xorg

#### Файл `xorg.conf`

Конфигурационный файл Xorg называется `xorg.conf` и находится в папке `/etc/X11`. В пакет Xorg-X11 входит его пример под названием `/etc/X11/xorg.conf.example`, который можно использовать при создании своей собственной конфигурации. Он подробно прокомментирован, но если вы нуждаетесь в подробном описании синтаксиса, не стесняйтесь обратиться к страницам справки:

**Листинг 3.1:** Чтение страницы справки `xorg.conf`  
# man 5 xorg.conf

Счастливого чтения всем желающим. Мы, конечно же, не из их числа, поэтому перейдем к выяснению, как можно создать файл автоматически.

#### По умолчанию: автоматическая генерация `xorg.conf`

Xorg способен самостоятельно подобрать большинство параметров за вас. Скорее всего, вам потребуется всего лишь изменить несколько строк, чтобы установить желаемое разрешение экрана. Если вы заинтересованы в более глубокой настройке, обязательно просмотрите ресурсы, указанные в конце этой главы. Но сначала давайте создадим (надеемся, работающий) конфигурационный файл Xorg.

**Листинг 3.2:** Генерация файла `xorg.conf`  
# Xorg -configure



Обязательно прочтите последние строки, выводимые после завершения опроса оборудования Xorg. Если вы увидите, что где-то Xorg не удалось правильно опознать устройства, то вам придётся править файл `xorg.conf` вручную. Если же всё прошло гладко, Xorg должен сообщить вам, что создан файл и готов для тестирования файл `/root/xorg.conf.new`. Давайте его испытаем :)

**Листинг 3.3:** Тестирование файла `xorg.conf.new`  
# X -config /root/xorg.conf.new

Если всё в порядке, вы увидите чёрно-белый узор. Проверьте, работает ли мышь, и подходит ли разрешение экрана. Вы вряд ли точно угадаете разрешение, но всё же заметите слишком низкое. Выйти можно в любой момент, нажав комбинацию клавиш `Ctrl+Alt+Backspace`.

### Альтернатива: полуавтоматическая генерация `xorg.conf`

В Xorg есть утилита `xorgconfig`, задающая различные вопросы о вашей системе (о графическом адаптере, клавиатуре и т.п.). Основываясь на ваших ответах, она создаст файл `xorg.conf`.

**Листинг 3.4:** Полуавтоматическая генерация `xorg.conf`  
# xorgconfig

Другая утилита, также входящая в пакет Xorg — `xorgcfg`, которая сначала пытается выполнить `Xorg -configure`, а затем запускает X-сервер для более тонкой настройки.

**Листинг 3.5:** Использование `xorgcfg`  
# xorgcfg  
(если X даст сбой, или настройка завершится неудачей, попробуйте:)  
# xorgcfg -textmode

### Копирование в `xorg.conf`

Теперь давайте скопируем `xorg.conf.new` в `/etc/X11/xorg.conf`, чтобы не приходилось постоянно запускать `Xorg -config`: набирать просто X или `startx` гораздо легче :)

**Листинг 3.6:** Копирование поверх `xorg.conf`  
# cp /root/xorg.conf.new /etc/X11/xorg.conf

### Использование `startx`

Теперь попробуйте ввести **startx**, чтобы запустить свой X-сервер. **startx** — это сценарий, запускающий сеанс X, то есть серверы X, а поверх них — некоторые графические приложения. Он решает, какие приложения запустить, исходя из следующей логики:

если в домашнем каталоге есть файл с именем **.xinitrc**, то выполняются команды, перечисленные в нём

в противном случае считывается значение переменной `XSESSION` и запускается один из указанных в `/etc/X11/Sessions/` сеансов (указать значение `XSESSION` по умолчанию, для всех пользователей системы, можно в файле `/etc/rc.conf`)

если вышеуказанное завершилось неудачей, производится откат к простейшему диспетчеру окон, обычно twm.

**Листинг 3.7:** Запуск X  
# startx

Если вы увидели уродливый, отвратительный, омерзительный, безобразный диспетчер окон, то это — twm. Чтобы завершить сеанс twm, наберите exit или нажмите Ctrl-D в одном из терминалов xterm. «убить» сеанс X также можно, нажав комбинацию клавиш Ctrl+Alt+Backspace, чтобы «выгнать его с позором» — наверно, такой вариант годится не всегда. Хотя это не причинит тяжёлого вреда :)

## 4. Настройка xorg.conf

### Установка разрешения экрана

Если вы чувствуете, что разрешение экрана неподходящее, вам потребуется проверить два раздела конфигурации. Прежде всего, разделе **Screen**, где перечисляются варианты разрешения экрана, с которыми может запускаться X-сервер. По умолчанию в этом разделе может вообще не быть никаких строк о разрешении экрана. В таком случае **Xorg** оценивает допустимое разрешение на основе данных из другого раздела: **Monitor**.

При этом Xorg для вычисления правильных вариантов разрешения использует значения **HorizSync** (частота строк) и **VertRefresh** (частота кадров) из раздела **Monitor**. Пока что оставьте эти параметры как есть. Лишь в том случае, когда изменения в разделе Screen (которые мы опишем чуть ниже) не помогают, вам придется заглянуть в технические характеристики своего монитора и указать правильные значения. Также можно воспользоваться программой, определяющей технические характеристики вашего монитора, например, sys-apps/ddcinfo-knoppix.

**Предупреждение:** Не меняйте «просто так» значения этих двух относящихся к монитору переменных, не сверившись с техническими характеристиками своего монитора. Неправильные значения могут привести в лучшем случае к срыву синхронизации, а в худшем случае — к сгоревшему монитору.

Теперь давайте поменяем значения разрешения. В следующем примере, взятом из /etc/X11/xorg.conf, мы добавим строчки Modes (режимы) и DefaultDepth (цветность), чтобы X-сервер по умолчанию запускался в режиме 24 бит при 1024x768 разрешении экрана. Особо не обращайтесь внимания на значения — это просто пример и, скорее всего, они будут отличаться от настроек вашей системы.

**Листинг 4.1:** Изменение раздела Screen /etc/X11/xorg.conf

```
Section "Screen"
    Identifier "Default Screen"
    Device     "S3 Inc. ProSavage KN133 [Twister K]"
    Monitor    "Generic Monitor"
    DefaultDepth 24
    # несколько строк пропущены для наглядности
```

```
SubSection "Display"
    Depth 24
    Modes "1024x768"
EndSubSection
EndSection
```

Запустите X (startx), чтобы обнаружить, что сервер использует желаемое разрешение :)

## Настройка клавиатуры

Чтобы настроить X на использование национальных раскладок, найдите раздел InputDevice (устройство ввода), определяющий настройки клавиатуры, и добавьте параметр XkbLayout с указанием необходимой раскладки. Для примера, покажем, как добавить бельгийскую раскладку. Просто измените код страны на свой:

### Листинг 4.2: Changing the keyboard layout

```
Section "InputDevice"
    Identifier "Generic Keyboard"
    Driver "keyboard"
    Option "CoreKeyboard"
    Option "XkbRules" "xorg"
    Option "XkbModel" "pc105"
    Option "XkbLayout" "be"
EndSection
```

## Настройка мыши

Если ваша мышь не работает, сначала придётся выяснить, обнаружена ли она вообще ядром. Мыши (в качестве устройств) выглядят как /dev/input/mouse0 (или /dev/input/mice, если вы хотите использовать несколько мышек). В некоторых случаях используется название /dev/psaux. В любом случае, вы можете убедиться, что устройство соответствует вашей мыши, просмотрев вывод соответствующего файла устройства, одновременно передвигая мышь. В большинстве случаев на экране должна появиться бессмыслица. Для остановки вывода нажимайте Ctrl-C.

### Листинг 4.3: Проверка файла устройства

```
# cat /dev/input/mouse0
```

(не забудьте нажать Ctrl-C для завершения)

Если ваша мышь не обнаружена, проверьте, все ли необходимые модули загружены.

Если же ваша мышь найдена, впишите устройства в соответствующий раздел InputDevice. В следующем примере видно, как мы устанавливаем еще два параметра: Protocol (определяет протокол, используемый мышью; у большинства пользователей — PS/2 или IMPS/2) и ZAxisMapping (позволяющий задействовать колесико, если есть).

### Листинг 4.4: Изменение настройки мыши в Xorg

```
Section "InputDevice"
    Identifier "TouchPad Mouse"
    Driver "mouse"
    Option "CorePointer"
    Option "Device" "/dev/input/mouse0"
```

```
Option "Protocol" "IMPS/2"
Option "ZAxisMapping" "4 5"
EndSection
```

Запустите **startx** и порауйтесь результату :). Поздравляем, теперь (надеемся) у вас есть работающий Xorg. Следующий шаг — удалить этот ужасный облегченный диспетчер окон, и использовать «нафаршированный» возможностями диспетчер окон (или даже рабочую среду), такой как KDE или GNOME, но это не входит в рамки данного руководства :)

## 5. Ресурсы

### Создание и детальная настройка xorg.conf

Прежде всего, в **man 5 xorg.conf** дается краткое, но полное описание синтаксиса конфигурационного файла. Обязательно держите эту страницу открытой в соседнем терминале при редактировании своего конфигурационного файла!

Второй источник информации на вашей системе — каталог `/usr/X11R6/lib/X11/doc`, где находятся различные описания README для конкретных графических чипсетов.

В сети также есть много сведений о настройке xorg.conf. Мы перечислим лишь некоторые из них; обязательно погуглите, чтобы найти другие :). Так как в большинстве параметров xorg.conf и XF86Config (конфигурационного файла для XFree86) используется одинаковый синтаксис, а информации о XFree86 больше, мы ссылаемся и на такие ресурсы.

The XFree Local Multi-User HOWTO

An Introduction to XFree 4.x Криса Хаузера (Chris Houser)

### Другие источники

Если вы хотите обновить свою систему со старым монолитным Xorg на новый модульный Xorg 7, прочтите руководство по переходу на модульный X (англ.)

Дальнейшая информация по настройке различных пакетов, для работы в среде X находится в разделе графическая среда Gentoo нашей документации.

## **Описание настройки МОДУЛЬНОГО X-сервера версий 7.X (Англ.)**

**Ссылка на оригинал:**

[http://gentoo-wiki.com/HOWTO\\_Modular\\_Xorg](http://gentoo-wiki.com/HOWTO_Modular_Xorg)

**С версии: 1.0**

### **Introduction**

This is a guide to Modular Xorg, mainly dealing with special circumstances and troubleshooting.

This article is currently being maintained by AllenJB. If you have suggestions for additions or fixes, please post them to the discussion and bugs page.

Please see also the official Modular Xorg Upgrade Guide.

### **Preparation**

Firstly it'll be useful to know what version of Xorg you're going to be emerging. If you don't know, run **emerge -p xorg-x11** to find out.

### **Nvidia Drivers**

The old `nvidia-glx` and `nvidia-kernel` packages have been replaced by a single `nvidia-drivers` package. If you're still using the old packages, you'll want to upgrade to the new package first, as the old ones will not work with modular Xorg, by running:

```
# emerge -Ca nvidia-glx nvidia-kernel
# emerge nvidia-drivers
#eselect opengl set nvidia
```

### **Masking Xorg 7.1 (optional)**

If your system is going to install `xorg-x11-7.1` and you wish to use `xorg-x11-7.0` instead you'll need to add the following to `/etc/portage/package.mask` (create it if it doesn't exist):

```
# Mask xorg 7.1
>=x11-base/xorg-x11-7.1
>=x11-base/xorg-server-1.1.0
>=x11-drivers/xf86-input-evdev-1.1.2-r1
>=x11-drivers/xf86-video-vesa-1.2.0
>=x11-drivers/xf86-input-mouse-1.1.1
>=x11-drivers/xf86-input-keyboard-1.1.0
>=x11-drivers/xf86-video-tdfx-1.2.0
```

## ***Backup***

Because Xorg is such a key part of a desktop system, you'll want to backup your current install before continuing. To do this we're going to use the quickpkg command from the gentoolkit package:

```
emerge -a gentoolkit  
quickpkg xorg-x11
```

If you want to find out more about the quickpkg tool, run: `man quickpkg`

Additionally, create a list of packages that have files in the Xorg specific directories by running:

```
if [[ ! -L /usr/X11R6 ]]; \  
then equery belongs /usr/X11R6 > ~/usr-x11r6-packages \  
&& rm -rf /usr/X11R6; fi
```

## ***Restoring the backup***

To restore your backed-up package run: `emerge --usepkgonly -a xorg-x11-6.8.2-r8`

If you can't remember the exact version you can find it out by running: `ls /usr/portage/packages/All/xorg-x11*`

You then want to re-emerge all the packages listed in `~/usr-x11r6-packages`.

## ***Uninstall old X.org***

**Warning:** If you are reading this from the same machine you will be upgrading X on, keep in mind two things:

After cleaning out the old X, and before the new X is installed, you will not be able to start any X applications as X is not technically installed. The applications depending on X currently running are just running off the memory of X. Closing them will not enable you to re-open them until the new X is installed.

Programs currently running in X may become unstable during the install. For example, firefox may unexpectedly crash due to not being able to find fonts which are no longer there, or other similar reasons.

It is hence highly recommended that you print out a copy of this guide (or save a tree and open this guide on a console (not in X)) before you begin. Your programs should go back to behaving normally once the install is complete, however you may want to restart X as soon as possible afterwards.

Uninstall your current xorg install by running: **`emerge -Ca xorg-x11`**

### ***/usr/X11R6 symlink***

Run **`ls -l /usr/X11*`** and check that **`/usr/X11R6`** is a symlink to `/usr`. It should look something like this

```
:  
lrwxrwxrwx 1 root root 6 Mar 27 2005 /usr/X11R6 -> ../usr
```

If it doesn't, delete it and recreate it by running: `cd /usr && ln -s ../usr X11R6`

Remove `/usr/lib/X11/xkb`

The following is a requirement of the xkeyboard-config package.

32-bit Users: Run `ls -l /usr/lib/X11/xkb` and if it exists, delete it with `rm -r /usr/lib/X11/xkb`

## Emergent Modular X.org

### Device Drivers

Modular Xorg introduces a new system for managing the driver packages for your devices. You need to set the appropriate **VIDEO\_CARDS** and **INPUT\_DEVICES** flags in `/etc/make.conf`. To see what flags are available, run **emerge -upv xorg-x11**. On a typical machine with an Nvidia video card, it might look like this:

**File: `/etc/make.conf`:**

```
INPUT_DEVICES="keyboard mouse"  
VIDEO_CARDS="nv nvidia vesa"
```

The **nv** driver is the **open source** unofficial driver that lacks 3D acceleration. The **nvidia** driver is the **official nvidia-drivers package**.

On an ATI System, the **VIDEO\_CARDS** line should be like this (unless you don't wish to use proprietary drivers `fglrx`):

**File: `/etc/make.conf`**

```
VIDEO_CARDS="radeon fglrx vesa"
```

Add the **dri USE** flag to `/etc/make.conf` to enable **Direct Rendering support**, which most graphics drivers use.

## Installation

Now run **emerge -Dvat xorg-x11** to install modular Xorg.

A few useful X11 tools that used to be included in the 6.8 build are now their own packages, you can emerge all of them now by running: `emerge -DuNav xev xdpinfo xvinfo xset xdriinfo xprop`

If you would like some extra fonts, you can emerge them now also:

```
emerge -DuNav font-adobe-100dpi font-adobe-75dpi font-adobe-utopia-100dpi \
font-adobe-utopia-75dpi font-bh-100dpi font-bh-75dpi font-bh-type1 \
font-bh-lucidatypewriter-100dpi font-bh-lucidatypewriter-75dpi \
font-bitstream-100dpi font-bitstream-75dpi ttf-bitstream-vera \
corefonts sharefonts freefonts font-ibm-type1
```

## Keyboard Mappings

There may be people who will search for the `ca_enhanced` keyboard mapping in Xorg 7.0. It has not disappeared, it just changed names to `ca(fr)`, as with this keyboard configuration:

**File: `/etc/X11/xorg.conf`:**

```
Section "InputDevice"
    Identifier "Keyboard1"
    Driver      "kbd"
    Option "AutoRepeat" "500 30"
    Option "XkbRules" "xorg"
    Option "XkbModel" "pc105"
    Option "XkbLayout" "ca(fr)"
EndSection
```

Notice that Option "XkbRules" "xfree86" won't work for modular xorg-x11. You have to change it to "xorg", or many keys will not produce the desired effect.

In case you defined a custom xkb model (e.g. if you added a diNovo keyboard), check if you modified `/usr/share/X11/xkb` and not `/usr/lib/X11/xkb`.

If your X server is restarting each time you pressed a non dead key, add the following link:  
**# ln -s /usr/share/X11/XKeysymDB /usr/lib/X11/XKeysymDB**

## OpenMotif fails to emerge

If building of openmotif fails run: `emerge -DuNav --oneshot printproto xbitmaps libXp`

Then restart the emerge using: **`emerge --resume`**

## GL Issues

Problems building **xorg-server**, **glx**, and other **GL-related** packages in modular X may be due to a multilib issue.

The first thing to try is to update `eselect-opengl` to at least 1.0.3.

Next check your opengl implementation with `eselect opengl list`. If `xorg-x11` isn't the only option, you probably want the alternate option, so change it with: `eselect opengl set <implementation>`



## **xorg-server tries to build every single input driver**

**Solution:** You need to specify which drivers you want by modifying `/etc/make.conf` to reflect your hardware. If nothing is specified, it will build them all!

**Example:** File: `/etc/make.conf`  
`INPUT_DEVICES="keyboard mouse synaptics evdev"`

## **Mouse Speed Uncontrollable with evdev Driver**

Please see TIP Setting Mouse Speed Through Command Line for more information.

## **Mouse Wheel No Longer Works**

Please see the Modular XOrg section of HOWTO Advanced Mouse for more extensive information.

## **Xorg won't start due to fglrx or nvidia module missing**

Re-emerge the appropriate drivers (`ati-drivers` or `nvidia-drivers`) to get X running again after the install. Make sure you recompile the the drivers with same compiler you used for the kernel, otherwise you'll get message about the module format being incompatible when you try to load the `fglrx` kernel module.

## **Matrox G550, Xinerama and mga driver**

The install script of the binary drivers available from Matrox (v4.4 at the time of writing) copies the driver files to  
**`/usr/lib/modules/drivers`**

instead of  
**`/usr/lib/xorg/modules/drivers`**

Copy the correct files (`mga_drv.so/mga_hal_drv.so`) manually to get Xinerama working again.

## **Problems with the keyboard**

The AltGr key and/or other national special keys may stop working after an update to modular X. To fix this, unmerge the new `xkeyboard-config` and re-emerge the `xkbdata` package which contains correct translations for that key.

If the above fix doesn't work, you can also try running: **`xmodmap -e "keycode 113 = Mode_switch"`**

Sometimes it can happen that GNOME / KDE keyboard layout switching doesn't work. To correct this, create `xkb` as a symbolic link in the `/usr/lib64/X11` lib directory with:

**`cd /usr/lib64/X11`**  
**`ln -s /usr/share/X11/xkb/ xkb`**

## Missing fonts

There are X11 fonts that may be in use but not available. See the Xorg and fonts HOWTO and install the listed fonts.

## Configuration file locations have changed

All of the configuration files and scripts in Xorg-6.8 were stored in `/etc/X11`, which was not standard. In modular Xorg, all of the configuration files are in the same place but configuration scripts and default files have moved to `/usr/lib/X11/` and `/usr/share/X11`.

When upgrading, since `/etc` is config-protected, your old configuration files would not have been removed -- this means you'll have some extra files in there that look right but don't do anything.

Also, these new locations are NOT config-protected, so when making changes to say, `app-defaults/XTerm-color`, be sure to add `CONFIG_PROTECT="/usr/share/X11/app-defaults"` to `/etc/make.conf`. Or when updating the init for `xm`, it would be a good idea to copy that file from `/usr/lib/X11/xm/Xsetup_0` to `/etc/X11/xm/` and update `/etc/X11/xm/xm-config` accordingly.

## Other problems and solutions

Please see Gentoo's bug tracker and the official modular X howto.

If none of the above helps generating a new `xorg.conf` may solve your problems.  
X -configure

## XkbOptions

Ссылка на оригинал: <http://ru.gentoo-wiki.com/XkbOptions>

С версии: 1.0

### Параметры "XkbOptions"

Варианты переключателей между языками в файле `/etc/X11/xorg.conf`

1. **grp:toggle** - переключение правым Alt;
2. **grp:shift\_toggle** - двумя клавишами shift;
3. **grp:ctrl\_shift\_toggle** - ctrl+shift;
4. **grp:alt\_shift\_toggle** - alt+shift;
5. **grp:ctrl\_alt\_toggle** - ctrl+alt;
6. **grp:caps\_toggle** - переключение по CapsLock;
7. **grp:lwin\_toggle** - переключение по левой "Win" клавише на Windows клавиатуре (для X Window старше 4.2);
8. **grp:rwin\_toggle** - переключение по правой "Win" клавише на Windows клавиатуре (для X Window старше 4.2);
9. **grp:menu\_toggle** - переключение по клавише "Контекстное меню" на Windows клавиатуре (для X Window старше 4.2);
10. **grp:lctrl\_toggle** переключение по левой клавише Control
11. **grp:rctrl\_toggle** переключение по правой клавише Control
12. **grp:switch** - переключение по правому Alt только на момент нажатия
13. **grp:lwin\_switch** - переключение по левой клавише Windows только на момент нажатия
14. **grp:rwin\_switch** - переключение по правой клавише Windows только на момент нажатия
15. **grp:win\_switch** - переключение по любой клавише Windows только на момент нажатия

### Возможные значения grp\_led:

1. **grp\_led** - использовать индикаторы на клавиатуре для показания изменения группы
2. **grp\_led:num** - индикатор Num\_Lock показывает изменения группы
3. **grp\_led:caps** - индикатор Caps\_Lock показывает изменение группы
4. **grp\_led:scroll** - индикатор Scroll\_Lock показывает изменение группы

### Пример

Файл: `/etc/X11/xorg.conf`

```
...
Section "InputDevice"
    ...
    Option "XkbOptions" "grp:switch,grp:caps_toggle,grp_led:scroll"
    ...
EndSection
...
```

Дополнительная информация

`/etc/X11/xkb/rules/xorg.lst`

## HOWTO Xorg and Fonts (Рус.)

Ссылка на оригинал: [http://gentoo-wiki.com/HOWTO\\_Xorg\\_and\\_Fonts](http://gentoo-wiki.com/HOWTO_Xorg_and_Fonts)

Перевод: Батогов Е.В. а.к.а. JohnBat26

С версии: 1.3

Дата окончания перевода: 27.10.2007

### Назначение этого HOWTO

После изменения названия X сервера с Xfree на X.org, многие люди (включая меня) столкнулись с проблемами, связанными с отрисовкой и сглаживанием шрифтов, особенно это касается тех, кто использует LCD или TFT мониторы. Этот HOWTO делает шрифты значительно более интересными. В дополнение к этому, этот HOWTO объясняет как надо устанавливать шрифты.

### Общие сведения

Компьютерные шрифты в целом, и в Linux (X11) в частности, это довольно запутанное дело. Существует много различных форматов шрифтов, в диапазоне от старых растровых шрифтов до шрифтов **Truetype**. Помимо этого, много шрифтов имеют проблемы лицензирования и внутренние проблемы (на которые компания Apple, к счастью, пока не обращает внимания). А если Вы захотите включить сглаживание, то это только добавит путаницы.

Многие программы, особенно **ghostscript**, **mozilla**, **openoffice**, и **tex**, имеют свои собственные представления о шрифтах и их настройке. Это значит, что информация в этом how-to может быть не эффективна для того, чтобы Ваши шрифты заработали.

**Смотрите:** X11 имеет не одну подсистему шрифтов, а две! Старая "core X font subsystem" использует наименования типа: -misc-fixed-medium-r-normal--10-100-75-75-c-60-iso8859-1. Новая система шрифтов, также известная как "fontconfig", работает с библиотекой **Xft** и, как обещается, полностью заменит старую систему. К тому же, существует ещё и сервер шрифтов, называющийся "**xfs**".

### Замечание по поводу DPI

До настройки любого программного обеспечения, лучшее из того, что Вы можете сделать для получения хорошо выглядящих шрифтов на Linux, это использование LCD/CRT монитора по крайней мере не меньше, чем с 96 dpi (точек на дюйм). Для расчёта этого значения, возьмите горизонтальное разрешение и разделите его на ширину TFT/LCD экрана в дюймах. Любое значение меньше, чем 96 dpi приведет к значительным проблемам которые нельзя будет устранить. CRT

мониторы будут почти всегда отрисовывать шрифты лучше, чем LCD мониторы потому, что они имеют внутреннее сглаживание от сканирования электронного луча и экрана покрытого люминофором.

## Замечание по поводу сравнения шрифтов на Windows/Mac и Linux

Шрифты в Windows (clear type) и в Mac OS всегда будут выглядеть лучше, чем шрифты в Linux, когда LCD монитор имеет dpi меньше, чем 96. Когда dpi равняется 96 и более, тогда шрифты в Linux будут выглядеть также или лучше, чем в Windows или Mac OS. Конечно, все это верно, если вы установите пропатченный libXft, доступный в 'xeffects' overlay.

## Получение необходимых пакетов

Я предполагаю, что Вы уже правильно установили X.org (X.org версии 6.8.0-r4 и выше должен быть установлен с включенными USE флагами: **truetype** и **type1**). Я также полагаю, что Вы не запускаете X-сервер для множества клиентов и, следовательно, USE флаг: font-server выключен, хотя он может по-прежнему применяться в других случаях.

### Необходимые пакеты:

- \* x11-base/xorg-x11
- \* media-libs/freetype
- \* media-fonts/corefonts
- \* media-fonts/freefonts

Тем не менее я советую Вам также установить следующие полезные шрифты (**замечание:** media-fonts/unifont блокируют dev-util/eclipse-sdk):

- \* media-fonts/ttf-bitstream-vera
- \* media-fonts/artwiz-fonts
- \* media-fonts/sharefonts
- \* media-fonts/terminus-font
- \* media-fonts/unifont
- \* media-fonts/dejavu

**Замечание:** f28: Более красивые шрифты: media-fonts/liberation-fonts-ttf, "A GPL-2 Helvetica/Times/Courier заменяют TrueType font set, с уважением к Red Hat"

Активизируйте USE флаги и установите следующие пакеты:

```
# euse --enable truetype type1
# emerge --newuse --noreplace freetype corefonts freefonts artwiz-fonts sharefonts \
terminus-font ttf-bitstream-vera unifont dejavu xorg-x11
```

Для тех, кто уже установил Xorg и желает обновить все остальные пакеты, которые

уже установлены, я рекомендую изменить файл `/etc/make.conf` добавив следующие USE флаги:

```
USE="truetype type1"
```

Затем необходимо заново собрать все пакеты которые могут использовать эти флаги:

```
# emerge -avuDN world
```

**Замечание для Xorg 7.0:** Кажется Вам необходимо установить ещё и пакет "intlfonts".

```
# emerge intlfonts
```

Вы можете также увидеть это в обсуждении:

<http://forums.gentoo.org/viewtopic.php?p=3643250#3643250>

## Информация о хинтинге (Hinting)

**Хинтинг** является частью процесса отрисовки шрифта, в общем, хинтинг руководит растеризатором так, что он производит шрифт, гораздо лучше выглядящий на экране (особенно при малых размерах шрифта).

Отдавая должное возможным проблемам с патентами, FreeType имеет выбор — собираться с TrueType's Byte Code Interpreter (или BCI) или нет.

Без BCI (по умолчанию), FreeType использует свой собственный автоматический хинтер для улучшения качества отображения шрифтов малых размеров. Этот автохинтер ещё находится в стадии совершенствования и улучшается с каждой версией FreeType. Автохинтер обычно дает великолепные результаты. Однако, многие считают, что родное TrueType BCI производит лучшие результаты, чем автохинтер, т.к. в этом случае используется информация о хинтинге, содержащаяся в шрифте.

Если Вы живете в стране, где патенты на программное обеспечение не являются проблемой, Вы можете включить использование BCI, выключая USE флаг: **bindist**. Дополнительную информацию о freetype и патентах смотрите на: <http://freetype.sourceforge.net/patents.html>.

```
#echo "media-libs/freetype -bindist" >> /etc/portage/package.use
#emerge -av freetype
```

Если Вы считаете, что шрифты слишком размытые на Вашем LCD мониторе, попробуйте включить флаг **bindist** и посмотреть разницу, может быть это будет лучше в Вашем случае.

## Изменение конфигурационных файлов

После установки шрифтов, Вам необходимо изменить конфигурационные файлы в порядке распознавания их системой. Вот эти файлы:

1. `/etc/fonts/local.conf`;
2. `/etc/X11/xorg.conf`;
3. `~/.fonts.conf` .

Для проверки того, какие шрифты установлены, Вы можете использовать **fc-list** и **xlsfonts**. Для просмотра шрифтов, Вы можете использовать **gfontview**.

Что делать, если Ваши шрифты не показываются утилитой **fc-list**, что делать если Ваши шрифты не отображаются в **xlsfonts**, что делать если Ваши шрифты не появляются в **OpenOffice**. Что, наконец, делать, если шрифты не показываются в **ghostscript** и/или в **tex**:

**Смотрите файлы: `/etc/fonts/local.conf` и `~/.fonts.conf`**

Редактируйте конфигурационные файлы для системы шрифтов: **Xft**. Изменения в файле `/etc/fonts/local.conf` касаются всей системы; выборочные изменения в пользовательском конфигурационном файле `~/.fonts.conf` имеют приоритет над системными установками в большинстве случаев. Единственная опция, которая не может быть перекрыта в пользовательском файле — это настройка сглаживания (anti-aliasing setting).

В файле `/etc/fonts/local.conf` нет необходимости устанавливать все директории со шрифтами, кроме тех случаев, когда шрифты устанавливаются в нестандартный каталог. Файл `/etc/fonts/fonts.conf` включает `/usr/share/fonts` и некоторые другие директории. Другими словами, Вы не должны изменять этот файл, т.к. Xorg будет просматривать Ваш файл `~/.fonts.conf`, если он имеется.

Заметьте то, что файлы `/etc/fonts/local.conf` и `~/.fonts.conf`, являются XML-файлами. И их легко изменять в редакторах, которые поддерживают синтаксис XML . Следующий пример содержит множество опций, которые Вы можете захотеть использовать:

**Файл: `/etc/fonts/local.conf` или `~/.fonts.conf`:**

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<!-- /etc/fonts/local.conf file to configure system font access -->

<fontconfig>

<!-- Enable sub-pixel rendering -->

<!--
    <match target="font">
        <test qual="all" name="rgba">
            <const>unknown</const>
        </test>
        <edit name="rgba" mode="assign"><const>rgb</const></edit>
```

```

    </match>
-->

<!-- Use the Autohinter -->

    <match target="font">
        <edit name="autohint" mode="assign"><bool>true</bool></edit>
    </match>

<!-- Disable Autohinting for bold fonts -->

    <match target="font">
        <test name="weight" compare="more">
            <const>medium</const>
        </test>
        <edit name="autohint" mode="assign"><bool>false</bool></edit>
    </match>

<!-- Exclude/Include a range of fonts for Anti Aliasing -->

<!--
<match target="font">
    <test qual="any" name="size" compare="more">
        <double>9</double>
    </test>
    <test qual="any" name="size" compare="less">
        <double>14</double>
    </test>
    <edit name="antialias" mode="assign">
        <bool>true</bool>
    </edit>
</match>
-->

<!-- And/Or disable Anti Aliasing for a range on pixel-based size.
Disabling this using both methods seems to fix Firefox. -->

<!--
<match target="font">
    <test compare="less" name="pixelsize" qual="any">
        <double>20</double>
    </test>
    <edit mode="assign" name="antialias">
        <bool>false</bool>
    </edit>
</match>
-->

    <dir>/home/johnbat26/extrafonts</dir>

</fontconfig>

```

В представленном выше файле опция: "sub-pixel rendering" закомментирована потому, что она используется только на LCD мониторах. Если её включить на CRT



мониторе, то это заставит Xorg производить много дополнительной работы по отрисовке. Возможные значения: "none", "rgb", "bgr", "vrgb" и "vbgr". Для большинства LCD мониторов наиболее подходящим значением будет: "rgb", которое и использует приведенный выше пример.

Вы конечно захотите использовать какую-нибудь форму хинтинга. Gentoo поддерживает 2 формы хинтинга:

- **AutoHinter**

По умолчанию используется auto hinter из пакета FreeType, и это также является установкой по умолчанию в приведенном выше примере конфигурационного файла. Тем не менее, когда используется автохинтинг, тогда Вы можете захотеть выключить его использование на жирных шрифтах, иначе они могут стать слишком широкими. Пример выше как раз и делает это.

- **TrueType Byte Code Interpreter (BCI)**

Если Вы собрали FreeType с активацией интерпретатора байт-кода (TrueType Byte Code Interpreter) (см. выше), и желаете использовать BCI, тогда выключите auto hinter полностью. (Убедитесь, что `<edit name="autohint" mode="assign"><bool>true</bool></edit>` использует "false" вместо "true". Существуют патентные проблемы на использование TrueType BCI; если Вы желаете избежать их полностью, тогда не используйте BCI.

И KDE и GNOME имеют прекрасный графический интерфейс, в котором пользователь может выбрать уровень хинтинга: "легкий", "средний", "сильный" ('light,' 'medium,' and 'strong'). Это позволяет обратиться к FreeType 'auto hinter', а не к BCI. Другими словами 'легкий,' 'средний,' и 'сильный' хинтинг, позволяют всем шрифтам выглядеть одинаково, когда BCI используется вместо автохинтера (auto hinter).

Вторая опция с окончания файла, также закомментирована, так как некоторые люди предпочитают исключать или включать набор размеров шрифтов для применения сглаживания к ним. Это общий способ, чтобы сделать Ваши шрифты выглядящими очень похоже на шрифты по умолчанию в Windows(R). Когда Вы отключаете сглаживание, убедитесь также, что выключена опция "sub-pixel rendering", - для достижения наилучших шрифтов без сглаживания.

Наконец, последняя опция, связанный с пикселем, размер шрифта (pixel-based font sizes)... Пожалуйста, читайте комментарии к этой опции!

Последняя строка в примере пользовательских шрифтов, это изменение директории по умолчанию (/usr/share/fonts/) для местоположения шрифтов. Скорее всего Вам не нужна данная строка, если так, то можете просто удалить её.

Помните, что все изменения в этом файле распространяются на всю систему и будут перекрыты ~/.fonts.conf, исключая только установку anti-aliasing setting, которая не может быть перекрыта, если установлена в этом файле. Также заметьте, что синтаксис файла: /etc/fonts/local.conf одинаков с локальным пользовательским конфигурационным файлом: ~/.fonts.conf, и, следовательно, Вы можете копировать глобальный файл, и просто переименовать его.

## /etc/X11/xorg.conf

Когда Вы настраиваете X ("иксы"), важно убедиться, что они правильно понимают размер Вашего экрана, и так же важно, что Вы запускаете LCD монитор на его родном разрешении, которое используется для вычисления горизонтального и вертикального DPI экрана — шрифты в GTK+ приложениях кажутся особенно безобразными на основе неправильной информации о экране.

Это может также привести к неправильному определению размера шрифтов в различных приложениях. Некоторые шрифты будут корректного размера в зависимости от используемого метода, а другие будут иметь неправильные размеры. Я заметил большие различия, когда просматривал страницы в Firefox. Если на странице использовался CSS, то она отображалась хорошо, иначе если использовался чистый HTML, то шрифты получаются либо огромные либо небольшие в зависимости от направления неправильной установки DPI. Я также обнаружил, что DPI 96 является хорошим размером.

Установка ширины и высоты экрана (в миллиметрах) используя следующую инструкцию:

**DisplaySize** <width> <height>

в секции **Monitor** файла: **/etc/X11/xorg.conf**, решает эту проблему.

### File: /etc/X11/xorg.conf

```
Section "Monitor"
    Identifier      "Monitor0"
    :
    DisplaySize     340 270
    :
EndSection
```

Простейший путь для измерения ширины и высоты— это воспользоваться линейкой. Например, монитор в 15 дюймов имеет размеры приблизительно 340мм x 270 мм, а 19 дюймов — 380мм x 305 мм.

Значения **DisplaySize** могут быть вычислены по формуле  **$25.4 \times \text{width} / \text{dpi}$**  или, соответственно,  **$25.4 \times \text{height} / \text{dpi}$** , 25,4 мм на дюйм. Например, если использовать DPI равным 75 на разрешении 1600x1200, то расчет будет следующий:  **$25.4 \times 1600 / 75 = 542$**  и  **$25.4 \times 1200 / 75 = 406$** , и в результате устанавливаем **DisplaySize 542 406**.

Другой способ для подсчёта этих значений, это попросить X сервер вычислить их для Вас. Запустите его используя команду **startx -- -dpi 96**, где **96** это требуемое значение для DPI, и проверьте размер:

```
#xdpyinfo | grep dimension
dimensions: 1400x1050 pixels (XXXxYYY millimeters)
```

Используйте эти значения: XXX и YYY для параметра **DisplaySize**.

Ещё один способ обнаружить размерность экрана, если Ваше аппаратное

обеспечение предоставляет эту информацию (Вам может потребоваться установить **emerge -tav xrandr** сначала) это использовать:

```
# xrandr --query
SZ:  Pixels      Physical      Refresh
*0  1680 x 1050  ( 431mm x 272mm ) *60
Current rotation - normal
Current reflection - none
Rotations possible - normal
Reflections possible - none
```

Я достиг лучших результатов, используя измерения, полученные с помощью этого калькулятора: <http://www.raydreams.com/prog/dpi.aspx>

Если вы используете драйверы nVidia, то наилучшим способом следовало бы добавить эти опции в секцию device. Это также намного лучше работает с двумя мониторами:

Файл: **/etc/X11/xorg.conf**

```
....
Option "UseEdidDpi" "FALSE"
Option "DPI" "96 x 96"
....
```

Далее, у нас есть список наших каталогов со шрифтами в файле **/etc/X11/xorg.conf**. Пути к шрифтам ищутся в том порядке, в котором они расположены в списке, поэтому мы должны сделать небольшие изменения.

Достаточно много современных приложений и окружений рабочего стола, таких как Gnome и KDE используют утилиту **fontconfig**, так что только шрифты, которые действительно требуются X сервером (или сервером шрифтов) это **fixed** и **cursor**.

**/usr/share/fonts/misc:unscaled**

Первым путем, который мы хотим видеть в нашем списке, является: **/usr/share/fonts/misc**. Там находятся все важные **fixed** и **cursor** шрифты. Они являются критичными для правильной работы **Xorg**. Там же находится курсор по умолчанию, и помещение его первым в списке сохраняет X от использования других курсоров, которые устанавливают некоторые пакеты шрифтов (artwiz)

Опция: **:unscaled** говорит X использовать **unscaled** версии этих шрифтов всякий раз, когда возможно предотвратить pixelation. Эта опция может применяться для любого bitmap шрифта

**/usr/share/fonts/75dpi:unscaled**  
**/usr/share/fonts/100dpi:unscaled**

Это основные bitmap шрифты. Мы обычно хотим загружать шрифты для 75dpi перед 100dpi, так как большинство тем ожидают этого. Тем не менее, большинство современных мониторов имеют более высокое значение dpi, так что если Вы обнаружите у себя косоглазие читая меню, Вы скорее всего захотите поэкспериментировать с переключением порядка в списке шрифтов

**/usr/share/fonts/Type1**  
**/usr/share/fonts/TTF**

Type1 и TrueType шрифты. Порядок не имеет большого значения, но заметьте что оба пакета содержат Luxi Sans/Serif/Mono шрифты.

```
/usr/share/fonts/corefonts
/usr/share/fonts/freefonts
/usr/share/fonts/sharefonts
```

это Microsoft's TrueType core fonts, и набор TrueType шрифтов и коллекция свободных Type1 Fonts, соответственно.

```
/usr/share/fonts/artwiz
/usr/share/fonts/terminus
/usr/share/fonts/ttf-bitstream-vera
/usr/share/fonts/unifont
```

Если Вы установили пакеты упомянутые выше, то Вам следовало бы добавить эти пути в список шрифтов.

```
/usr/share/fonts/local
/usr/share/fonts/cyrillic
```

эти шрифты обычно не нужны и вероятно пустые или не существуют. Для гарантии, проверьте файл `font.dir` внутри каждой директории. Если этот файл пуст, тогда этот каталог нет необходимости добавлять в список путей к шрифтам.

```
/usr/share/fonts/*everythingelse*
```

все директории новых шрифтов, которые Вы установили. Большинство из них уникальны. Если Вы обеспокоены о каком-нибудь отдельном шрифте, поместите его наверх списка. Вы можете проверить список из доступных шрифтов, посмотрев файл **fonts.dir**, находящийся в директории шрифта.

Теперь сделаем несколько финальных изменений в плане порядка следования путей в списке. Возьмём основные bitmap шрифты и поместим их последними. Это делается потому, что у нас есть намного лучшие TTF или Type1 версии некоторых из этих шрифтов которые мы чаще используем. Также пакет **media-fonts/corefonts** содержит большое количество core fonts и его следовало бы поместить повыше в списке.

В результате список должен выглядеть подобно:

### Файл: `/etc/X11/xorg.conf` (Список путей к шрифтам)

```
...
Section "Files"
    FontPath "/usr/share/fonts/misc:unscaled"
    FontPath "/usr/share/fonts/Type1/"
    FontPath "/usr/share/fonts/TTF/"
    FontPath "/usr/share/fonts/corefonts"
    FontPath "/usr/share/fonts/freefonts"
    FontPath "/usr/share/fonts/sharefonts"
    FontPath "/usr/share/fonts/terminus"
    FontPath "/usr/share/fonts/ttf-bitstream-vera"
    FontPath "/usr/share/fonts/unifont"
    FontPath "/usr/share/fonts/75dpi:unscaled"
    FontPath "/usr/share/fonts/100dpi:unscaled"
    FontPath "/usr/share/fonts/artwiz"
    FontPath "/usr/share/fonts/cyrillic"
EndSection
...
```

Также проверьте в том же файле, что загружается модуль freetype

Файл: /etc/X11/xorg.conf

Load "freetype"

## Перезагрузка служб

После изменения конфигурационных файлов, Вам необходимо перезагрузить Xorg. Если Вы используете X display manager, такой как GDM, KDM, XDM и т.д. тогда просто выйдите из Вашей сессии и перезагрузите X, нажав "**Ctrl+Alt+Backspace**", находясь на экране входа в систему. Если это не работает, то рестарт Xorg возможен с помощью перезагрузки службы **xdm**.

Кто-то могут заметить подвисание системы, запуская следующие команды в Xorg, так что я советую запускать их в другом терминале, нажав "CTRL + ALT + F1" и войти в систему. (Нажатие ALT + F7 вернет Вас обратно в X из консоли.)

**# /etc/init.d/xdm restart**

После выполнения этой команды, Ваш X display manager будет перезагружен. Вновь войдя в систему, Вы должны получить шрифты со сглаживанием (anti-aliased).

## Mozilla Firefox and Thunderbird

### **Mozilla Firefox 1.0+**

Для активизации шрифтов со сглаживанием в Firefox 1.0. убедитесь, что Вы собрали Firefox с выключенным USE флагом: **moznoxt**. Запустите утилиту **ufed** для этого (не забудьте установить её: **emerge ufed**).

Итак:

**#emerge mozilla-firefox**

Необходимо хотя бы один раз запустить Firefox, чтобы он создал профиль в :  
~/.mozilla/firefox/default.xxx/.

Направляйтесь в **about:config** в Вашем браузере, или альтернативно, измените файл **user.js** в ~/.mozilla/firefox/default.xxx/, где xxx являются буквами или цифрами.

Firefox 1.0.4 сохраняет файл: **prefs.js** в ~/.mozilla/firefox/xxx.default/, где xxx это строка из цифр и букв.

Либо через диалог **about:config**, либо в файле user.js Вы должны вставить следующие строки, как показано ниже:

**Внимание:** Установка "font.FreeType2.printing" в true, может вызвать внезапное падение Firefox при попытке печати через.

**Файл:** ~/.mozilla/firefox/\_\_\_\_.default/prefs.js

```
// TrueType
user_pref ("font.FreeType2.enable", true);
user_pref ("font.freetype2.shared-library", "libfreetype.so.6");
// if libfreetype was built without hinting compiled in
// it is best to leave hinting off
user_pref ("font.FreeType2.autohinted", true);
user_pref ("font.FreeType2.unhinted", true);
// below a certain pixel size anti-aliased fonts produce poor results
user_pref ("font.antialias.min", 0);
user_pref ("font.embedded_bitmaps.max", 1000000);
// sample user_prefs for TrueType font dirs
user_pref ("font.directory.truetype.1", "/usr/share/fonts/ttf-bitstream-vera");
user_pref ("font.directory.truetype.2", "/usr/share/fonts/TTF");
user_pref ("font.directory.truetype.3", "/usr/share/fonts/corefonts");
user_pref ("font.directory.truetype.4", "/usr/share/fonts/freefont");
user_pref ("font.FreeType2.printing", true);
```

Если Вы следуете этому HOWTO, тогда Вы не будете иметь проблем с перечисленными каталогами, иначе Вы можете изменить их.

В Firefox 1.0, следуйте в Edit » Preferences » General » Fonts & Colors.

В Firefox 1.5, следуйте в Edit » Preferences » Content » Fonts & Colors » Advanced.

Затем, установите эти параметры:

- \* **Proportional: Serif (Size: 16)**
- \* **Serif: Bitstream vera serif**
- \* **Sans-serif: Bitstream vera sans**
- \* **Monospace: Bitstream vera sans mono (Size: 12)**
- \* **Display Resolution: System setting**

Также проверьте, что опция: "Всегда использовать мои: Шрифты" не выбрана.

Затем перезагрузите Firefox и направляйтесь на [www.gentoo.org](http://www.gentoo.org) со сглаженными шрифтами!

Если Вы, используя KDE и Firefox, не получили сглаженных шрифтов после этого, тогда идите в Центр управления KDE / Внешний вид и темы / Шрифты, выключите сглаживание, нажмите применить, затем включите сглаживание и вновь нажмите применить.

### ***Mozilla Thunderbird 0.7, 1.0.2 и выше***

Существует очень мало опций в Thunderbird, которые связаны со шрифтами, таким образом все опции должны быть применены в user.js. Создайте этот файл в ~/.thunderbird/default/xxxxxxxx.slt/ (или, в новых версиях, ~/.thunderbird/xxxxxxxx.default/). В файле user.js Вы можете вставить те же строки, которые вставляли для Mozilla Firefox: заметьте только, что Вы вставляли в файл prefs.js ?

**Файл:** ~/.thunderbird/default/xxxxxxxx.slt/user.js

```
// TrueType
user_pref ("font.FreeType2.enable", true);
```

```

user_pref ("font.family.shared-library", "libfreetype.so.6");
// if libfreetype was built without hinting compiled in
// it is best to leave hinting off
user_pref ("font.FreeType2.autohinted", true);
user_pref ("font.FreeType2.unhinted", true);
// below a certain pixel size anti-aliased fonts produce poor results
user_pref ("font.antialias.min", 0);
user_pref ("font.embedded_bitmaps.max", 1000000);
user_pref ("font.scale.tt_bitmap.dark_text.min", 0);
user_pref ("font.scale.tt_bitmap.dark_text.gain", "0.0");
// sample user_prefs for TrueType font dirs
user_pref ("font.directory.truetype.1", "/usr/share/fonts/ttf-bitstream-vera");
user_pref ("font.directory.truetype.2", "/usr/share/fonts/TTF");
user_pref ("font.directory.truetype.3", "/usr/share/fonts/corefonts");
user_pref ("font.directory.truetype.4", "/usr/share/fonts/freetype");
user_pref ("font.FreeType2.printing", true);

```

Затем, когда откроете Thunderbird 1.0.2, следуйте в : Edit » Preferences » Fonts Or (on 0.7) Tools » Options » Fonts и установите эти параметры:

**Proportional: Serif (Size: 16)**

**Serif: Bits-bitstream vera serif**

**Sans-serif: Bits-bitstream vera sans**

**Monospace: Bits-bitstream vera sans mono (Size: 12)**

**Minimum font size: 9**

Также проверьте, что выбрана опция: "Позволять сообщениям использовать другие шрифты".

Затем перезагрузите Thunderbird и читайте Вашу почту с новыми шрифтами!

## Проблемы с размерами шрифтов

Если в Ваших приложениях из KDE шрифты имеют неправильный размер, и кроме этого, KDE не было установлено (т.е. Центр управления KDE недоступен :( ), тогда Вам следует изменить Ваш файл ~/.kde/share/config/kdeglobals следующим образом:

**Файл:** ~/.kde/share/config/kdeglobals

```

[General]
StandardFont=Bitstream Vera Sans,7,-1,5,50,0,0,0,0,0
activeFont=Bitstream Vera Sans,7,-1,5,75,0,0,0,0,0
fixed=Bitstream Vera Sans Mono,7,-1,5,50,0,0,0,0,0
font=Bitstream Vera Sans,7,-1,5,50,0,0,0,0,0
menuFont=Bitstream Vera Sans,7,-1,5,50,0,0,0,0,0
taskbarFont=Bitstream Vera Sans,7,-1,5,50,1,0,0,0,0
toolBarFont=Bitstream Vera Sans,7,-1,5,50,0,0,0,0,0

```

Число после имени шрифта это его размер (Я не знаю, что означают другие числа)

Так получилось, что шрифты в меню Firefox получились огромными. Чтобы решить эту проблему просто добавьте следующее в файл:

**Файл:** ~/.mozilla/firefox/<yourprofile>/chrome/userChrome.css

```
* {
  font-family: Sans-Serif !important;
  font-size: 8pt !important;
}
```

и сделайте тоже самое для Thunderbird  
(~/.thunderbird/<profile>/chrome/userChrome.css)

В моем случае, после применения всех этих изменений, шрифты для полей ввода (кнопок, текстовых полей или выпадающих меню) на некоторых страницах подобным Google или Amazon были слишком большими и использовался один из ужасных шрифтов. Чтобы изменить это, следуйте в userContent.css (он отличается от файла userChrome.css, который мы только что изменяли), и добавьте следующее:

**Файл:** ~/.mozilla/firefox/<yourprofile>/chrome/userContent.css

```
input, textarea, select, button,
input[type="button"], input[type="reset"], input[type="submit"] {
  font-size: 8pt !important;
  font-family: Verdana !important;
}
```

## Что делать, если Linux шрифт в Konsole не работает.

Пользователи KDE вероятно знакомы с Konsole — эмулятором терминала KDE. Но некоторые пользователи получают следующую ошибку когда они пытаются использовать Linux шрифты: **Font '-misc-console-medium-r-normal--16-160-72-72-c-80-iso10646-1' not found**

Вы можете решить эту проблему следующими действиями:

1. Запустите **kdesu kcmshell kcmfontinst**, когда Вы находитесь в KDE.
2. Введите пароль root.
3. Нажмите на иконку "add fonts icon" (an @-sign with a +-sign on it)
4. Направляйтесь в каталог **/usr/kde/3.2/share/fonts/** и добавьте доступные шрифты (files ending in .pcf.gz).

## Что, если Ваши шрифты стали размытыми и не читаемыми

У меня шрифты временами становились не читаемыми в Gnome 2.12

Полное выключение сглаживания (AntiAliasing) (Убирание комментариев со следующих секций в файлах: **/etc/fonts/local.conf** и **~/.fonts.conf**, как они объявлены в примерах выше) решит эту проблему.

Если это не помогло и Вы используете драйвер **media-video/nvidia-kernel-1.0.8178-r3**, попробуйте убрать комментарий на опции **render acceleration** в секции Device файла **xorg.conf**:

Option "RenderAccel"



## Ошибка при запуске startx

Если Вы получаете ошибку: "Fontconfig error" когда запускаете **startx**:

"Fontconfig warning: no <cachedir> elements found. Check configuration."

"Fontconfig warning: adding <cachedir>/var/cache/fontconfig</cachedir>"

"Fontconfig warning: adding <cachedir>~/.fontconfig</cachedir>"

Просто исправить её можно запуском:

```
emerge --noconfmem --oneshot fontconfig && dispatch-conf
```

## XEmacs

Если после всей проделанной работы, Вы обнаружили, что не имеете хорошего шрифта для XEmacs (который не поддерживает сглаживание), тогда дополнительно установите следующее:

```
* x11fonts-jmk
```

```
emerge x11fonts-jmk
```

Затем добавьте ещё одну строку FontPath в Ваш файл: xorg.conf:

```
FontPath "/usr/share/fonts/jmk"
```

Перезагрузите подсистему шрифтов (просто выйдя из X). Затем попытайтесь установить Ваш Xemacs шрифт в "**Neep Alt**". Вы вероятно хотите альтернативный вариант, вместо обычного, для отображения символа амперсанда.

## Полезные ресурсы

Это ресурсы из которых я брал информацию, когда писал этот HOWTO:

- \* [How to enable AA in Fluxbox, GKrellM, Abiword, Phoenix etc...](#)
- \* [X.Org & Fonts](#)
- \* [\[xfree\] caratteri sfocati](#)
- \* [Moz Firefox 0.9 \(all versions\) general discussion thread.](#)

## Смотрите также

- \* [Control Font DPI in X](#)
- \* [Forum Post](#)
- \* [XFree86 Font De-uglification HOWTO](#)
- \* [Optimal Use of Fonts on Linux](#)
- \* [Mini HOWTO: Getting Perfect Fonts on Mandrake 10.1](#)

## Примечание (от JohnBat26), мой файл: ~/.fonts.conf:

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
<!-- Info at http://gentoo-wiki.com/HOWTO_Xorg_and_Fonts --><!-- Replace Courier with a better-looking font
-->
  <match target="pattern" name="family" >
    <test name="family" qual="any" >
      <string>Verdana</string>
    </test>
    <edit mode="assign" name="family" >
<!-- Other choices - Courier New, Luxi Mono    <string>Bitstream Vera Sans Mono</string> --><!-- Other
choices - Courier New, Luxi Mono -->
      <string>Verdana</string>
    </edit>
  </match>
  <match target="font" >
    <edit mode="assign" name="rgba" >
      <const>rgb</const>
    </edit>
    <edit mode="assign" name="autohint" >
      <bool>true</bool>
    </edit>
    <edit mode="assign" name="antialias" >
      <bool>true</bool>
    </edit>
    <edit mode="assign" name="hinting" >
      <bool>true</bool>
    </edit>
    <edit mode="assign" name="hintstyle" >
      <const>hintmedium</const>
    </edit>
  </match>
  <!-- Disable autohint for bold fonts, otherwise they look *too* bold --> <match target="font" >
    <test compare="more" name="weight" >
      <const>medium</const>
    </test>
    <edit mode="assign" name="autohint" >
      <bool>false</bool>
    </edit>
  </match>
  <!-- Reject bitmap fonts in favour of Truetype, Postscript, etc. --> <selectfont>
    <rejectfont>
      <pattern>
        <patelt name="scalable" >
          <bool>false</bool>
        </patelt>
      </pattern>
    </rejectfont>
  </selectfont>
  <match target="font" >
    <edit mode="assign" name="rgba" >
      <const>rgb</const>
    </edit>
  </match>
  <match target="font" >
    <edit mode="assign" name="hinting" >
      <bool>true</bool>
    </edit>
```

```
</match>
<match target="font" >
  <edit mode="assign" name="hintstyle" >
    <const>hintmedium</const>
  </edit>
</match>
<match target="font" >
  <edit mode="assign" name="antialias" >
    <bool>true</bool>
  </edit>
</match>
</fontconfig>
```

# Руководство по видеокартам nVidia в Gentoo Linux

Ссылка на оригинал:

<http://www.gentoo.org/doc/ru/nvidia-guide.xml>

С версии: 1.0

## 1. Введение

Компания **nVidia** выпускает свои собственные драйверы для Linux, отличающиеся хорошей производительностью и поддержкой трехмерного ускорения. В дереве Portage присутствуют два драйвера — **nvidia-drivers** для новых видеокарт nVidia и **nvidia-legacy-drivers** для старых.

**Примечание:** Раньше в Gentoo было два отдельных пакета — для модуля ядра nVidia (**nvidia-kernel**) и для библиотек GLX X11 (**nvidia-glx**). Эти пакеты были удалены из дерева Portage для замены на **nvidia-drivers** и **nvidia-legacy-drivers**. Если вы до сих пор используете **nvidia-kernel** и **nvidia-glx**, то вы должны перейти на использование новых пакетов.

## 2. Настройка карты

### Конфигурирование ядра

Как уже было сказано, драйвер **nVidia** настраивается и запускается для текущего ядра. Он собирается в качестве модуля, поэтому ядро должно поддерживать загрузку модулей. Если вы использовали **genkernel** для конфигурации ядра, тогда все уже сделано. Если же нет, тогда дважды проверьте конфигурацию вашего ядра на наличие поддержки загружаемых модулей:

**Листинг 2.1:** Включение загрузки модулей ядра  
Loadable module support --->  
[\*] Enable loadable module support

Также вам потребуется включить в ядре **Memory Type Range Register**:

**Листинг 2.2:** Включение MTRR  
Processor and Features --->  
[\*] MTRR (Memory Type Range Register) support

Также, если у вас AGP-видеокарта, вы можете включить поддержку **agpgart** в ядре

или в виде модуля. Если вы не используете встроенный **agpgart**, то драйверы будут использовать собственную реализацию **agpgart** под названием **NvAGP**. На одних системах эта реализация работает лучше, чем встроенная, на других — хуже. Вам понадобится проверить это на своей собственной системе, чтобы сделать выбор в пользу лучшей производительности. Если вы не хотите экспериментировать, просто используйте встроенный в ядро **agpgart**:

**Листинг 2.3:** Включение **agpgart**

```
Device Drivers --->
Character devices --->
<*> /dev/agpgart (AGP Support)
```

### Замечания к архитектурам

**Важно:** Что касается процессоров **x86** и **AMD64**, встроенный в ядро драйвер конфликтует с двоичным драйвером, выпускаемым **nVidia**. Если вы собираете ядро для этих процессоров, следует полностью убрать поддержку встроенного драйвера, как показано ниже:

**Листинг 2.4:** Удаление встроенного драйвера

```
Device Drivers --->
Graphics Support --->
< > nVidia Framebuffer Support
< > nVidia Riva support
```

Хорошей альтернативой кадрового буфера будет VESA:

**Листинг 2.5:** Включение поддержки VESA

```
Device Drivers --->
Graphics Support --->
<*> VESA VGA graphics support
```

Затем в разделе «VESA driver type» выберите либо **vesafb**, либо **vesafb-tng**. Если у вас процессор AMD64, вы должны выбрать **vesafb**, а не **vesafb-tng**:

**Листинг 2.6:** Выбор типа кадрового буфера

```
(X) vesafb
( ) vesafb-tng
```

Если вы используете **vesafb**, то дополнительную информацию можно посмотреть в файле **/usr/src/linux/Documentation/fb/vesafb.txt**, или же найдите нужную документацию по кадровому буферу в каталоге **/usr/src/linux/Documentation/fb/**.

### Последующая настройка ядра

The **nvidia-drivers** and **nvidia-legacy-drivers** **ebuilds** автоматически определяют версию ядра на основании символической ссылки **/usr/src/linux**. Пожалуйста проверьте правильность конфигурации ядра и что эта ссылка правильно указывает

на исходные коды. Обратитесь к разделу «Конфигурирование ядра» из Настольной книги за подробностями конфигурирования ядра.

Если вы используете gentoo-sources-2.6.11-r6, каталог /usr/src может выглядеть примерно так:

**Листинг 2.7:** Проверка символической ссылки /usr/src/linux

```
# cd /usr/src
```

```
# ls -l
```

(Проверьте, что ссылка linux указывает на нужный каталог)

```
lrwxrwxrwx  1 root root  22 Apr 23 18:33 linux -> linux-2.6.11-gentoo-r6
```

```
drwxr-xr-x  4 root root 120 Apr  8 18:56 linux-2.4.26-gentoo-r4
```

```
drwxr-xr-x 18 root root 664 Dec 31 16:09 linux-2.6.10
```

```
drwxr-xr-x 18 root root 632 Mar  3 12:27 linux-2.6.11
```

```
drwxr-xr-x 19 root root 4096 Mar 16 22:00 linux-2.6.11-gentoo-r6
```

В приведенном выводе можно увидеть, что символическая ссылка linux указывает на ядро linux-2.6.11-gentoo-r6.

Если ссылка не указывает на правильные исходники, вы должны обновить ее:

**Листинг 2.8:** Создание или обновление символической ссылки /usr/src/linux

```
# cd /usr/src
```

```
# ln -snf linux-2.6.11-gentoo-r6 linux
```

Необязательный шаг: уточнение поддержки устаревших видеокарт

**Примечание:** К несчастью, некоторые устаревшие видеокарты не поддерживаются более новыми версиями пакетов nvidia-drivers. У nVidia есть список поддерживаемых видеокарт. Посмотрите этот список, прежде чем будете устанавливать драйверы.

В следующем списке указаны неподдерживаемые ныне видеокарты:

**Листинг 2.9:** Неподдерживаемые видеокарты

TNT2

TNT2 Pro

TNT2 Ultra

TNT2 Model 64 (M64)

TNT2 Model 64 (M64) Pro

Vanta

Vanta LT

GeForce 256

GeForce DDR

GeForce2 GTS

GeForce2 Pro

GeForce2 Ti

GeForce2 Ultra

GeForce2 MX Integrated graphics

Quadro

Quadro2 Pro

Quadro2 EX

Если ваша карта присутствует в списке устаревших карт, то для поддержки 3D вам потребуется установить **nvidia-legacy-drivers**.

## Установка подходящих драйверов

Теперь пора установить драйверы.

**Листинг 2.10:** Установка драйверов nVidia  
(если вашей карты нет в списке устаревших карт)  
**# emerge nvidia-drivers**  
(если ваша карта присутствует в списке устаревших карт)  
**# emerge nvidia-legacy-drivers**

**Важно:** Каждый раз, когда вы собираете новое ядро или пересобираете текущее, вам понадобится запускать `emerge nvidia-drivers` или `emerge nvidia-legacy-drivers` для переустановки модулей ядра nVidia.

После завершения установки запустите `modprobe nvidia` для загрузки модуля в память.

**Листинг 2.11:** Загрузка модуля  
`# lsmod | grep nvidia && rmmod nvidia`  
`# modprobe nvidia`

Чтобы не приходилось вручную загружать модуль после каждой загрузки, вероятно, вам потребуется, чтобы это делалось автоматически при загрузке, поэтому откройте `/etc/modules.autoload.d/kernel-2.6` (или `kernel-2.4` в зависимости от используемой версии ядра) и добавьте строку `nvidia` в файл. После этого не забудьте запустить **modules-update**.

**Важно:** Если вы собрали `agpgart` в качестве модуля, то вам понадобится указать его в `/etc/modules.autoload.d/kernel-2.6` (или `kernel-2.4`, в зависимости от версии ядра).

**Листинг 2.12:** Запуск `modules-update`  
**# modules-update**

## Настройка X-сервера

Как только необходимые драйверы установлены, вам потребуется настроить X-сервер, чтобы задействовать драйвер `nvidia` вместо используемого по умолчанию `nv`.

Откройте файл `/etc/X11/xorg.conf` своим любимым текстовым редактором (как `nano` или `vim`) и найдите раздел `Device`. В этом разделе измените строку `Driver`:

**Листинг 2.13:** Замена `nv` на `nvidia` в файле настроек X-сервера  
Section "Device"  
Identifier "nVidia Inc. GeForce2"

```
Driver "nvidia"  
VideoRam 65536  
EndSection
```

Теперь перейдите в раздел Module и проверьте, что будет загружаться модуль glx, а dri — не будет:

**Листинг 2.14:** Изменение раздела Module

```
Section "Module"  
(...)  
# Load "dri"  
Load "glx"  
(...)  
EndSection
```

Далее, в разделе Screen проверьте, что каждая из инструкций DefaultDepth установлена в 16 или 24, или что у вас есть только подразделы Display с установленной Depth в 16 или 24. Без этого расширение nVidia GLX не будет запускаться.

**Листинг 2.15:** Обновление раздела Screen

```
Section "Screen"  
(...)  
DefaultDepth 16  
Subsection "Display"  
(...)  
EndSection
```

Запустите **eselect**, чтобы X-сервер использовал GLX-библиотеки nVidia:

**Листинг 2.16:** Запуск eselect  
**# eselect opengl set nvidia**

## Добавление пользователей в группу video

Добавьте ваших пользователей в группу video, чтобы у них был доступ к файлам устройств nvidia:

**Листинг 2.17:** Добавление пользователя в группу video  
**# gpasswd -a youruser video**

Если вы не используете файловую систему **udev**, то, в целом, нет необходимости делать шаги, описываемые в следующем разделе, но они ничего не испортят и сделают систему стойкой к переменам.

## Тестирование карты



Для тестирования карты nVidia запустите X и наберите `glxinfo | grep direct`. Вы должны увидеть, что прямая прорисовка (direct rendering) включена:

**Листинг 2.18:** Проверка статуса прямой прорисовки

```
$ glxinfo | grep direct
direct rendering: Yes
```

Чтобы узнать скорость прорисовки FPS (кадров в секунду), используйте программу **glxgears**.

## Включение поддержки nvidia

Некоторые пакеты, например, `mplayer` и `xine-lib`, используют локальный USE-флаг «nvidia», который разрешает поддержку аппаратного ускорения потока Mpeg (XvMCNVIDIA), полезного при просмотре широкоформатных фильмов. Добавьте «nvidia» к переменной USE в файле `/etc/make.conf` или добавьте его в качестве USE-флага для `media-video/mplayer` и/или `media-libs/xine-lib` в `/etc/portage/package.use`.

Затем запустите `emerge -uD --newuse world`, чтобы пересобрать приложения, которых коснулось изменение USE-флагов.

## Использование инструмента nVidia Settings

Начиная с версии 1.0.6106, компания nVidia предоставляет программу настройки. Эта программа может изменять настройки графики без перезапуска X-сервера и находится в дереве Portage под названием `media-video/nvidia-settings`.

## 3. Проблемы

### Работа 2D на компьютерах с ОЗУ 4 ГБ и более

Если вы сталкиваетесь с проблемами при 2D-ускорении nVidia, скорее всего, вы не можете установить диапазон совмещения записи (write-combining range) для MTRR. Чтобы проверить это, просмотрите содержимое `/proc/mtrr`:

**Листинг 3.1:** Включена ли поддержка write-combining

```
# cat /proc/mtrr
```

Каждая строка должна содержать «write-back» или «write-combining». Если вы видите строку, содержащую «uncachable», то вам понадобится изменить настройки BIOS, чтобы исправить это.

Перезагрузитесь и войдите в BIOS, затем найдите настройки MTRR (возможно в «CPU Settings»). Измените установки с «continuous» на «discrete» и загрузитесь назад в Linux. Теперь вы не нигде не должны увидеть значения «uncachable», и 2D-ускорение должно работать без какого-либо замедления.

## Пытаясь загрузить модуль ядра, я в ответ получаю «no such device»

Обычно это происходит, когда у вас нет подходящей видеокарты. Убедитесь, что ваша видеокарта основана на чипе **nVidia** (это можно проверить командой `lspci`).

Если вы уверены, что у вас карта **nVidia**, проверьте BIOS и проверьте, включен ли параметр Assign IRQ to VGA.

## 4. Настройка для опытных

### Документация

Пакет драйверов nVidia поставляется с исчерпывающей документацией. Она устанавливается в `/usr/share/doc`, ее можно посмотреть следующей командой:

**Листинг 4.1:** Просмотр документации NVIDIA  
(для nvidia-drivers)  
**\$ less /usr/share/doc/nvidia-drivers-\*/README.gz**  
(для nvidia-legacy-drivers)  
**\$ less /usr/share/doc/nvidia-legacy-drivers-\*/README.gz**

### Параметры модуля ядра

Модуль ядра nvidia способен принимать некоторые параметры, с помощью которых вы можете изменять поведение драйвера. Большинство из них описано в документации. Чтобы добавить или изменить эти параметры, отредактируйте файл `/etc/modules.d/nvidia`. Не забудьте запустить `modules-update` после изменения этого файла и держите в голове, что вам понадобится перезагрузить модуль nvidia, чтобы новые настройки подействовали.

**Листинг 4.2:** Изменение параметров nvidia  
(Отредактируйте `/etc/modules.d/nvidia` вашим любимым редактором)  
**# nano -w /etc/modules.d/nvidia**  
(Обновление информации о модулях)  
**# modules-update**  
(Выгрузка модуля nvidia...)  
**# modprobe -r nvidia**  
(...и его загрузка снова)  
**# modprobe nvidia**

### Расширенная настройка X-сервера

GLX-интерфейс также имеет множество настраиваемых параметров. Они управляют настройками TV-вывода, двойных мониторов, определением частоты монитора и т.д. И снова, все доступные параметры подробно описаны в

документации.

Если вы хотите задействовать любой из этих параметров, то вам нужно перечислить их в соответствующем разделе Device конфигурационного файла X-сервера (обычно /etc/X11/xorg.conf). Например, я хочу отключить всплывающий логотип nVidia при загрузке X-сервера:

**Листинг 4.3:** Расширенная конфигурация nvidia в настройках X-сервера

```
Section "Device"
  Identifier "nVidia Inc. GeForce2"
  Driver    "nvidia"
  Option    "NoLogo" "true"
  VideoRam  65536
EndSection
```

# Распространенные вопросы об ATI в Gentoo Linux

**Ссылка на оригинал:**

<http://www.gentoo.org/doc/ru/ati-faq.xml>

**С версии: 1.0**

## 1. Поддержка устройств

### Поддерживается ли моя видеоплата ATI?

В xorg-x11 поддерживаются многие (но не все) видеоплаты ATI, по крайней мере, поддерживаются их функции ускорения двумерной графики. Поддержка трехмерного ускорения обеспечивается либо проектом DRI, входящим в состав xorg-x11, либо драйверами с закрытым исходным кодом, поставляемыми фирмой ATI.

Графический процессор	Распространенное название	Поддержка
Rage128	Rage128	xorg DRI
R100	Radeon 7xxx, Radeon 64	xorg DRI
R200, R250, R280	Radeon 8500, Radeon 9000, Radeon 9200	xorg DRI, ATI DRI
R300, R400, R500	Radeon 9500—x800	xorg 2D, ATI DRI

### У меня есть видеоплата All-In-Wonder/Vivo. Поддерживаются ли ее мультимедийные возможности?

Мультимедийные возможности этой платы поддерживаются проектом GATOS. Эти драйверы будут в скором времени включены в дерево разработки xorg.

### Я использую не архитектуру x86. Какие у меня варианты?

Поддержка X11 на платформах PPC и Alpha сходна с поддержкой X11 на x86. Однако, драйверы производства ATI с закрытым исходным кодом не поддерживаются на PPC или Alpha, поэтому вы не сможете использовать трехмерные возможности графического процессора (ГП) R300. Если у вас именно такая видеоплата, и вы хотите, чтобы она поддерживалась в X11, свяжитесь с ATI и потребуйте, чтобы они выпустили описание вашего ГП. Для AMD64 уже выпущены закрытые драйверы, поэтому теперь пользователи AMD64 могут наслаждаться теми же возможностями видеоплат, что и пользователи x86.

Важно: Для включения поддержки agpgart на некоторых материнских платах AMD64 вам потребуется отключить поддержку K8 IOMMU.

### У меня есть ноутбук. Поддерживается ли моя модель ATI Mobility?

Должна, но у вас может возникнуть проблема с настройкой из-за идентификатора OEM PCI ID, присвоенного изготовителем. В большинстве случаев, вам может потребоваться самостоятельно составить файл конфигурации или использовать

утилиту `xorgconfig`.

## 2. Установка

### Пакеты

В пакете **xorg-x11** представлена реализация X11

В ядрах 2.6.x модули DRI могут собираться вместе с ядром или устанавливаться пакетом **x11-drm**

В ядрах 2.4.x необходимо использовать пакет **x11-drm**

В пакете **ati-drivers** представлены закрытые драйверы ATI для X и модули ядра для ядер как серии 2.4, так и 2.6

Если вы хотите использовать внутреннюю поддержку `agpgart` производства ATI вместо предоставляемой в ядре Linux, то драйвер `agpgart` и драйвер материнской платы (в конфигурации вашего ядра) должны быть собраны как модули или вообще не собираться.

### Листинг 2.1: Установка драйверов

(если нужно установить с X11 только драйверы и модули Rade128)

```
# VIDEO_CARDS="r128" emerge x11-drm
```

(чтобы получить только поддержку Radeon)

(R100, R200, R250, R280, пока без R300)

```
# VIDEO_CARDS="radeon" emerge x11-drm
```

(чтобы установить драйверы фирмы ATI с закрытым исходным кодом)

(R200, R250, R280 и R300)

```
# emerge ati-drivers
```

(чтобы установить X11 без каких-либо модулей ядра)

```
# emerge xorg-x11
```

### Настройка

Для создания конфигурационного файла `xorg.conf` рекомендуется использовать `xorgcfg` или `xorgconfig`. В качестве альтернативы вы можете использовать параметр автоконфигурации Xorg:

### Листинг 2.2: Автоматическая настройка X

```
# X -configure
```

За дополнительной информацией о том, как создать базовый файл конфигурации `xorg.conf`, пожалуйста обратитесь к описанию настройки X-сервера в Gentoo.

**Примечание:** Если вы установили пакет `ati-drivers`, вы также сможете использовать `fglrxconfig`.

**Важно:** У пользователей PPC есть возможность использовать отдельный конфигуратор **Xorgautoconfig**, установив пакет `Xorgautoconfig`, но это необязательно.

### Переключение на OpenGL

Как только X установлен, настроен и заработал, он способен использовать библиотеки ATI OpenGL:

**Листинг 2.3:** Запуск `eselect`  
`# eselect opengl set ati`

### 3. Дополнительные источники

Прочтите Hardware Acceleration Guide для получения дополнительных сведений по настройке своей графической карты ATI.

Дополнительные сведения о Gentoo Linux и двоичных драйверах ATI можно прочитать в Wedge Unofficial Gentoo ATI Radeon FAQ.

# HOWTO fbsplash

**Ссылка на оригинал:**

[http://ru.gentoo-wiki.com/HOWTO\\_fbsplash](http://ru.gentoo-wiki.com/HOWTO_fbsplash)

**С версии: 1.2**

## Введение

Во первых я хочу выразить благодарность gentoo-разработчику [Spock](#) за разработку gensplash (fbsplash and splashutils) и vesafb-tng. Как альтернативу этому how-to, вы можете попробовать Spock's [Gensplash-in-5-easy-steps How-to](#).

## Что такое gensplash?

Gensplash - это проект основанный [Spock`ом](#), цель которого — создание набора инструментов, способных "обогащать" процесс загрузки Gentoo (отображение сообщений, анимации, полос состояния, и т. д.).

## Что такое fbsplash (framebuffer splash)?

**Fbsplash** - это часть gensplash проекта. Согласно документации по fbsplash, "framebuffer splash - это особенность ядра которая разрешает отображать фоновые изображения на выбранных консолях и переключать первую консоль в так называемый безмолвный режим (silent mode), в то время когда система загружается/перезагружается/выключается. Доступ до устройства fbsplash осуществляется путем обращения к /dev/fbsplash. Для работы с fbsplash, требуется устройство поддерживающее framebuffer.

## Что такое устройство frame buffer?

Устройство frame buffer обеспечивает уровень абстракции устройствам вывода графики, что обеспечивает буфер кадров некоторых видеоустройств и интерфейс доступа к ним для программного обеспечения, т. е. программе не обязательно знать о существовании низкоуровневых системных вызовов, регистров и т. д. Доступ к устройству осуществляется через определенные файлы, обычно расположенные в ветке */dev* дерева каталогов, в частности */dev/fb\**.

Существует множество различных frame buffer устройств.

1. Некоторые из них, стандартные поддерживают большинство современных видеокарт:
  - vesafb (vesa frame buffer);
  - vesafb-tng (vesa frame buffer нового поколения).
2. Другие специфичные, будут работать только с определенными моделями:
  - rivafb (может работать с nvidia видеокартами);
  - radeonfb (может работать с ati видеокартами).

Предполагается, что Fbsplash будет работать на любом из frame buffer устройств (и вообще на любой видеокарте способной на 8/15/16/24/32 битную цветовую гамму и directcolor/pseudocolor режимы).

## становка fbsplash и splashutils

В дальнейшем как пример используется vesafb-tng frame buffer устройство. Также подразумевается что вы используете Gentoo дистрибутив.

```
Code: Emerging splashutils & themes
```

```
# emerge -va splashutils splash-themes-gentoo splash-themes-livedd
```

## Установка заплат ядра

Во первых убедитесь что fbsplash и vesafb-tng заплаты установлены в вашем ядре. Начиная с версии 2.6.10 ( а может и раньше ) всё это входит в gentoo-sources. Если у вас нет этих заплат вы можете взять их с [Spock's homepage](#)

## Конфигурация ядра

Во первых нужно включить поддержку vesafb-tng frame buffer в вашем ядре (вы можете пропустить этот шаг если у вас уже настроен framebuffer драйвер отличный от vesa-tng, например i810fb):

```
Device Drivers ---> Graphics support--->:
<*> Support for frame buffer devices
<*>   VESA VGA graphics support
      VESA driver type (vesafb-tng)
(1280x1024@60) VESA default mode
```

Еще вам нужна поддержка fbsplash в вашем ядре.

```
Device Drivers > Graphics support > Console display driver support:
<*> Framebuffer Console support
<*> Support for the framebuffer splash
```

Если вы хотите загружать initramfs изображение динамически во время загрузки ( смотрите "Загрузка initramfs изображений динамически во время загрузки" раздел этого howto), вы можете включить еще одну специальную опцию. Если вы ещё не знаете что хотите делать, включите её все равно ( это не повредит :)

```
Device Drivers > Block devices:
<*> RAM disk support
(4096) Default RAM disk size (kbytes) (NEW)
[*]   Initial RAM disk (initrd) support
```

Initial RAM disk (initrd) support теперь (в ядре 2.6.22) находится в General setup.

## Компиляция ядра (Вручную)

```
# cd /usr/src/linux
```

Вы должны убедиться что не осталось файлов после предыдущей компиляции в каталоге вашего ядра:

```
# make mrproper
```

Теперь настраиваем ядро.

```
# make menuconfig
```

После того как вы настроили ваше ядро, откомпилируйте (Важно: этот шаг нужно



сделать до установки splashutils). "make" скомпилирует одновременно и образ ядра('make bzImage') и модули ('make modules').

```
# make
```

## Конфигурация / компиляция ядра (Для пользователей genkernel)

Необходимо запустить genkernel со следующими ключами:

```
# genkernel --gensplash=livedcd-2006.0 --gensplash-res=1280x1024 --menuconfig all
```

где ключи означают следующее:

--gensplash=livedcd-2006.0 - использовать тему livedcd-2006.0 (из пакета splash-themes-livedcd),

--gensplash-res=1280x1024 - при загрузке необходимо использовать разрешение 1280x1024,

--menuconfig - необходимо запустить ncurses конфигуратор ядра,

all - компилировать ядро, модули и initrd образ.

## Установка splashutils

Теперь настало время для установки splashutils. Splashutils это ПО используемое для загрузки изображений (а также перекодировки jpg, png).

```
# emerge splashutils
```

Конечно, /usr/src/linux должна ссылаться на исходники с установленными заплатками..

## Конфигурация / компиляция ядра (ручная работа)

Теперь ваше ядро и система подготовлены для того чтобы показать splash screen. Для завершения компиляции и установки вашего нового ядра осталось сделать следующее.

Если /boot расположен на отдельном разделе подмонтируйте его.

1. mount /boot

Затем инсталируйте модули и образ ядра.

1. make modules\_install
2. make install

## Создание образа initramfs

Приведенных выше инструкций (а также второй части следующего howto, [Configuring your bootloader](#)) достаточно, чтобы получить splash screen после того как ядро будет загружено. Однако если вы хотите чтобы сексапильная девка с пингвином на маечке появлялась раньше (сразу после инициализации - vesafb-tng), вам понадобится **initramfs**.

## Что такое initramfs?

(Цитата [Spock'a](#)): "Initramfs (initial ramfs) это кусок кода который распаковывает сжатый сrio образ во время загрузки ядра."

Чтобы использовать initramfs его надо сначала создать. Образ initramfs должен содержать:

1. картинки, которые будут использоваться как фон;
2. необходимы файлы конфигурации;
3. утилиту помощника (splash\_helper).

Изображения и конфигурационные файлы вместе образуют **тему gensplash**.

Существует **два** пути для использования initramfs image:

1. компиляция initramfs image непосредственно в ядро;
2. загрузка initramfs image динамически во время загрузки системы (с использованием 'initial ram disk' - initrd).

Вы можете сами выбирать вариант, который вам по душе. .

## Компиляция образа initramfs непосредственно в ядро

Сначала удалите старый образ initramfs:

```
# rm /usr/src/linux/usr/initramfs_data.cpio.gz
```

Теперь создайте новый образ initramfs, содержащий изображения, конфигурационные файлы и userspace helper. Проще всего будет сделать это при помощи **splash\_geninitramfs** (идущим в составе splashutils). Его использование похоже на утилиту 'splash' из пакета bootsplash.

```
# splash_geninitramfs -v -g /usr/src/linux/usr/initramfs_data.cpio.gz -r 1024x768 emergence
```

При компиляции образа initramfs непосредственно в ядро, вам придется пересобрать ядро (этого недостатка лишен способ с использованием initrd):

```
# cd /usr/src/linux
# touch usr/initramfs_data.cpio.gz
# make bzImage
# mount /boot
# make install
```

## Загрузка образа initramfs динамически

Этот вариант практически идентичен сборке образа initramfs вместе с ядром. Различие заключается в том, что вместо создания образа **внутри** ядра (и последующей сборке его командой 'make' и копированием в точку монтирования /boot при помощи 'make install'), вы создадите образ initramfs как **отдельный (initrd) файл** и вам не придется пересобирать ядро. Однако, придется добавить строчку в конфигурационный файл загрузчика (по которой он найдет initrd файл). Этим мы и займемся в этой части howto

Сначала при помощи splash\_geninitramfs создадим initrd файл (который содержит изображения, конфиги и userspace helper):

```
# mount /boot
# cd /etc/splash
# splash_geninitramfs -v -g /boot/fbsplash-emergence-1024x768 -r 1024x768 emergence
```

Как видно файл будет создан в точке монтирования(или каталоге) /boot.

К сожалению, по [словам Spock'a](#), "вы не сможете дописать новую initramfs

информацию к *initrd* образу". Это означает, что данной возможностью не смогут воспользоваться пользователи *genkernel* ядер, т. к. *genkernel* создает собственный образ *initrd* для процесса загрузки.

*solshark* [*solshark\_at\_rambler\_dot\_ru*]: *genkernel* 3.3.6 позволяет сделать так:

1. *genkernel --gensplash=DangerGirl initrd*

Так что теперь и для пользователей *genkernel* есть возможность воспользоваться данной фишкой.

Если вы не меняете ядро, то просто поменяйте строку описания *initrd* в *grub.conf* с:

```
root=/dev/ram0 real_root=/dev/$ROOT
initrd (hd0,0)/boot/initrd
```

на:

```
root=/dev/$ROOT
initrd (hd0,0)/boot/fbsplash-emergence-1024x768
```

## Создание собственных тем для gensplash

Стандартная тема 'emergence' и её конфигурационные файлы находятся в */etc/splash/emergence*.

Если хотите создать новую (например с названием 'newtheme'), вам следует:

1. создать для неё каталог в */etc/splash*:

```
# mkdir /etc/splash/newtheme
```

2. создать каталог 'images' в */etc/splash/newtheme*:

```
# mkdir /etc/splash/newtheme/images
```

и скопировать туда картинки (в форматах *png/jpg*) которые будут использованы в теме.

3. создать необходимые конфигурационные файлы. Они должны иметь расширение *.cfg* и находиться в каталоге темы (*/etc/splash/newtheme*). Образцы конфигурационных файлов можно посмотреть в теме ('emergence'):

```
# ls -l /etc/splash/emergence/
images/
1024x768.cfg
1280x1024.cfg
800x600.cfg
```

Очень важно убедиться в том в опциях 'jpeg=' и 'silentjpeg=' (в *.cfg* файлах) прописаны корректные ссылки на файлы с изображениями (в каталоге */etc/splash/newtheme/images*). Если вы не используете *jpeg* для создания темы, замените опции на 'pic=' и 'silentpic='.

Для более близкого знакомства с форматом тем *fbsplash*, изучите документацию пакета *splashutils*:

```
# less /usr/doc/splashutils-version/theme_format.gz
```

## Перевод тем bootsplash в темы gensplash

Так же Spock сделал удобную утилиту **bootsplash2fb splash** (этот скрипт идет в комплекте со splashutils) ,который сделает легким преобразование тем bootsplash формат gensplash.

Например если у вас есть тема 'gentoo-highquality' ,расположенная в [/etc/bootsplash](#) ,вам нужно только скомандовать(от рута):

```
# bootsplash2fb splash gentoo-highquality
```

и утилита сделает gensplash-совместимую тему в каталоге [/etc/splash](#) ([/etc/splash/gentoo-highquality](#)) со всеми необходимыми файлами.

## Использование собственных тем gensplash

Для создания нового образа из темы которую вы только что сделали(пусть ее имя 'newtheme',а разрешение 1280x1024), вам необходимо использовать splash\_geninitramfs.

Если вы решили вкомпилировать образ в ядро(первый вариант):

```
# splash_geninitramfs -v -g /usr/src/linux/usr/initramfs_data.cpio.gz -r 1280x1024 newtheme
# cd /usr/src/linux
# touch usr/initramfs_data.cpio.gz
# make bzImage
# mount /boot
# make install
```

Если вы используете initrd (второй вариант):

```
# mount /boot
# splash_geninitramfs -v -g /boot/fbsplash-newtheme-1280x1024 -r 1280x1024 newtheme
```

Вот что означают опции splash\_geninitramfs:

```
-v splash_geninitramfs выводит дополнительную информацию
-g указывает splash_geninitramfs создать образ initramfs (/usr/src/linux/usr/initramfs_data.cpio.gz или /boot/fbsplash-newtheme-1280x1024)
-r говорит splash_geninitramfs использовать разрешение 1280x1024
-'newtheme'указывает splash_geninitramfs искать .cfg файлы в каталоге /etc/splash/newtheme
```

Для просмотра описания всех опций splash\_geninitramfs' введите:

```
# splash_geninitramfs --help
```

Последний пункт в этой части how-to,полезный совет из документации splashutils':

Если вы хотите сохранить немного памяти и не планируете использовать все доступные разрешения вы можете скопировать .cfg файлы и фоновые картинки только для того разрешения, которое вы используете.

А теперь пришло время настроить ваш загрузчик.

## Настройка загрузчика

### Параметры ядра

Вот опции, которые необходимо добавить в конфигурационные файлы grub(lilo):

1. Необходимо указание устройства frame buffer. Вот как это может выглядеть например для драйвера vesafb-tng:

```
video=vesafb:ywrap,mtrr,1024x768-16@85
```

Вышеприведенный пример указывает на то, что мы хотим использовать разрешение 1024x768, 16 бит цвет и частоту обновления 85Hz. Если вы используете vesafb-tng, вам также **необходимо** удалить опцию 'vga=' из файла конфигурации вашего загрузчика (эта опция используется только 'обычными' vesafb frame buffer устройствами):

```
# НЕ СТАВЬТЕ опцию 'vga=' для vesafb-tng (если она стоит, то удалите ее):  
vga=791  
vga=0x517
```

2. fb splash'y необходимы свои параметры:

```
splash=verbose,theme:emergence
```

Сначала fb splash выбирает **режим** (в примере используется режим 'verbose' ). Вот цитата из документации по fb splash *"fb splash может работать в двух режимах: **verbose** и **silent**. Первый режим подразумевает простую консоль с фоновым рисунком. Режим **silent** был впервые реализован в bootsplash. Когда этот режим включен, консоль переключается в графический режим. При этом, текст не печатается. Все что будет отображаться в этом режиме зависит от программ, запущенных в пользовательском пространстве. Таким образом, может быть реализован, например, индикатор прогресса загрузки.*

Вторая часть параметра fb splash позволяет указать какую **тему** нужно использовать (в примере используется стандартная тема 'emergence').

Для работы fb splash необходимо указать оба параметра: режим и тему.

3. Если вы решили использовать **initrd** (посмотрите предыдущую главу - [Загрузка образа initramfs](#)), вам необходимы дополнительные опции в конфигурации загрузчика:

```
initrd /boot/fb splash-emergence-1024x768      # Grub  
initrd = /boot/fb splash-emergence-1024x768    # Lilo
```

В приведенном примере подразумевается, что имя рамадиска fb splash-emergence-1024x768 (и он размещен в каталоге или разделе /boot).

Следующие примеры для обоих загрузчиков grub и lilo. Используется тема 'newtheme', разрешение 1280x1024, 32 бит цвета частота обновления 70Hz. Указан режим fb splash 'silent' и используется initrd файл.

### Пример конфигурации Grub

```
title Gentoo  
root (hd0,0)  
kernel (hd0,0)/kernel-2.6.8-gentoo-r1 root=/dev/hda3 video=vesafb:ywrap,mtrr,  
1280x1024-32@70 splash=silent,theme:newtheme  
initrd (hd0,0)/fb splash-newtheme-1280x1024
```

Замените (hd0,0) своим загрузочным разделом (или,если /boot не на отдельном разделе,используйте корневой раздел добавив '/boot' к путям образа ядра и initrd)

Пользователи genkernel должны использовать что-то типа этого(все в одной строке):

```
kernel (hd0,0)/kernel-2.6.8-gentoo-r1 init=/linuxrc root=/dev/hda3
video=vesafb:ywrap,mtrr,1280x1024-32@70 splash=silent,theme:newtheme
```

## Пример конфигурации Lilo

```
image=/boot/kernel-2.6.8-gentoo-r1
label = "Gentoo"
root = /dev/hda3
append="video=vesa:ywrap,mtrr,1280x1024-32@70 splash=silent,theme:newtheme"
read-only
initrd = /boot/fbsplash-newtheme-1280x1024
```

Не забудьте запустить lilo после изменения lilo.conf:

```
# mount /boot
# lilo -v
```

После ребута и вы получите графическую загрузку.

## Добавление фона на все консоли

Если вы хотите чтобы на всех консолях был сплеш (картинка из режима 'verbose'),а не только на первой,то отредактируйте [/etc/conf.d/splash](#) установив нужные опции и и добавьте splash в ваш стандартный уровень запуска:

```
# rc-update add splash default
```

Если вы используете bootsplash, удалите его из уровня запуска:

```
# rc-update del bootsplash
```

## Gensplash для других дистрибутивов

(Jakub) Я думаю этот раздел должен состоять из трех частей:

**Вступление:** Gensplash работает на других дистрибутивах.

1. Что нужно сделать по-другому (в сравнении с этим руководством) - к примеру из-за отсутствия emerge. Это касается всех дистрибутивов не подобных Gentoo.
2. Какие дополнительные действия нужно предпринять - например модификация загрузочных скриптов. Тоже для не Gentoo дистрибутивов.
3. Другая специфика не охваченная предыдущими двумя разделами.

### Различия

1. Наложение патчей на ядро - необходимо. Мы не используем gentoo-dev-sources (хотя все возможно)
2. Конфигурация/сборка ядра - из-за невозможности использования genkernel.
3. Установка splashutils - нет package.keywords, нет emerge. Ручками:). Добавить ссылки. Предложение перейти на Gentoo было бы своевременно.
4. Добавление splash screen на все терминалы(и удаление bootpsplash) - скорее всего зависит от дистрибутива. Способ для Gentoo скорее всего работать не

будет (очевидно).

## Дополнительные шаги

Модификация загрузочных скриптов потребуется для работы индикатора прогресса. Два варианта:

- с использованием bootsplash (простой)
- без использования bootsplash (более сложный)

## Тут должно быть название пункта 3

Вариант для Debian упоминался Spock'ом на [forums.gentoo.org](http://forums.gentoo.org).

## Решение проблем

### Если /usr на отдельном разделе

Если у вас splashutils версии 0.9-pre08 и новее и каталог /usr монтируется с другого раздела, gensplash должен работать "из коробки" (без напильника). Однако, если у вас более древняя версия splashutils и, по каким-либо причинам, нет возможности ее обновить следуйте нижеописанным инструкциям.

Программы и библиотеки, которые нужны splash\_util'y чтобы запустить fbset и отобразить вашу сплешку расположены в /usr/bin и /usr/lib. И если эти каталоги находятся на других разделах (например при использовании RAID или LVM2 архитектуры), они могут быть недоступны (не примонтированы) в момент, когда они нужны splash\_utils и вы не сможете увидеть свою замечательную сплешку.

Как решить? Скопируйте их в место где они **доступны** как описано в (спасибо **aslocum** и форумам Gentoo) [Gensplash-How-to with progressbar!!!!](#) треде:

```
# cp /usr/bin/fbset /sbin
# cp /usr/lib/libjpeg.so.62 /lib
# cp /usr/lib/libpng.so.3 /lib
```

поправьте /sbin/splash (например при помощи nano, набрав *nano -w /sbin/splash*) и приведите строку 31 к виду:

```
res=`/sbin/fbset | grep "^mode" | sed -r 's/mode "([0-9]+x[0-9]+).*/\1/'`
```

Где надо только прописать /sbin/ перед fbset. Сохраните результат и выйдите из редактора, и ваша сплешка должна появиться.

### Установка фона во время загрузки

Вы наверное заметили, что фоновое изображение загружается с небольшой задержкой, даже если вы используете образ initramfs. До этого момента используется разрешение 640x480 и черно-белая цветовая гамма.

Такое положение дел нормально. Единственным выходом (чтобы получить фон как можно раньше) будет использование драйвера vesafb вместо vesafb-tng. Вот [ответ Spock'a](#) по этому поводу: "Образ initramfs используется, чтобы подключить корневую ФС, и до того как инициализируется кадровый буфер, так что не волнуйтесь по этому поводу. Как вы заметили, компьютер загружается в текстовом режиме, а не в графическом. Графический режим устанавливается после инициализации подсистемы кадрового буфера. Но до этого момента

должна произойти инициализация других вещей, а это требует времени. Вот почему в течение секунды-двух вы видите сообщения ядра. Вы наверно скажете, что с vesafb у вас такого не было. И будете правы. Но vesafb исключение, а не правило (его инициализация происходит во время загрузки ядра). Все остальные кадровые драйверы (vesafb-tng, rivaafb, radeonfb) инициализируются с подсистемой fbdev".

## Ссылки

- [Домашняя страница Spock'a](#)
- [Spock's Gensplash-in-5-easy-steps](#)

Из форумов Gentoo:

- [Gensplash-How-to с прогрессбаром!!!!](#)

Оффлайновая документация:

- Документация по Fbsplash (/usr/src/linux/Documentation/fb/splash.txt).
- Документация по кадровым буферам (/usr/src/linux/Documentation/fb/framebuffer.txt).
- Документация по splashutils (/usr/share/doc/splashutils-0.9\_pre07/).



# Hardware 3D Acceleration Guide (Англ.)

Ссылка на оригинал: <http://www.gentoo.org/doc/en/dri-howto.xml>

С версии: 1.0

## 1. Introduction

What is hardware 3D acceleration and why do I want it?

With hardware 3D acceleration, three-dimensional rendering uses the graphics processor on your video card instead of taking up valuable CPU resources drawing 3D images. It's also referred to as "hardware acceleration" instead of "software acceleration" because without this 3D acceleration your CPU is forced to draw everything itself using the Mesa software rendering libraries, which takes up quite a bit of processing power. While Xorg typically supports 2D hardware acceleration, it often lacks hardware 3D acceleration. Three-dimensional hardware acceleration is valuable in situations requiring rendering of 3D objects such as games, 3D CAD and modeling.

### How do I get hardware 3D acceleration?

In many cases, both binary and open-source drivers exist. Open-source drivers are preferable since we're using Linux and open source is one of its underlying principles. Sometimes, binary drivers are the only option, like with nVidia's cards. Binary drivers include x11-drivers/nvidia-drivers and x11-drivers/nvidia-legacy-drivers for nVidia cards and media-video/ati-drivers for ATI cards. Other open-source drivers include media-video/kyro-kernel for Kyro cards and media-video/ati-gatos for ATI cards, which aim to support ATI's video capabilities more fully.

### What is DRI?

The Direct Rendering Infrastructure, also known as the DRI, is a framework for allowing direct access to graphics hardware in a safe and efficient manner. It includes changes to the X server, to several client libraries and to the kernel. The first major use for the DRI is to create fast OpenGL implementations.

### What is X11-DRM and how does it relate to regular Xorg?

X11-DRM is an enhancement to Xorg that adds 3D acceleration for cards by adding the kernel module necessary for direct rendering.

### Purpose

This guide is for people who can't get direct rendering working with just Xorg. X11-DRM works for 3dfx, gamma, i8x0, matrox, rage128, radeon, mach64 and sis300 series drivers. VIA card owners should not use x11-drm, but should instead use the DRI provided in recent kernels (>2.6.13). See the DRI homepage for more info and documentation.

### Feedback

With suggestions, questions, etc., e-mail Donnie Berkholz.

## ***2. Install Xorg and configure your kernel***

### **Install Xorg**

Please read our Xorg Configuration Guide to get Xorg up and running.

### **Configure your kernel**

Probe for your chipset and enable just that one.

**Code Listing 2.1:** Checking your AGP chipset

```
# emerge pciutils; lspci | grep AGP
```

```
# 00:01.0 PCI bridge: Intel Corp. 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (rev 03)
```

```
(Your output may not match the above due to different hardware.)
```

If your chipset is not supported by the kernel you might have some success by passing `agp=try_unsupported` as a kernel parameter. This will use Intel's generic routines for AGP support. To add this parameter, edit your bootloader configuration file!

Most, if not all, kernels should have these options. This was configured using `gentoo-sources-2.4.20-r5`.

**Code Listing 2.2:** Configuring the kernel

```
# ls -l /usr/src/linux
```

```
lrwxrwxrwx 1 root root 22 May 29 18:20 /usr/src/linux -> linux-2.4.20-gentoo-r5
```

```
(Make sure /usr/src/linux links to your current kernel.)
```

```
# cd /usr/src/linux
```

```
# make menuconfig
```

**Code Listing 2.3:** make menuconfig options

Processor type and features --->

<\*> MTRR (Memory Type Range Register) support

Character devices --->

<M> /dev/agpgart (AGP Support)

[M] Intel 440LX/BX/GX and I815/I820/I830M/I830MP/I840/I845/I850/I860 support

(Enable your chipset instead of the above.)

< > Direct Rendering Manager (XFree86 4.1.0 and higher DRI support)

Make sure the Direct Rendering Manager (DRM) is off. The X11-DRM package will provide its own.

### **Compile and install your kernel**

**Code Listing 2.4:** Compiling and installing kernel

(This example is for a 2.4 kernel)

```
# make dep && make clean bzImage modules modules_install
```

```
# mount /boot
```

```
# cp arch/i386/boot/bzImage /boot
```

If you want your kernel to be named something other than `bzImage`, be sure to copy

to /boot/yourname instead. Don't forget to set up grub.conf or lilo.conf and run /sbin/lilo if you use LILO.

### ***3. Install X11-DRM and configure direct rendering***

#### **Install X11-DRM**

**Code Listing 3.1:** Installing X11-DRM  
**# emerge x11-drm**

#### **Configure Xorg.conf**

Some chipsets require you to rebuild xorg-x11 with USE="insecure-drivers". This applies to mach64, unichrome and savage chipsets on xorg-x11-6.8.2, and to mach64 and unichrome on xorg-x11-6.8.99.x. Savage users should not try xorg-x11-6.8.99.x as support for the savage there is broken.

**Code Listing 3.2:** Rebuilding xorg-x11  
(Add the insecure-drivers USE flag if you use one of the aforementioned chipsets)  
**# echo "x11-base/xorg-x11 insecure-drivers" >> /etc/portage/package.use**  
  
**# emerge xorg-x11**

Open /etc/X11/xorg.conf with your favorite text editor and edit it to enable DRI and GLX.

**Code Listing 3.3:** xorg.conf

```
...
Section "Module"
    Load "dri"
    Load "glx"
...
EndSection
...
Section "Device"
    Driver "radeon"
...
EndSection
...
Section "dri"
    Mode 0666
EndSection
```

If you are using a different driver, replace "radeon" with yours.

#### **Changes to modules.autoload.d**

You will need to add the module name that your card uses to /etc/modules.autoload.d/kernel-2.6 to ensure that the module is loaded automatically when the system starts up.

**Code Listing 3.4:** Editing /etc/modules.autoload.d/kernel-2.6  
(Change module name as required.)  
intel-agp

**Note:** If you compiled agpgart as a module, you will also need to add it to /etc/modules.autoload.d/kernel-2.6.

## 4. Test 3D acceleration

### Reboot to the new kernel

Reboot your computer to your new kernel and login as a normal user. It's time to see if you have direct rendering and how good it is.

**Code Listing 4.1:** Testing rendering

**\$ startx**

(No need to load modules for your driver or agpgart, even if you compiled them as a module.)

(They will be loaded automatically.)

**\$ glxinfo | grep rendering**

direct rendering: Yes

(If it says "No", you don't have 3D acceleration.)

**\$ glxgears**

(Test your frames per second (FPS) at the default size. The number should be )  
(significantly higher than before installing x11-drm. Do this while the CPU is as idle as possible.)

## 5. Using the CVS sources

Warning: Don't do this if the package worked.

### Do you need the CVS?

First you have to check whether the x11-drm package works. If it doesn't and you have checked your logs to verify it's not a configuration error, you might want to consider the CVS sources. There are also daily driver snapshots available if you do not wish to build the full CVS.

### Do the CVS sources support your card?

Check the DRI supported cards list to see if the CVS supports your card. Even if it doesn't, but it supports a similar card, try it.

### Follow the CVS Instructions

The DRI project has a document about CVS compiling themselves. Please read the document and follow the instructions up to the Installing for X.org part.

### Install the CVS

Verify that the DRI kernel module(s) for your system were built:

**Code Listing 5.1:** Verification

**# cd ~/DRI-CVS/build xc/programs/Xserver/hw/xfree86/os-support/linux/drm/kernel; ls**

For the 3dfx Voodoo, you should see tdfx.o. For the Matrox G200/G400, you should see mga.o. For the ATI Rage 128, you should see r128.o. For the ATI Radeon, you should see radeon.o. For the Intel i810, you should see i810.o. If the DRI kernel module(s) failed to build, you should verify that you're using the right version of the Linux kernel. The most recent kernels are not always supported.

Install over your X.org installation. You may wish to back up xorg-x11.

**Code Listing 5.2:** Backing up Xorg

```
# quickpkg xorg-x11
```

```
(This backs up your Xorg-X11 package.)
```

```
# make install
```

Follow the "Configure Xorg" section above.

To load the appropriate DRM module in your running kernel, copy the kernel module to `/lib/modules/`uname -r`/kernel/drivers/char/drm/` then run `modules-update` and restart your X server. If you're not running the kernel you'll be using it in, instead of ``uname -r``, use that kernel's name.

Warning: Make sure you first unload any older DRI kernel modules that might be already loaded. Note that some DRM modules require that the `agpgart` module be loaded first.

## 6. *Tweak your performance*

Get the most out of direct rendering

A few options may increase performance by up to 30 percent (or more) over the default. Set them in `/etc/X11/xorg.conf`. However, you will first need to check that your motherboard and video card support these options.

First, let's see if your video card can support fast writes. We'll do this by closely inspecting the output from `lspci`. Specifically, we are looking at the "VGA compatible controller" information.

**Code Listing 6.1:** Video card check

```
# lspci -vv
```

```
01:00.0 VGA compatible controller: ATI Technologies Inc Radeon Mobility M6 LY (prog-if 00 [VGA])
```

```
...
```

```
Capabilities: [58] AGP version 2.0
```

```
    Status: RQ=48 Iso- ArqSz=0 Cal=0 SBA+ ITACoh- GART64- HTrans- 64bit- FW+ AGP3-  
    Rate=x1,x2,x4
```

While quite a lot of information is produced, we are looking for `FW` in the "Status" subsection of the AGP "Capabilities" section. If `FW+` is present in the "Status" line, it means the card supports fast writes. We can now check if the motherboard supports fast writes.

Important: If you do not see `FW+`, but instead see `FW-`, you cannot enable fast writes in `xorg.conf`. Your card does not support fast writes.

Now let's make sure the motherboard supports fast writes as well. This time, look at the "Host bridge" section of your lspci output.

**Code Listing 6.2:** Motherboard check

```
# lspci -vv
```

```
00:00.0 Host bridge: Intel Corporation 82830 830 Chipset Host Bridge (rev 02)
```

```
...
```

```
Capabilities: [a0] AGP version 2.0
```

```
Status: RQ=32 Iso- ArqSz=0 Cal=0 SBA+ ITACoh- GART64- HTrans- 64bit- FW+ AGP3- Rate=x1,x2,x4
```

Again, examine the "Status" subsection of your AGP "Capabilities" section. Look for FW. If you see FW+, your motherboard supports fast writes.

Important: Remember, both your video card information and your motherboard information must show the same FW capability. If either device shows FW- in "Status", you cannot enable fast writes in xorg.conf.

Assuming that all has gone well and both your motherboard and video card support fast writes, let's enable this option in /etc/X11/xorg.conf and get the best performance out of your hardware.

**Code Listing 6.3:** xorg.conf

```
Section "Device"
```

```
Option "AGPMode" "4"
```

```
(This increased FPS from 609 to 618.)
```

```
Option "AGPFastWrite" "True"
```

```
(This had no measurable effect, but it may increase instability of your computer.)
```

```
(You may also need to set it in your BIOS.)
```

```
Option "EnablePageFlip" "True"
```

```
(This improved FPS from 618 to 702. It also is "risky" but few people have reported problems.)
```

```
...
```

```
EndSection
```

**Warning:** Enabling AGPFastWrite on a VIA chipset will very likely cause your machine to lock up. VIA chipsets do not play nicely with fast writes, so use this setting at your own risk.

**Note:** Remember, if you want fast writes to work properly, you will have to first enable the appropriate option in your BIOS.

If you want to set even more features, check out the feature matrix on the DRI web site or the features listing on Sourceforge.

## 7. Troubleshooting

**It doesn't work. I just recompiled my kernel or switched to a new one.**

Whenever you rebuild your kernel or switch to another kernel, you'll have to rebuild the kernel module. Note that you don't need to remerge xorg-x11, but you will need to remerge x11-drm.

**It doesn't work. I don't have rendering, and I can't tell why.**

Try `insmod radeon` before you start the X server. Also, try building `agpgart` into the kernel instead of as a module.

### **When I startx, I get this error: "[drm] failed to load kernel module agpgart"**

That's because you compiled `agpgart` into the kernel instead of as a module. Ignore it unless you're having problems.

### **Direct rendering doesn't work, and in `/var/log/Xorg.0.log` I have an error about driver version too low.**

You aren't using the `x11-drm` driver. Check if you compiled DRM and the driver into the kernel; you shouldn't have.

### **I have a Radeon, and I want TV-Out.**

Check out `ati-gatos` drivers. `emerge -s gatos`.

### **It doesn't work. My card is so incredibly new and cool that it isn't supported at all.**

Try out the binary drivers. For `ati-drivers`, a listing is at [http://www.schneider-digital.de/html/download\\_ati.php](http://www.schneider-digital.de/html/download_ati.php). If those don't support it, use `fbdev`. It's slow, but it works.

### **I have a PCI card and it doesn't work. Help!**

In section "Device" enable `ForcePCIMode`.

**Code Listing 7.1:** Enabling `ForcePCIMode`  
Option "`ForcePCIMode`" "`True`"

## **8. Acknowledgments**

Christopher Webber for suggesting a troubleshooting question about changing or recompiling kernels

Steve, for suggesting consistency between the cases of `dri` and `DRI` in `XF86Config`

## **9. References**

- <http://forums.gentoo.org/viewtopic.php?t=46681>
- <http://forums.gentoo.org/viewtopic.php?t=29264>
- <http://dri.freedesktop.org/>
- [http://www.retinalburn.net/linux/dri\\_status.html](http://www.retinalburn.net/linux/dri_status.html)

# HOWTO по настройке KDE

**Ссылка на оригинал:**

<http://www.gentoo.org/doc/ru/kde-config.xml>

**С версии: 1.0**

## 1. Что же такое K Desktop Environment?

### Проект

Проект **KDE** - проект открытого программного обеспечения посвящённый разработке **KDE**, графической рабочей среде для рабочих станций Linux и Unix. Разработкой занимаются несколько сотен разработчиков со всего мира, преданных разработке бесплатного программного обеспечения. Смотрите также Что из себя представляет проект **KDE**.

### Программное обеспечение

K Desktop Environment - это простая рабочая среда построенная на хорошо продуманной прикладной структуре (application framework), которая предоставляет функциональную совместимость, drag 'n drop и многое другое. Кроме необходимых компонентов, KDE предоставляет готовые к использованию приложения для всех мыслимых задач: файловый менеджер, интернет-браузер, офис приложения, email-клиент и т.д. И всё это относится к KDE проекту.

KDE доступен более чем на 70 языках и охватывает огромное количество пользователей. Для тех, кто интересуется, есть много скриншотов. Для большей информации о KDE, читайте статью Что такое KDE? на KDE.org.

### Сообщество

Существует много сайтов KDE сообществ. На KDEnews.org вы найдёте последние новости про KDE вообще. KDEdevelopers.org специализируется на разработке KDE, а KDE-форум больше подходит для широкой массы. Больше ссылок можно найти на Странице семейства KDE.

## 2. Установка KDE

### Что вам нужно?



Прежде всего, если вы заинтересованы в установке KDE (или поддержке kde), вам придётся убедиться, что переменная USE содержит kde и qt флаги. Для тех, кто не знает, Qt - графическая библиотека элементов управления (graphical widget library), которую использует KDE.

Во-вторых, вам нужно подумать над тем, что вы хотите установить из дистрибутива KDE. Как ранее упоминалось, KDE очень богат возможностями и включает в себя много программ. Для этого вам не обязательно верить нашим словам -- сами проверьте :)

**Листинг 2.1:** Перечисляем все пакеты KDE  
`# emerge --pretend kde | less`

Если вы не заинтересованы в установке всех этих пакетов, вы можете поставить лишь некоторые из них. Очевидно, что вам понадобится kdebase, так как он содержит основные и необходимые программы. Следующая таблица показывает несколько пакетов, которые вы можете установить.

**Таблица:** Пакеты KDE.

Пакет	Описание
<b>kdeaccessibility</b>	Программы для людей с ограниченными возможностями разрабатываются Проектом KDE Accessibility
<b>kdeadmin</b>	Административные утилиты, такие как KCron (Планирование задач), KUser (Управление пользователями) и KDat (Управление резервными копиями)
<b>kdeartwork</b>	Некоторые программы, связанные с искусством, включая хранители экрана и темы. Также смотрите <a href="http://artist.kde.org">artist.kde.org</a> .
<b>kdeedu</b>	Образовательные KDE приложения для школьников от 3 до 18 лет. Также смотрите Проект KDE Edu.
<b>kdegames</b>	KDE игры. Больше информации можно найти по ссылке KDE центр игр.
<b>kdegraphics</b>	Графические приложения для KDE, плюс KSnapshot (Программа для снятия скриншотов), KPaint (Простой графический редактор), Kpdf (Для просмотра PDF), KIconEdit (Icon Editor) и KPovModeler (Создание 3D моделей).
<b>kde-i18n</b>	Этот пакет включает поддержку для многих языков (также валют, форматов чисел и дат, ...) и документации. Больше информации по ссылке Проект KDE i18n.
<b>kdemultimedia</b>	Мультимедийные приложения, включая поддержку CD, MP3, DVD, форматов, звуковых и видео приложений. Больше информации по ссылке Проект KDE Multimedia website.
<b>kdenetwork</b>	Сетевые приложения, такие как kppp (Dial-In) и lisa (networking). Заметьте, что konqueror (файловый менеджер и веб-браузер) часть пакета kdebase!
<b>kdepim</b>	Личные Информационные инструментальные средства Управления, такие как KOrganizer (Журнал), KAddressbook (Адресная книга), Kontact (Программное обеспечение для

	коллективной работы) и KMail (E-mail). Больше информации по ссылке Проект KDE PIM.
<b>kdesdk</b>	Основные инструменты разработки, включая KBabel (Утилита для перевода), KBugBuster (Front end для отслеживания ошибок KDE) и Kompare (GUI для просмотра разницы между файлами).
<b>kdeutils</b>	Различные безделушки, которыми можно позабавится, в ожидании, пока доставят вашу пиццу. В этом пакете вы найдёте такие апплеты, как eyesapplet и fifteenapplet, а также стильные утилиты как amog, которые ничего не делают, кроме как пожирают ресурсы :)
<b>kdeutils</b>	Графические системные утилиты как kcalc (Калькулятор), kdessh (SSH терминал), kfloppy (производит действия связанные с floppy), и другие.

К примеру, чтобы поставить KDE только с сетевыми и административными приложениями:

**Листинг 2.2:** Пример установки индивидуальных компонентов KDE  

```
# emerge kdbase kdenetwork kdeadmin
```

Если вам интересно знать: компиляция KDE занимает очень много времени :)

## Первые ощущения

Теперь давайте взглянем на результаты. Твоя мама наверное говорила тебе никогда не работать под root-ом. Ну что ж, последуем маминому совету и запустим KDE под пользователем. Зайдите под вашим пользователем и настройте вашу сессию так, чтобы загружался KDE после выполнения команды startx. Вы можете это сделать добавив строчку `exec startkde` в `~/.xinitrc`:

**Листинг 2.3:** Настраиваем локальную сессию  

```
$ echo "exec startkde" > ~/.xinitrc
```

Теперь выполните команду `startx` для загрузки вашей графической среды.

**Листинг 2.4:** Запускаем KDE  

```
$ startx
```

Вас встретит приложение под названием **KPersonalizer**. Поздравляем, теперь давайте посмотрим, как можно настроить KDE...

### 3. Настройка KDE

#### KPersonalizer

**KPersonalizer** - приложение, которое настроит KDE за вас. Это очень полезный мастер, который помогает быстро изменить KDE под ваши нужды. KPersonalizer автоматически появится при первой загрузке KDE.

Первое, о чём спросит вас KPersonalizer будет страна и язык на ваш выбор. Так как мы не устанавливали никаких языковых пакетов, выбор языков будет небольшим -- очевидно в списке будет только английский. Не волнуйтесь, мы сменим язык позже (если вы конечно захотите этого).

Потом вам нужно будет выбрать System Behaviour. Эта настройка включает в себя активацию окна, выбор мыши, и другое. При выборе определённого поведения, можно увидеть его описание. Если вы не уверены что выбрать, не паникуйте -- всегда можно будет поменять настройки.

Далее KPersonalizer спросит о количестве разнообразных эффектов. Чем больше эффектов вы активируете, тем моднее будет KDE, но и процессор будет более загружен. Однако, не стоит так переживать из-за этого -- на компьютере с процессором в 600 Mhz с 128 Mb памяти разрешение всех эффектов не особо не повлияет на быстродействие системы.

Наконец, остаётся выбрать стиль. Стиль определяет декорацию окон, тему, вид кнопок, и другое. Попробуйте несколько стилей, чтобы выбрать тот, который вам понравится. Я ещё не говорил вам, что KDE полностью настраиваемый?

Откиньтесь на спинку кресла и наслаждайтесь -- запустится KDE и вас встретит милая, чистая и функциональная интерактивная рабочая среда.

#### Установка языковых пакетов

Если ваш родной язык не английский или вы хотите работать в KDE на другом языке, тогда читайте дальше. Мы установим языковые пакеты для языков, которые вам нужны.

Все языковые пакеты содержатся в одном **kde-i18n** пакете. Чтобы поставить только нужные пакеты, а не все, нужно присвоить значение переменной LINGUAS. Рекомендуется прописать переменную в /etc/make.conf.

**Листинг 3.1:** Прописываем переменную LINGUAS в /etc/make.conf

```
# nano -w /etc/make.conf
```

(К примеру, мы поставим языковые пакеты для русского (ru)

и французского (fr))

```
LINGUAS="ru fr"
```

Теперь выполните команду `emerge kde-i18n`. Когда установка закончится, запускайте KDE и KDE центр управления (Control Center) (K-menu > Settings > Control Center). Это приложение, где можно изменить намного больше настроек KDE, чем в KPersonalizer.

Чтобы изменить язык, идите в Regional & Accessibility, Country/Region & Languages. Потом добавьте язык на ваш выбор и перезагрузите KDE.

## Графический вход в систему

Если вы хотите использовать kdm как менеджер графического входа в систему (что значит, что вам не придётся каждый раз набивать на клавиатуре команду startx) сначала загляните в /etc/X11/Sessions, чтобы узнать, как называется KDE сессия:

**Листинг 3.2:** Узнаем название KDE сессии

```
# ls /etc/X11/Sessions
```

```
Xsession fluxbox kde-3.2.1
```

В примере выше, KDE сессия называется kde-3.2.1. Давайте добавим её в /etc/rc.conf в переменную XSESSION. Также измените значение переменной DISPLAYMANAGER на kdm.

**Листинг 3.3:** Изменяем значение переменной XSESSION в /etc/rc.conf

```
# nano -w /etc/rc.conf
```

(Измените две следующие переменные)

```
XSESSION="kde-3.2.1"
```

```
DISPLAYMANAGER="kdm"
```

Завершаем добавив **xdm** в **default runlevel**:

**Листинг 3.4:** Добавляем xdm в default runlevel

```
# rc-update add xdm default
```

После перезагрузки системы, как графический вход в систему будет использоваться KDM.

# Руководство по настройке GNOME

Ссылка на оригинал:

<http://www.gentoo.org/doc/ru/gnome-config.xml>

С версии: 1.2

## 1. Что такое GNOME?

### О проекте

[GNOME](#) - это свободное ПО. Проект посвящен разработке GNOME - рабочего стола и пакет разработчика для Unix/Linux систем. [GNOME Foundation](#) координирует разработку и другие аспекты проекта GNOME.

### О Программном обеспечении

**GNOME** представляет собой рабочую среду а также платформу для разработчиков. Это "открытое ПО" используется в качестве рабочей среды для рабочих станций, некоторыми промышленными лидерами. Эта среда представляет интерес, как для бизнес пользователей, домашних пользователей так и разработчиков.

### Сообщество

Как и любого большого открытого проекта, у Gnome очень большое количество разработчиков, а также большая пользовательская база. [Footnotes](#) содержит новости для пользователей GNOME Desktop; [GnomePlanet](#) предназначена для хакеров/помощников, а [Developer.Gnome.Org](#) для разработчиков.

## 2. Установка GNOME

### Что вам понадобится?

Перед тем как устанавливать **GNOME**, вы может быть захотите отредактировать переменные USE. Убедитесь что **gtk** и **gnome** прописаны как переменные USE в `/etc/make.conf`. Если вы не планируете использовать **KDE** (другая подобная рабочая среда) тогда, удалите **qt** и **kde**.

**Листинг 2.1:** Пример использования переменных USE в `/etc/make.conf`  
`USE="-qt -kde gtk gnome"`

После этого, начните установку GNOME с помощью `emerging` **gnome** и **xscreensaver**:

**Листинг 2.2:** Установка GNOME  
`# emerge gnome xscreensaver`

Это займет какое-то время, за это время вы можете прочитать все те книги которые для вас приготовила мам, и которые вы никогда в жизни не открывали. Готово? Отлично, теперь вы должны обновить переменные рабочей среды.

**Листинг 2.3:** Обновление зависимостей рабочей среды  
`# env-update && source /etc/profile`

Если вы были внимательны, то после завершения выполнения команды `emerge`, вам предлагают добавить **famd** по умолчанию, чтобы **nautilus** и **gnome-vfs** могли отслеживать изменения файлов:

**Листинг 2.4:** Добавление famd по умолчанию

```
# /etc/init.d/famd start
# rc-update add famd default
```

## Первые впечатления

Давай для начала посмотрим что у нас получилось. Выйдите из `root shell` и зайдите в систему как обычный пользователь. Сейчас мы сконфигурируем сессию GNOME с помощью команды **startx**:

**Листинг 2.5:** Использование рабочей среды GNOME по умолчанию

```
$ echo "exec gnome-session" > ~/.xinitrc
```

Запускаем графическую среду, набрав команду **startx**:

**Листинг 2.6:** Запуск GNOME

```
$ startx
```

Если все сделано правильно, GNOME приветствует вас. Поздравляем. Теперь давайте попробуем сконфигурировать рабочую среду GNOME.

## 3. Настройка GNOME

### Графический менеджеры GNOME входа в систему

Если вы хотите, чтобы GNOME Display Manager (GDM) запускался автоматически при загрузке (таким образом вы сможете зайти в систему с помощью графического интерфейса), то вы должны установить `xdm` в `default runlevel`.

**Листинг 3.1:** Установка запуска `xdm` по умолчанию

```
# rc-update add xdm default
```

Проверьте содержание `/etc/X11/Sessions`:

**Листинг 3.2:** Проверка содержания `/etc/X11/Sessions`

```
# ls /etc/X11/Sessions
Xsession Gnome
```

Как видите, теперь доступна сессия `gnome`. Теперь отредактируйте `/etc/rc.conf` и измените 2 переменные: **DISPLAYMANAGER** (должно быть `gdm`) и **XSESSION** (должно быть `Gnome`):

**Листинг 3.3:** Редактирование `/etc/rc.conf`

```
DISPLAYMANAGER="gdm"
XSESSION="Gnome"
```

После перезагрузки, GNOME Display Manager спросит имя пользователя и пароль. GNOME будет рабочей средой по умолчанию (конечно же здесь есть возможность выбора).

# Использование UTF-8 в Gentoo

**Ссылка на оригинал:**

<http://www.gentoo.org/doc/ru/utf-8.xml>

**С версии: 1.0**

## 1. Предупреждение или Напутствие переводчика

Путь Юникода в русском Gentoo-сообществе сложен. Мы постоянно сталкивались с проблемами неполной поддержки или же неподдержки вовсе отдельными приложениями стандарта UTF-8.

Но всё меняется к лучшему. Большинство приложений так или иначе стало поддерживать региональные настройки UTF-8. А для не поддерживающих уже разработаны способы исправления этого недостатка. Но всё же, если у вас возникли трудности с настройкой Юникода в Gentoo, то вы всегда найдете помощь и поддержку (иногда сквозь стиснутые зубы :) ) на следующих страницах:

- [Статья в gentoo-wiki о поддержке Юникода](#)
- [Форум русского сообщества Gentoo](#)
- [Русскоязычный форум на официальном сайте Gentoo](#)

Если вы заметили неточность или хотите дополнить данное руководство, пожалуйста свяжитесь переводчиком или одним из редакторов перевода.

## 2. Кодировки

### Что такое кодировка?

Компьютеры неспособны воспринимать текст. Вместо этого они каждый символ представляют как число. Традиционно каждый набор чисел, соответствующий алфавиту и символам (известный как кодировка или набор символов), ограничен в размере в силу ограничений оборудования.

### История кодировок символов

Самой распространенной (или наиболее принятой повсеместно) кодировкой является ASCII (Американский стандартный код для обмена информацией, American Standard Code for Information Interchange). Повсеместное использование говорит о том, что ASCII — наиболее успешный программный стандарт современности. Нынешний ASCII стандартизован в 1986 году (ANSI X3.4, RFC 20, ISO/IEC 646:1991, ECMA-6) Американским национальным институтом по стандартизации (American National Standards Institute, ANSI).

ASCII является строго семибитной кодировкой, из чего следует, что она использует семь двоичных цифр, то есть интервал от 0 до 127. ASCII начинается с 32 невидимых управляющих символов (с кодами от 0 до 31) и заканчивается управляющим символом DEL (или delete) с кодом 127. Символы в диапазоне от 32 до 126 видимы — это пробел, знаки препинания, латинские буквы и цифры.

Восьмой бит в ASCII изначально использовался как бит целостности для проверки ошибок при передаче информации. Если не указывалось особо, он всегда равнялся 0. Это означает, что в ASCII каждый символ равняется одному байту.

Хотя ASCII было достаточно для передачи информации на английском, для других европейских языков, содержащих символы с ударениями, это было не так просто. Для них были разработаны стандарты семейства ISO 8859. Они были обратно совместимы с ASCII, но также использовали восьмой бит для дополнения таблицы дополнительными 127 символами для каждой кодировки. Ограничения ISO 8859 были незначительными, поэтому стандарт получил распространение. На данный момент существует 15 вариантов стандарта ISO 8859 (от 8859-1 до 8859-15). Однако для каждого символа вне ASCII-совместимого диапазона между этими стандартами возникал конфликт. Чтобы еще больше усилить конфликт между различными кодировками, для восточноевропейских языков корпорацией Microsoft был введен набор Windows-1252, использованный в ранних версиях Windows. Этот набор является надстройкой ISO 8859-1, обладающий собственными изменениями. Однако все же эти наборы были совместимы с ASCII.

Необходимость разработки совершенно отличных от ASCII однобайтовых кодировок для нелатинских алфавитов, как например EUC (Extended Unix Coding), используемый в японском и корейском (и в меньшей степени китайском) алфавитах, породила еще большую неразбериху с кодировками, из-за чего ряд операционных систем все еще использует различные наборы символов для одного и того же языка, например японские Shift-JIS и ISO-2022-JP. А пользователи, желающие видеть кириллицу, вынуждены были выбирать между KOI8-R (для русского и болгарского языков) и KOI8-U (для украинского языка), неудачной ISO 8859-5 и популярной Windows-1251. Все три семейства кодировок не полностью совместимы с ASCII (хотя в KOI8 фонетически созвучные кириллические символы расположены так же, как и латинские, благодаря чему даже при отбрасывании восьмого бита текст оставался читабельным в ASCII-терминалах в виде транслита).

Все это привело к путанице и полной невозможности многоязычного общения, в особенности с использованием различных алфавитов. Переходим к Юникоду.

### **Что такое Юникод?**

Юникод отбрасывает лимит традиционных однобайтовых кодировок. Он использует 17 «плоскостей», содержащих по 65 536 кодов символов. Таким образом, максимальное возможное число символов равно 1 114 112. Поскольку в первой плоскости («Basic Multilingual Plane» или BMP) содержится почти всё, что вам может понадобиться, многие ошибочно посчитали, что Юникод — это 16-битный набор символов.

Юникод реализован несколькими способами, но распространены только два — UTF (Unicode Transformation Format) и UCS (Universal Character Set). Число после UTF обозначает число бит на каждый символ, тогда как число после UCS обозначает число байт. UTF-8 стала наиболее распространенной при обмене текстом Юникод из-за малых (сравнительно) потерь при хранении текста. Поэтому она и является



основной темой данного документа.

## UTF-8

UTF-8 — это кодировка переменной длины. Это означает, что для каждого символа она может использовать от одного до четырех байт. Первый байт UTF-8 используется для кодирования ASCII, благодаря чему UTF-8 полностью обратно совместима с ASCII. UTF-8 подразумевает, что ASCII и латинские символы равнозначны с небольшим увеличением размера данных, так как используется только первый байт. Пользователи восточных кодировок (например, японского), которым назначены старшие диапазоны, не так рады, так как в результате их данные разбухают на 50 %.

### Что UTF-8 может сделать для вас

UTF-8 позволяет вам работать в совместимой со всеобщими стандартами и принятой по всему миру многоязычной среде, правда с небольшими потерями при обработке данных. UTF-8 — это наилучший способ для передачи не-ASCII символов через интернет, будь то электронная почта, IRC-сети или что-нибудь еще. Несмотря на это, множество людей расценивают использование UTF-8 при передаче данных неприличным и неуважительным. Всегда следует выяснять, поддерживает ли определенный канал, группа Usenet или список рассылки UTF-8 перед тем, как использовать символы из не-ASCII диапазона UTF-8.

## 3. Включение UTF-8 в Gentoo Linux

### Поиск или создание локалей UTF-8

Теперь, когда вы поняли основы Юникода, вы готовы начать использование UTF-8 в вашей системе.

Главным требованием для UTF-8 является наличие установленной библиотеки glibc с поддержкой национального языка. При этом рекомендуется использовать файл `/etc/locale.gen`. Описание использования этого файла, однако, выходит за рамки данного документа. Его использование описано в Руководстве по локализации Gentoo.

Далее, нужно определить, доступна ли нам локаль UTF-8 для нашего языка или придётся её создать.

**Листинг 3.1:** Поиск существующей локали UTF-8

(Замените "ru\_RU" своим регионом)

```
# locale -a | grep 'ru_RU'
```

```
ru_RU
```

```
ru_RU.UTF-8
```

На выходе этой команды мы должны получить хотя бы одну строку, содержащую суффикс `.UTF-8`. Если таковых нет, то нам придётся создать локаль, совместимую с UTF-8.

**Примечание:** Запускайте следующую команду, если у вас нет UTF-8 локали для вашего языка.

**Листинг 3.2:** Создание локали UTF-8  
(Замените "ru\_RU" своим регионом)  
`# localedef -i ru_RU -f UTF-8 ru_RU.UTF-8`

Другим способом включить локаль UTF-8 является добавление её в файл `/etc/locale.gen` и генерация нужных локалей командой `locale-gen`.

**Листинг 3.3:** Одна из строк в `/etc/locale.gen`  
`ru_RU.UTF-8/UTF-8`

## Настройка локали

Есть одна переменная среды, которую необходимо установить, чтобы использовать UTF-8 локали: `LANG` (эту переменную также можно перезаписывать переменной `LC_ALL`). Есть множество способов сделать это. Некоторые люди предпочитают использовать UTF-8 только для определенного пользователя, поэтому они устанавливают эту переменную в своём `~/.profile` (если используется `/bin/sh`), `~/.bash_profile` или `~/.bashrc` (если используется `/bin/bash`).

Другие предпочитают установить локаль глобально. Есть по крайней мере один весомый аргумент в пользу этого подхода — при использовании `/etc/init.d/xdm`, так как сценарии инициализации запускают диспетчер окон до того, как будут загружены конфигурационные файлы командной оболочки.

При установке глобальной локали должен быть использован файл `/etc/env.d/02locale`. Он должен выглядеть следующим образом:

**Листинг 3.4:** Вид `/etc/env.d/02locale`  
(Как и всегда, замените "ru\_RU.UTF-8" вашей локалью)  
`LANG="ru_RU.UTF-8"`

**Примечание:** Вы можете также указывать `LC_ALL` вместо `LANG`. Так локаль устанавливается для всех категорий, включая числовые и денежные значения. На небольшом количестве систем это может вызывать некоторые проблемы. Однако, большинство пользователей может использовать `LC_ALL` без проблем. За дополнительными сведениями о категориях, затронутых использованием `LC_ALL`, пожалуйста, обращайтесь к странице локалей GNU.

Далее следует обновить среду.

**Листинг 3.5:** Обновление среды  
`# env-update`  
`>>> Regenerating /etc/ld.so.cache...`  
`* Caching service dependencies ...`  
`# source /etc/profile`

Теперь запустите `locale` без аргументов, чтобы увидеть, что у вас верные переменные среды:

**Листинг 3.6:** Проверка новой применённой локали

```
# locale
LANG=
LC_CTYPE="ru_RU.UTF-8"
LC_NUMERIC="ru_RU.UTF-8"
LC_TIME="ru_RU.UTF-8"
LC_COLLATE="ru_RU.UTF-8"
LC_MONETARY="ru_RU.UTF-8"
LC_MESSAGES="ru_RU.UTF-8"
LC_PAPER="ru_RU.UTF-8"
LC_NAME="ru_RU.UTF-8"
LC_ADDRESS="ru_RU.UTF-8"
LC_TELEPHONE="ru_RU.UTF-8"
LC_MEASUREMENT="ru_RU.UTF-8"
LC_IDENTIFICATION="ru_RU.UTF-8"
LC_ALL=ru_RU.UTF-8
```

Вот и всё. Теперь вы используете локаль UTF-8, и следующим этапом является настройка повседневно используемых вами приложений.

## 4. Поддержка приложениями

Пока Юникод делает первые шаги в мире программного обеспечения, и многобайтовые кодировки еще не включены в языки программирования, наподобие C, и множество повседневно используемых программ. Даже сейчас некоторые программы не способны работать с UTF-8 как надо. К счастью, таких не так много!

### Имена файлов, NTFS и FAT

В конфигурационном меню ядра Linux есть множество параметров NLS, но главное — не паниковать! В большинстве случаев всё, что вам нужно — включить поддержку UTF-8 NLS в ваше ядро и изменить NLS по умолчанию на utf8.

**Листинг 4.1:** Конфигурация ядра для UTF-8 NLS

```
File Systems -->
Native Language Support -->
(utf8) Default NLS Option
<*> NLS UTF8
(также <*> для других кодировок, используемых в ФС FAT или Joilet CD-ROM)
```

Если вы планируете монтировать разделы NTFS, то вам понадобится указать параметр при монтировании `nls=`. Если вы будете монтировать разделы FAT, вам понадобится указать при монтировании параметр `codepage=`. Также вы можете установить для FAT кодовую страницу по умолчанию при конфигурации ядра. Помните, что параметр `codepage`, указываемый при монтировании, будет иметь приоритет над настройками ядра.

**Листинг 4.2:** Настройки FAT при конфигурировании ядра

```
File Systems -->
DOS/FAT/NT Filesystems -->
(866) Default codepage for fat
```

Не устанавливайте Default iocharset for fat в UTF-8, так как это не рекомендуется.

Вместо этого укажите параметр `utf8=true` при монтировании раздела FAT. Для более детальной информации, смотрите `man mount` и документацию по ядру `/usr/src/linux/Documentation/filesystems/vfat.txt`.

Чтобы сменить кодировку имен файлов, используйте `app-text/convmv`.

**Листинг 4.3:** Пример использования `convmv`

```
# emerge --ask app-text/convmv
```

(Формат команды)

```
# convmv -f <current-encoding> -t utf-8 <filename>
```

(Замените `koi8-r` кодировкой, с которой хотите конвертировать)

```
# convmv -f koi8-r -t utf-8 filename
```

Для изменения содержимого файлов, используйте утилиту `iconv`, поставляемую вместе с `glibc`:

**Листинг 4.4:** Пример использования `iconv`

(Замените `koi8-r` кодировкой, с которой хотите конвертировать)

(Проверка верности вывода)

```
# iconv -f koi8-r -t utf-8 filename
```

(Конвертация файла, вы должны указать другое имя файла)

```
# iconv -f koi8-r -t utf-8 filename > newfile
```

Также для этой цели может быть использована утилита `app-text/recode`.

## Системная консоль

**Важно:** Вам необходим `>=sys-apps/baselayout-1.11.9` для поддержки Юникода в консоли.

Для включения UTF-8 в консоли вы должны отредактировать `/etc/rc.conf` и установить переменную `UNICODE="yes"`. Также прочтите комментарии в этом файле — важно, чтобы в системе были шрифты с нужным диапазоном символов (если вы хотите выжать из Юникода всё).

В файле `/etc/conf.d/keymaps` переменной `KEYMAP` должна соответствовать раскладка Unicode.

**Листинг 4.5:** Пример файла `/etc/conf.d/keymaps`

(Замените `"ru4"` на вашу любимую раскладку)

(Для русского языка нужен ключ `-u` — прим. переводчика)

```
KEYMAP="-u ru4"
```

(Подмена кодировки для русского языка — прим. переводчика)

```
DUMPKEYS_CHARSET="koi8-r"
```

**Важно:** Помните, что для правильного отображения вам понадобятся шрифты с поддержкой Юникода. Подробнее смотрите раздел X11 и шрифты этого руководства.

А теперь выберем подходящий шрифт, поддерживающий Юникод.

**Листинг 4.6:** Файл /etc/conf.d/consolefont  
(Отредактируйте переменную CONSOLEFONT)  
CONSOLEFONT="ter-k14n" # шрифт из пакета terminus-font)

## ncurses и slang

**Примечание:** Если вы не устанавливали или не используете slang, не принимайте во внимание упоминания о нём в этом разделе.

Будет хорошим решением добавить unicode к глобальным USE-флагам в файле /etc/make.conf, а затем при необходимости пересобрать sys-libs/ncurses и sys-libs/slang. Portage это сделает автоматически при обновлении системы:

**Листинг 4.7:** Обновление системы  
# emerge --update --deep --newuse world

Также понадобится пересобрать пакеты, зависящие от них, раз изменения USE вступили в силу. Используемая утилита (revdep-rebuild) входит в пакет gentoolkit.

**Листинг 4.8:** Пересборка программ, скомпонованных с ncurses или slang  
# revdep-rebuild --soname libncurses.so.5  
# revdep-rebuild --soname libslang.so.1

## KDE, GNOME и Xfce

Все основные графические оболочки полностью совместимы с Юникодом и не требуют дополнительной настройки, кроме той, что описана в этом документе. Все это благодаря тому, графические библиотеки (Qt и GTK+2) совместимы с UTF-8. Следовательно, все приложения, работающие на основе этих библиотек, также должны поддерживать UTF-8 без дополнительных настроек.

Исключением в этом правиле является Xlib и GTK+1. GTK+1 требует iso-10646-1 FontSpec в ~/.gtkrc, например -misc-fixed-\*\*-\*-iso10646-1. Также приложения, использующие Xlib или Xaw, нуждаются в подобной FontSpec, иначе они не будут работать.

**Примечание:** Если у вас установлен Центр управления gnome1, используйте его. Воспользуйтесь любым из шрифтов семейства iso10646-1.

**Листинг 4.9:** Пример ~/.gtkrc (для GTK+1), определяющий Юникод-совместимый шрифт  
style "user-font"  
{  
 fontset="-misc-fixed-\*\*-\*-iso10646-1"  
}  
widget\_class "\*" style "user-font"

**Важно:** В Xorg 6.8.2 для славянских языков эта проблема подобным образом не решается. Обратитесь к руководствам на сайтах Fantoo или Gentoo-wiki для решения данной проблемы либо откатите Xorg до версии 6.8.0.

Если приложение одинаково поддерживает Qt и GTK+2, то обычно GTK+2 GUI выдает лучшие результаты при работе с Юникодом.

## **X11 и шрифты**

Важно: По сравнению с XFree86 x11-base/xorg-x11 лучше поддерживает Юникод, и поэтому настоятельно рекомендуется использовать именно его.

Шрифты TrueType обычно совместимы с Юникодом, и большинство шрифтов, поставляемых с Xorg, имеют великолепную поддержку кодировок, хотя не все глифы Юникода могут быть отображены для конкретного шрифта. Чтобы собрать шрифты с поддержкой Восточноазиатских алфавитов (включающие набор Bitstream Vera) для X-сервера, добавьте USE-флаг `sjk`. Много приложений используют этот флаг, поэтому неплохо бы внести его как постоянный флаг.

Также несколько пакетов шрифтов в Portage совместимы с Юникодом.

**Листинг 4.10:** Необязательно: установка некоторых Юникод-совместимых шрифтов  
`# emerge terminus-font intlfonts freefonts cronyx-fonts corefonts`

## **Диспетчеры окон и терминальные эмуляторы**

Диспетчеры окон, не использующие GTK или Qt, обычно очень хорошо поддерживают Юникод, так как чаще всего для отображения шрифтов используют библиотеку Xft. Если ваш диспетчер окон не использует Xft, то вы все еще можете использовать FontSpes, указанный в предыдущем разделе в качестве шрифта Юникода.

Эмуляторы терминала, использующие Xft, в большинстве случаев поддерживают Юникод. Кроме Konsole и gnome-terminal, лучшим выбором в Portage будет `x11-terms/rxvt-unicode`, `xfce-extra/terminal`, `gnustep-apps/terminal`, `x11-terms/mlterm` или чистый `x11-terms/xterm`, собранный с USE-флагом `unicode` и запускаемый как `uxterm`. `app-misc/screen` тоже поддерживает UTF-8, если запускается с параметром `screen -U`, или внесите следующее в файл `~/.screenrc`:

**Листинг 4.11:** `~/.screenrc` с UTF-8  
`defutf8 on`

## **Vim, Emacs, Xemacs и Nano**

Vim полностью поддерживает UTF-8 и к тому же автоматически определяет файлы с UTF-8. Для более детальной информации используйте в Vim `:help mbyte.txt`.

Emacs 22.x и выше полностью поддерживает UTF-8. Xemacs 22.x ещё не поддерживает различные кодировки.

Более старшим версиям Emacs и/или Xemacs понадобятся пакеты `app-emacs/mule-ucs` и/или `app-xemacs/mule-ucs` и добавление следующих строчек в файл `~/.emacs` для поддержки CJK-языков в UTF-8:

**Листинг 4.12:** Emacs с поддержкой CJK UTF-8  
`(require 'un-define)`  
`(require 'jisx0213)`

```
(set-language-environment "Japanese")  
(set-default-coding-systems 'utf-8)  
(set-terminal-coding-system 'utf-8)
```

**Nano полностью поддерживает UTF-8 начиная с версии 1.3.6.**

## Командные оболочки

На данный момент только `bash` полностью поддерживает Юникод через библиотеку `GNU readline`. Пользователи оболочки `Z` находятся в промежуточном положении — пока ни одна из составных оболочки еще поддерживает Юникод, хотя на данный момент ведутся усиленные работы для поддержки многобайтовых кодировок.

**Оболочки `C`, `tcsh` и `ksh` не поддерживают UTF-8.**

## Irssi

`Irssi` полностью поддерживает UTF-8, хотя для этого требуется дополнительная настройка пользователем.

**Листинг 4.13:** Включение UTF-8 в `Irssi`  
`/set term_charset UTF-8`

Для каналов, где не-ASCII символы чаще всего передаются в не-UTF-8 кодировках, может пригодиться команда `/recode` для перекодировки символов. Наберите `/help recode` для большей информации.

## Mutt

Почтовый агент `Mutt` очень хорошо справляется с Юникодом. Чтобы использовать UTF-8 для `Mutt`, вам нет необходимости что-либо добавлять в конфигурационные файлы. `Mutt` будет работать с Юникодом без модификации, при условии, что все ваши конфигурационные файлы (включая подпись) сохранены в UTF-8.

**Примечание:** Вы всё ещё можете видеть знаки вопроса при чтении почты с помощью `Mutt`. Это случается из-за того, что некоторые люди используют почтовые клиенты, не устанавливающие используемую кодировку письма. Вы ничего больше не можете сделать, кроме как попросить их правильно настроить свои клиенты.

См. [Mutt Wiki](#) для большей информации.

## Man

Man-страницы являются неотъемлемой частью любой системы Linux. Чтобы удостовериться, что все страницы отображаются верно, отредактируйте `/etc/man.conf` и замените следующую строку.

**Листинг 4.14:** Изменения `man.conf` для поддержки Юникода  
(Это старая строка)  
`NROFF /usr/bin/nroff -Tascii -c -mandoc`  
(Замените ее этой)  
`NROFF /usr/bin/nroff -mandoc -c`

## elinks и links

Это самые популярные текстовые браузеры, и вы узнаете, как установить поддержку UTF-8 для них. В elinks и links есть два способа сделать это — через меню Setup (Установки) браузера или отредактировав конфигурационный файл. Чтобы настроить параметры в самом браузере, откройте какой-нибудь сайт с помощью elinks или links и нажмите Alt+S. Вы попадете в меню настроек (Setup Menu), после чего выберите параметры терминала (Terminal options), или нажмите T. Прокрутите вниз и выберите последний параметр — UTF-8 I/O, нажав на Enter. После сохранения и выйдите из меню. В links вы можете сделать то же самое, нажав Alt+S, а затем S для сохранения. Пример конфигурационного файла показан ниже.

### Листинг 4.15: Включение UTF-8 в elinks/links

(Для elinks добавьте в /etc/elinks/elinks.conf или ~/.elinks/elinks.conf следующую строку)

```
set terminal.linux.utf_8_io = 1
```

(Для links добавьте ~/.links/links.cfg следующую строку)

```
terminal "xterm" 0 1 0 us-ascii utf-8
```

## Samba

**Samba** — это набор программ, реализующих протокол SMB (Server Message Block) для UNIX-систем (Mac, Linux и FreeBSD). Этот протокол также иногда упоминается как Common Internet File System (CIFS). Samba содержит также систему NetBIOS, используемую для предоставления доступа к файлам в Windows-сетях.

### Листинг 4.16: Включение UTF-8 в Samba

(Отредактируйте /etc/samba/smb.conf и добавьте следующее в раздел [global])

```
dos charset = 866
```

```
unix charset = UTF-8
```

```
display charset = UTF-8
```

## Проверка работоспособности

Есть множество сайтов, проверяющих UTF-8. net-www/w3m, net-www/links, net-www/elinks, net-www/lynx, все браузеры, основанные на движке Mozilla (включая Firefox) поддерживают UTF-8. Konqueror и Opera тоже полностью поддерживают UTF-8.

При использовании текстовых браузеров, удостоверьтесь, что используете совместимый с Юникодом терминал.

Если вы видите символы, отображаемые как квадратики с буквами или цифрами внутри, то это значит, что в ваших шрифтах нет соответствующего глифа для символа. Вместо этого, они отображают квадрат с шестнадцатичным кодом символа UTF-8.

- [Тестовая страница W3C с UTF-8](#)
- [Тестовая страница Франкфуртского университета](#)



## Ввод символов

Dead keys могут быть использованы для ввода символов, не отображенных на клавиатуре, в X-сервере. Для этого нужно, удерживая правый Alt (в некоторых странах он называется AltGr), нажать на дополнительную неалфавитную клавишу с символом, не являющимся буквой или цифрой (модификатор), а затем нажать на необходимую букву. Dead key изменит ее. Ввод может быть изменен также с помощью Shift при нажатых AltGr и модификаторе.

Чтобы включить dead keys в X, нужно иметь раскладку, поддерживающую их. Большинство европейских раскладок уже имеют dead keys по умолчанию. Тем не менее, они не являются настоящими североамериканскими раскладками. Хотя существуют незначительные отличия между различными раскладками, простейшим решением будет использование раскладки «en\_US» вместо просто «us». Раскладка указывается в /etc/X11/xorg.conf:

**Листинг 4.17:** Пример раздела /etc/X11/xorg.conf

```
Section "InputDevice"
    Identifier "Keyboard0"
    Driver      "kbd"
    Option      "XkbLayout" "en_US" # Вместо просто "us"
    (Другие параметры Xkb)
EndSection
```

**Примечание:** Эти изменения нужны только в том случае, если вы используете североамериканскую раскладку либо любую другую, в которой dead keys не работают. Европейским пользователям нет нужды что-либо менять.

Изменения вступят в силу только при перезагрузке X-сервера. Чтобы изменения вступили немедленно, используйте утилиту setxkbmap, например, setxkbmap en\_US.

Рассмотрим действия dead keys на примерах. Хотя результат зависит от текущей локали, сам принцип действия должен работать в нее зависимости от нее. Примеры содержат символы Юникода, так что убедитесь, что ваш браузер корректно отображает их.

При нажатии единожды нажатых AltGr и [, а затем нажатой «а» мы получаем «а». При единожды нажатых AltGr и [, а затем нажатой «е», получаем «е». Нажав на AltGr и ;, мы получим «а», а нажав AltGr и ;, а затем на «е», получаем «е».

Нажав на AltGr, Shift и [, отпустив их и нажав «а», мы получим скандинавскую «а». Также при нажатии AltGr, Shift и [, отпуская только [ и повторном ее нажатии мы получим только «?». Хотя этот символ (U+02DA) и похож на символ градуса (U+00B0), на самом деле он им не является. Подобным образом работают и другие модификаторы — нажатие AltGr and [, отпуская [ и повторное нажатие выводят «?».

AltGr может быть использована вместе с одной из алфавитных клавиш. Например, комбинация AltGr и m выводит греческую строчную букву мю: «ч». AltGr и s выводит немецкий символ эсцет: «?». Множество европейцев захотят установить сочетание AltGr и 4 (или E в зависимости от клавиатурной таблицы), выводящее символ Евро, «?» (так как на их клавиатурах он нанесен).

# Руководство по русской локализации Gentoo Linux

*Ссылка на оригинал:*

<http://www.gentoo.org/doc/ru/guide-localization.xml>

*С версии: 1.0*

## 1. Введение

### Что такое русская локализация

В понятие локализации входит обеспечение вывода и ввода символов национальных языков (в случае использования набора символов, отличных от чистого латинского) и определение переменных окружения, зависящих от страны, языка и набора символов (то есть собственно установка locale). В число таких переменных, кроме страны, языка и набора символов, входят также представление даты и времени, разделителей десятичных дробей, денежных единиц и так далее.

Вследствие различия подходов в этом документе будут отдельно рассмотрены локализация текстового (то есть Linux-консоли) и графического (оконной системы X) режимов.

### О кириллических кодировках

В условиях России и некоторых других стран бывшего СССР локализация осложняется множественностью наборов символов (charsets), используемых для представления алфавитов, основанных на кириллице. Причём для Unix-систем получили распространение минимум три таких набора, в просторечии именуемых кодировками: традиционная для свободных Unix-клонов (в том числе и Linux) кодировка KOI8-R, кодировка ISO-8859-5 (именуемая кодировкой ГОСТ), используемая в проприетарных Unix-системах, и кодировка CP1251 (или кодировка Windows). Кроме того, для обеспечения экранного вывода используется кодировка CP866 (или кодировка DOS). Наконец, в последнее время всё большую популярность получает кодировка UTF-8.

В Gentoo Linux штатно (как говорится, из коробки) поддерживается кодировка KOI8-R, и всё сказанное ниже относится именно к ней. Локализацию для кодировок ISO-8859-5 или CP1251 пользователь может выполнить самостоятельно, получив из независимых источников некоторые дополнительные компоненты. О локализации для кодировки UTF-8, призванной унифицировать подход к интернационализации, будет говориться в следующих версиях этого документа, когда использование её станет общепринятым.

## 2. Базовая кириллизация консоли

### Введение

Итак, базовая локализация консоли включает обеспечение вывода национальных символов на экран и ввода их с клавиатуры. Однако в случае кириллической локализации вследствие особенностей наборов символов и аппаратных особенностей РС для ввода и для вывода используются, как правило, разные кодировки. И это требует еще двух дополнительных шагов — установки карты соответствия между наборами символов ввода и вывода, и активизация её на каждой из виртуальных консолей.

### Установка экранного шрифта

Экранный шрифт, загружаемый по умолчанию при старте системы, определяется в файле `/etc/rc.conf` строкой

**Листинг 2.1:** Строка, определяющая экранный шрифт

```
CONSOLEFONT="default8x16"
```

Для вывода русских букв "умолчальное" значение следует заменить на имя какого-либо файла шрифта с поддержкой кириллицы, например, `cp866-8x16` для представления кодировки DOS или `koi8r-8x16` — для кодировки KOI8-R. Например,

**Листинг 2.2:** Определение экранного шрифта для кодировки `cp866`

```
CONSOLEFONT="cp866-8x16"
```

Или —

**Листинг 2.3:** Определение экранного шрифта для кодировки KOI8-R

```
CONSOLEFONT="koi8r-8x16"
```

**Примечание:** Файлы шрифтов (и не только кириллических) находятся в каталоге `/usr/share/consolefonts/` и имеют вид `*.gz`, `*.psf.gz` или `*.psfu.gz`. При стандартном их размещении и форме ни полного пути, ни "расширения" имени указывать не обязательно.

**Важно:** В кириллическом Linux'е для экранного вывода, как правило, используется кодировка `cp866` (вне зависимости от того, какой набор символов принят для клавиатурного ввода). О причинах этого говорить можно было бы долго — достаточно указать, что в любом ином случае программы, использующие

псевдографику (например, Midnight Commander) приобретут весьма уродливый вид.

Экранные кириллические шрифты, поставляемые с Gentoo Linux, далеки от эстетического совершенства (справедливости ради заметим, что это относится почти ко всем Unix-клонам). И потому в качестве источника дополнительных шрифтов можно рекомендовать пакет `console-tools-cyrillic` Виктора Вагнера. В результате приведенная выше в качестве примера строка может принять вид вроде

**Листинг 2.4:** Один из лучших кириллических шрифтов для консоли

```
CONSOLEFONT="UniCyr-sans"
```

## Установка раскладки клавиатуры

Загружаемая по умолчанию раскладка клавиатуры также определяется в файле `/etc/rc.conf` строкой

**Листинг 2.5:** Строка, определяющая раскладку клавиатуры

```
KEYMAP="us"
```

По аналогии с предыдущим случаем легко догадаться, что значение `us` следует заменить на имя файла какой-либо кириллической (на этот раз уже в кодировке KOI8-R) раскладки. Место этих файлов (имеющих вид `*.map.gz`) — в каталоге `/usr/share/keymaps/i386/qwerty`, где для русской кодировки KOI8-R имеется несколько раскладок, например, `ru1-ru4`. Различия между ними — в соответствии различной маркировке клавиш (DOS или Windows — не путать с одноименными кодировками) и положении переключателя латиница/кириллица. Рискну взять на себя смелость рекомендовать раскладку `ru4` — для Windows-маркированных клавиатур, в которой переключателем служит CapsLock (при этом фиксация верхнего регистра обеспечивается одновременным нажатием клавиш Shift+CapsLock). То есть строка примет вид:

**Листинг 2.6:** Один из примеров определения кириллической раскладки

```
KEYMAP="ru4"
```

**Примечание:** Как и в случае с экранными шрифтами, при стандартном расположении файлов раскладок указывать полный путь к ним и "расширение" имени нет необходимости.

**Предупреждение:** При желании вы можете выбрать и раскладку клавиатуры для кодировки `sr1251` (например, `ru_win`). Однако в этом случае придётся отыскать или самостоятельно изготовить таблицу перекодировки (`mapscreen`) `sr1251->sr866`.

**Примечание:** Нет ничего более уродливого, чем программы, использующие псевдографику, при экранном представлении в кодировке cp1251.

## Установка карты соответствия (mapscreen)

Теперь остаётся только обеспечить соответствие между кодировками ввода (KOI8-R) и вывода (cp866), то есть загрузить соответствующую таблицу перекодировки (mapscreen). Для этого снова обращаемся к файлу /etc/rc.conf. Строка в нём, отвечающая за эту операцию, имеет по умолчанию такой вид:

**Листинг 2.7:** Строка, определяющая раскладку клавиатуры

```
#CONSOLETRANSLATION="cp437_to_iso01"
```

Необходимо, сняв с неё символ комментария (#), заменить "умолчальное" значение на необходимое нам, и посмотреть его можно в каталоге /usr/share/consoletrans — это koi2alt:

**Листинг 2.8:** Установка таблицы соответствия cp866->KOI8-R

```
CONSOLETRANSLATION="koi2alt"
```

**Предупреждение:** Пожалуйста, не забудьте снять символ комментария со строки CONSOLETRANSLATION — кто только на моей памяти не наступал на эти грабли :-)

**Примечание:** Если вы по каким-либо причинам остановились на представлении экранных шрифтов в кодировке KOI8-R, загрузка и активизация карты соответствия, разумеется, не нужна.

## Активизация карты соответствия

Результатом выполненной операции будет то, что после перезагрузки машины мы получили бы нормальный вывод кириллического текста (записанного в кодировке KOI8-R), но только — на первой (т.н. системной) консоли. Чтобы иметь то же самое на остальных виртуальных консолях (а в Gentoo Linux их по умолчанию еще 5), та же карта соответствия должна быть активизирована на каждой из них с помощью направления Escape-последовательности \033(K (символ \ в данном случае показывает, что следующие за ним символы рассматриваются как специальные). Достигнуть этого можно разными способами. Например, создать в каталоге /etc/init.d/ файл consoletrans и в любом текстовом редакторе внести в него следующие строки:

**Листинг 2.9:** Сценарий активизации mapscreen

```
for i in 1 2 3 4 5 6; do  
echo -ne "\033(K' > /dev/vc/$i  
done
```

**Примечание:** Это — для случая 6-ти консолей по умолчанию. Если количество их изменялось пользователем, значения `in`, очевидно, должны быть приведены в соответствие с реальностью.

**Примечание:** Следует отметить, что начиная с `baselayout-1.8.x` необходимость в этом отпала, поскольку этот код был встроен в `/etc/init.d/consolefont`.

Теперь командой `chmod a+x /etc/init.d/consoletrans` созданному скрипту следует присвоить бит исполнения, после чего он добавляется к уровню исполнения по умолчанию (`default runlevel`):

**Листинг 2.10:** Сценарий активизации mapscreen

```
# rc-update add consoletrans default
```

**Примечание:** Механизм автоматического обновления стартовых скриптов подробно описан в соответствующем разделе документации.

Вот теперь после перезагрузки машины мы получим полноценно локализованную консоль.

**Примечание:** Все действия, описанные в данной главе, могут быть выполнены как на заключительных шагах (*final steps*) установки Gentoo Linux, так и в произвольный момент времени в дальнейшем.

### 3. Установка русской *locale*

#### Установка `locale ru_RU.KOI8-R`

Установить локально-зависимые переменные можно различными способами, в том числе и просто в профильных файлах отдельных пользователей. Однако в Gentoo Linux предусмотрен автоматический механизм установки любых переменных окружения в качестве общесистемных — `env-update`, подробно описанный в соответствующих разделах документации.

Для использования этого механизма следует в файл `/etc/env.d/02locale` добавить строку

**Листинг 3.1:** Установка переменной LANG

```
LANG="ru_RU.KOI8-R"
```

которая определяет не только язык locale, но и все остальные локально-зависимые переменные. Далее выполняем команду `env-update`, дополняющую глобальный профильный файл `/etc/profile`, после чего остаётся только перечитать последний командой `source /etc/profile`. Теперь в ответ на команду `locale` последует вывод

**Листинг 3.2:** Вывод команды locale после установки переменной LANG

```
LANG=ru_RU.KOI8-R
LC_CTYPE="ru_RU.KOI8-R"
LC_NUMERIC="ru_RU.KOI8-R"
LC_TIME="ru_RU.KOI8-R"
LC_COLLATE="ru_RU.KOI8-R"
LC_MONETARY="ru_RU.KOI8-R"
LC_MESSAGES="ru_RU.KOI8-R"
LC_PAPER="ru_RU.KOI8-R"
LC_NAME="ru_RU.KOI8-R"
LC_ADDRESS="ru_RU.KOI8-R"
LC_TELEPHONE="ru_RU.KOI8-R"
LC_MEASUREMENT="ru_RU.KOI8-R"
LC_IDENTIFICATION="ru_RU.KOI8-R"
LC_ALL=
```

свидетельствующий, что все локально зависимые переменные приняли значения для страны России (`ru`), русского языка (`_RU`) и набора символов `KOI8-R`. Те же значения этих переменных будут устанавливаться и в сеансах всех пользователей, вошедших в систему после выполнения команды `env-update`.

**Предупреждение:** Механизм `env-update` однозначно работает в том случае, если пользовательской оболочкой (`login shell`) является `/bin/bash`. При использовании иных командных оболочек могут потребоваться некоторые дополнительные действия. Например, если в качестве `login shell` выступает оболочка `zsh`, следует отказаться от создания файла `~/.zshenv` в домашнем каталоге пользователя.

**Важно:** Переменная `LANG=ru_RU.KOI8-R` должна быть установлена в любом случае, даже если вы не любите сообщений на русском языке: в противном случае во многих программах (в том числе для `XFree86`) окажется просто невозможным, например, ввод символов кириллицы.

Как уже сказано, установка переменной LANG автоматически определяет и все прочие локально-зависимые переменные. Что в ряде случаев может быть нежелательным. Например, некоторые программы расчетного характера требуют, чтобы в качестве десятичного разделителя использовался символ точки, как в locale POSIX, а не запятой, как это принято в locale ru\_\*. Чтобы добиться этого, достаточно в конец файла /etc/env.d/02locale, после определения LANG, добавить строку вида:

**Листинг 3.3:** Установка переменной LC\_NUMERIC

```
LC_NUMERIC="POSIX"
```

Аналогичным образом могут поступить пользователи, испытывающие стойкую неприязнь к русскоязычным меню и сообщениям программ:

**Листинг 3.4:** Установка переменной LC\_MESSAGES

```
LC_MESSAGES="POSIX"
```

## 4. Установка часового пояса

### Системные часы по Гринвичу

Если системные часы машины в BIOS Setup установлены на время по Гринвичу (UTC), для настройки часового пояса достаточно создать символическую ссылку /etc/localtime на файл, описывающий данный часовой пояс. Например, для установки московского времени это будет выглядеть так:

**Листинг 4.1:** Установка московского времени

```
ln -sf /usr/share/zoneinfo/Europe/Moscow /etc/localtime
```

**Примечание:** Точное обозначение часовых поясов для России можно посмотреть в каталогах /usr/share/zoneinfo/Europe (имеет силу также для Украины и Белоруссии) и /usr/share/zoneinfo/Asia (для азиатской части России). Например, пользователям из Петропавловска-Камчатского указанная команда потребуется в следующем виде:

**Листинг 4.2:** Установка времени для Петропавловска-Камчатского

```
ln -sf /usr/share/zoneinfo/Asia/Kamchatka /etc/localtime
```



## Системные часы по местному времени

Если системные часы данной машины установлены по местному времени, кроме создания указанной ссылки потребуется внести еще и изменения в файл `/etc/rc.conf`, а именно: строку `CLOCK="UTC"` заменить на `CLOCK="local"`.

## 5. Локализация оконной системы X

### Введение

Локализация оконной системы X (конкретно — её свободной реализации XFree86) может быть выполнена двумя путями: с помощью специальных утилит конфигурирования (обычно — `xf86config`, используемой для настройки графического режима вообще) или прямым редактированием конфигурационного файла `/etc/X11/XF86Config`.

### Локализация с использованием утилиты `xf86config`

Утилита `xf86config` в большинстве случаев позволяет корректно выполнить кириллизацию системы XFree86 уже на стадии начальной её настройки. Для этого необходимо дать (в консольном режиме) одноименную команду и правильно ответить на вопросы, относящиеся к определению языково-зависимых параметров, и следующие после указания выбора типа клавиатуры.

Первый из таких вопросов — выбор страны (Enter a number to choose the country), ответ на который, на самом деле, определяет базовую раскладку клавиатуры. Очевидными ответами являются 8 Belarusian для Белоруссии и 70 Ukrainian для Украины, тогда как для России предлагается два варианта: 53 Russian и 54 Russian (cyrillic phonetic). Первый — это традиционная раскладка `qwerty`, второй же — фонетическая раскладка `uwert`, в которой символы кириллицы расположены на месте созвучных латинских. Она использовалась на старых терминалах и ныне практически не встречается. Так что на самом деле и тут выбор 53 Russian практически однозначен.

Далее следует предложение указать вариант русской раскладки (Please enter a variant name for 'ru' layout). Если, просто нажав Enter, отказаться от этого, будет установлен так называемый DOS-вариант раскладки (вернее, маркировки клавиш); маркированные таким образом клавиатуры практически вышли из употребления, поэтому следует, введя `winkeys`, выбрать Windows-вариант (не следует путать его с Windows-кодировкой, иначе говоря — `cp1251`).

Затем, введя `y`, следует положительно ответить на вопрос об указании дополнительных опций расширения XKB (Do you want to select additional XKB options

(group switcher, group indicator, etc.)?): это позволит определить переключатель латиница/кириллица (в силу исторических причин русская раскладка клавиатуры в XFree86 по умолчанию такого не имеет вообще).

В результате следующим шагом будет предложение выбрать клавишу или клавишную комбинацию для такого переключателя. Возможные варианты включают:

### **Листинг 5.1: Варианты выбора переключателя латиница/кириллица**

- 1 R-Alt switches group while pressed
- 2 Left Win-key switches group while pressed
- 3 Right Win-key switches group while pressed
- 4 Both Win-keys switch group while pressed
- 5 Right Alt key changes group
- 6 Caps Lock key changes group
- 7 Menu key changes group
- 8 Left Win-key changes group
- 9 Right Win-key changes group
- 10 Both Shift keys together change group
- 11 Control+Shift changes group
- 12 Alt+Control changes group
- 13 Alt+Shift changes group

Из них следует выбирать, руководствуясь своими предпочтениями. Однако по вполне понятным причинам можно рекомендовать, чтобы переключатель раскладок был идентичен для текстового (консольного) и графического режимов. ИМХО, лучше всего на эту роль подходит клавиша CapsLock (для этого в консоли должна быть установлена раскладка ru4). При этом первичная функция этой клавиши (фиксация верхнего регистра), как и в консоли, будет выполняться одновременным нажатием Shift+CapsLock

**Примечание:** Не менее удобным представляется и использование в качестве переключателя какой-либо из т.н. Windows-клавиш, имеющих на всех современных клавиатурах. Однако при этом для достижения идентичности переключения в консоли и в XFree86 потребуются расширенные раскладки клавиатуры из упомянутого выше пакета console-tools-cyrillic.

Следующие несколько вопросов позволяют указать переключатель для третьей клавиатурной раскладки, дополнительные позиции управляющей клавиши Control, индикацию текущей раскладки (например, индикаторами CapsLock или ScrollLock) и ещё некоторые опции. Если в них нет необходимости — ответы на эти вопросы можно просто пропустить, нажимая клавишу Enter.

Покончив с настройкой клавиатуры, следует завершить конфигурирование XFree86 (включая настройку видеорежимов) и выйти из программы `xf86config` с сохранением результатов в файле `/etc/X11/XF86Config`.

## Установка кириллических шрифтов

Программа `xf86config` обеспечивает настройку клавиатурного ввода, но не подключает по умолчанию шрифтов с символами кириллицы — это достигается редактированием файла `/etc/X11/XF86Config`. Его следует открыть в любом текстовом редакторе и отыскать в Section "Files" описание путей к шрифтовым файлам.

**Предупреждение:** При вызове файла `/etc/X11/XF86Config` необходимо отключить режим переноса строк в используемом редакторе.

Пути к файлам шрифтов описываются следующим образом:

**Листинг 5.2:** Описание путей к файлам шрифтов

```
FontPath "/usr/X11R6/lib/X11/fonts/local/"
FontPath "/usr/X11R6/lib/X11/fonts/misc/"
FontPath "/usr/X11R6/lib/X11/fonts/75dpi:unscaled"
FontPath "/usr/X11R6/lib/X11/fonts/100dpi:unscaled"
FontPath "/usr/X11R6/lib/X11/fonts/Type1/"
FontPath "/usr/X11R6/lib/X11/fonts/Speedo/"
FontPath "/usr/X11R6/lib/X11/fonts/75dpi/"
FontPath "/usr/X11R6/lib/X11/fonts/100dpi/"
```

В этот список следует внести путь к файлам шрифтов с поддержкой кириллицы. По умолчанию в любой современной версии XFree86 имеется один набор кириллических шрифтов — `Cronyx`, расположенный в каталоге `/usr/X11R6/lib/X11/fonts/cyrillic/`. И потому первая строка указанной секции должна принять вид

**Листинг 5.3:** Описание пути к файлам кириллических шрифтов

```
FontPath "/usr/X11R6/lib/X11/fonts/cyrillic/"
```

**Важно:** Строка с кириллическими шрифтами должна быть первой в списке путей, дабы именно они обнаруживались бы любыми программами в первую очередь при совпадении их имен или псевдонимов с таковыми шрифтов, не имеющих символов кириллицы.

**Примечание:** Шрифты производства Cronyx содержат ограниченное число гарнитур. Кроме того, они являются растровыми и вообще далеки от эстетического совершенства. Поэтому их по возможности лучше дополнить (или заменить) шрифтами от независимых производителей. В качестве таковых можно рекомендовать коллекцию масштабируемых (TTF и ATM) лицензионно чистых шрифтов Валентина Филиппова, входящих в российскую сборку OpenOffice и в последние версии дистрибутивов Altlinux.

## Ручная настройка клавиатуры для ввода кириллицы

К кириллизации XFree86 посредством программы `xf86config` целесообразно прибегать только при начальной настройке этой системы. Если же в ходе использования Gentoo Linux возникнет необходимость дополнительного конфигурирования (например, изменения положения переключателя раскладок), проще обратиться к прямому редактированию файла `/etc/X11/XF86Config`.

За настройку клавиатуры в нём отвечает Section "InputDevice". В ней, после указания идентификатора клавиатуры (а оконная система X в состоянии поддерживать, даже на локальной машине, более одной физической клавиатуры) и её драйвера, следует описание т.н. правил (rules), определяющих раскладку клавиатуры. Оно имеет вид вроде следующего:

### Листинг 5.4: Описание раскладки клавиатуры

```
Option "XkbRules" "xfree86"
Option "XkbModel" "pc105"
Option "XkbLayout" "ru"
Option "XkbVariant" "winkeys"
Option "XkbOptions" "grp:caps_toggle,grp_led:caps"
```

Здесь, разумеется, можно изменить всё, что угодно. Однако практически имеет смысл рассмотреть только возможные варианты значений для строки Option "XkbOptions". Значение, приведенное в примере, соответствует переключателю раскладок CapsLock (`grp:caps_toggle`) и индикатору Capslock как показателю включения русской раскладки (`grp_led:caps`). И то, и другое можно изменить, если не устраивает. Доступные значения содержатся (вместе с их краткими описаниями) в секции ! option файла `/usr/X11R6/lib/X11/xkb/rules/xfree86.lst`

# Руководство Gentoo Linux ALSA

Ссылка на оригинал:

<http://www.gentoo.org/doc/ru/alsa-guide.xml>

С версии: 1.0 (обновлено в 1.2 !)

## 1. Введение

### Что такое ALSA?

**ALSA**, или **Advanced Linux Sound Architecture** — продвинутая звуковая архитектура Linux, позволяет работать аудио и **MIDI (Musical Instrument Digital Interface** — Цифровой интерфейс музыкальных инструментов) в операционной системе Linux. ALSA является основной звуковой подсистемой в ядрах 2.6, заменяя собой OSS (Open Sound System — Открытая звуковая система), которая использовалась в ядрах 2.4.

Главные преимущества ALSA включают эффективную поддержку всех типов аудио интерфейсов, начиная от широко распространённых аудио карт и заканчивая профессиональным звуковым оборудованием, полностью модульные драйверы, поддержку многопроцессорных систем и потоковую безопасность, обратную совместимости с OSS, а также пользовательскую библиотеку `alsa-lib`, делающую разработку приложений лёгкой.

### ALSA в Gentoo

Одна из сильных сторон Gentoo заключается в предоставлении пользователю максимального контроля над тем, как система установлена/сконфигурирована. ALSA в Gentoo следует этому принципу. Существуют два способа, с помощью которых вы можете установить и запустить ALSA на вашей системе. Мы подробно рассмотрим их в следующей главе.

## 2. Установка ALSA

### Варианты

**Предупреждение:** Способы, описываемые ниже, являются взаимно исключающими. Вы не можете одновременно собрать ALSA в ядре и установить **media-sound/alsa-driver**. Это не получится.

Два варианта установки ALSA драйверов:

1. Использовать ALSA драйверы, предоставляемые вашим ядром. Этот метод предпочтительный и рекомендуемый.
2. Использовать пакет `media-sound/alsa-driver`.

Драйверы, предоставляемые ядром, могут немного отличаться от предоставляемых

пакетом **alsa-driver**; возможности и исправления одного могут оказаться всё ещё не включёнными в другой. Разработчики ALSA осознают ситуацию, но эти два драйвера по существу являются отдельными ветвями проекта ALSA; они не идентичны. Вы должны понимать, что они могут по-разному функционировать, поэтому если один из них у вас не работает, попробуйте другой! Мы бегло рассмотрим оба варианта перед принятием окончательного решения.

Преимущества и недостатки использования ALSA драйверов, предоставляемых ядром:

#### **ALSA в ядре      за и против**

- + Нет необходимости устанавливать ещё один пакет; драйверы включены в ядро
- + Единое решение, никаких повторных команд emerge
- Может немного отличаться от alsa-driver

А если вы выберете alsa-driver, то:

#### **Драйверы ALSA    за и против**

- + Самые свежие драйверы от проекта ALSA
- + Удобно, если вы собираетесь разрабатывать драйверы для аудио устройств
- Каждая пересборка ядра требует повторной переустановки alsa-driver
- Определенные параметры конфигурации ядра должны быть отключены

#### **Итак...**

Как сказано выше, отличия между драйверами из пакета alsa-driver и ALSA драйверами, поставляемыми с ядром, очень незначительны. Так как между ними нет большой разницы, сначала попробуйте ALSA драйвера, поставляемые ядром, так как их проще использовать. Перед тем как сообщить о любой проблеме, связанной со звуком, в [Gentoo Bugzilla](#), пожалуйста, попробуйте её воспроизвести, используя alsa-driver, и создайте запрос об ошибке вне зависимости от результата.

#### **Перед тем как вы продолжите**

Какие драйверы использует ваша карта. В большинстве случаев звуковые карты (встроенные и подключаемые) основаны на PCI, и lspci поможет вам раскопать необходимую информацию. Пожалуйста, если вы ещё не установили lspci, установите командой emerge sys-apps/pciutils. Если у вас USB звуковая карта, вам может помочь lsusb из sys-apps/usbutils. Для карт ISA попробуйте sys-apps/isapnptools. Кроме того, следующие ресурсы могут помочь владельцам ISA звуковых карт:

- [Страница ISAPNPTOOLS](#)
- [Статья о PnP в LinuxJournal](#)
- [TLDP Sound HOWTO](#)

**Примечание:** Ради простоты в оставшейся части руководства мы предположим, что у пользователя звуковая карта, основанная на PCI.

Теперь мы попробуем найти информацию о звуковой карте.

Теперь мы попробуем найти информацию о звуковой карте.

**Листинг 2.1:** Подробности звуковой карты

```
# lspci -v | grep -i audio
0000:00:0a.0 Multimedia audio controller: Creative Labs SB Live! EMU10k1 (rev 06)
```

Теперь мы знаем, что звуковая карта, установленная в компьютере, — Sound Blaster Live!, а производителем является Creative Labs. Зайдём на страницу с [таблицей звуковых карт ALSA](#) и выберем Creative Labs из списка. В результате вы попадёте на страницу, содержащую таблицу продуктов Creative Labs, из которой вы можете узнать, что SB Live! использует модуль emu10k1. Эта та информация, которая нам и нужна. Если вам интересна более подробная информация, то вы можете перейти по ссылке «Details» на страницу, посвящённую emu10k1.

Если вы намереваетесь использовать MIDI, то перед установкой любых пакетов ALSA необходимо добавить midi к USE-флагам в файле /etc/make.conf. Ниже в руководстве мы продемонстрируем, как [настроить MIDI](#).

## Использование ALSA драйверов, предоставляемых ядром

Если вам нравится идти по пути наименьшего сопротивления, то этот способ для вас.

**Примечание:** Начиная с выпуска 2005.0, Gentoo Linux в качестве основного ядра использует ядра ветки 2.6. Пожалуйста, удостоверьтесь, что у вас ядро ветки 2.6. Этот способ не применим для ядер ветки 2.4.

А теперь давайте сконфигурируем ядро так, чтобы включить в нём ALSA.

**Важно:** Пользователи genkernel должны запустить genkernel --menuconfig all и следовать инструкциям из раздела [Параметры ядра для ALSA](#).

**Листинг 2.2:** Погружение в исходные коды

```
# cd /usr/src/linux
# make menuconfig
```

**Примечание:** В только что приведённом примере предполагалось, что символическая ссылка /usr/src/linux указывает на исходные коды используемого вами ядра. Пожалуйста, перед тем как продолжить, проверьте, что у вас так оно и есть.

Теперь рассмотрим часть параметров конфигурации ядра 2.6, которые гарантируют работоспособность ALSA с нашей звуковой картой.

Обратите внимание, что во всех примерах мы собираем ALSA модулями. Мы советуем вам поступать так же, так как в дальнейшем это позволит использовать alsacnf, упрощающую настройку звуковой карты. Пожалуйста, не пропустите раздел [Настройка](#). Если вы всё же решаете не использовать модули, удостоверьтесь, что вы соответствующим образом изменили вашу конфигурацию.

### Листинг 2.3: Параметры ядра для ALSA

Device Drivers --->

Sound --->

(Это необходимо включить)

<M> Sound card support

(Убедитесь, что OSS отключена)

Open Sound System --->

< > Open Sound System (DEPRECATED)

(Вернитесь на шаг назад и войдите в раздел ALSA)

Advanced Linux Sound Architecture --->

<M> Advanced Linux Sound Architecture

(Выберите, если вам нужен MIDI sequencing и routing)

<M> Sequencer support

(Поддержка старых /dev/mixer\* и /dev/dsp\*. Рекомендуется.)

<M> OSS Mixer API

<M> OSS PCM (digital audio) API

(Теперь вы можете выбрать устройства, поддержка которых вам требуется.

Обычно в системе есть только одна звуковая карта. Если у вас их несколько, включите поддержку для каждой.)

(Для тестирования и разработки, обычным пользователям они не требуются, только если вы знаете, что делаете...)

Generic devices --->

(Для звуковых карт ISA)

ISA devices --->

(Если у вас Gravis, включите этот параметр)

<M> Gravis UltraSound Extreme

(Перейдите на один уровень назад и войдите в раздел PCI-устройств.

Большинство современных звуковых карт являются таковыми)

PCI devices --->

(Теперь выберем драйвер emu10k1 для нашей карты)

<M> Emu10k1 (SB Live!, Audigy, E-mu APS)

(Или для карты Intel)

<M> Intel/SiS/nVidia/AMD/ALi AC97 Controller

(А может у вас карта на чипсете VIA?)

<M> VIA 82C686A/B, 8233/8235 AC97 Controller

(Вернитесь на один уровень назад и, если у вас звуковая карта USB, включите)

USB Devices --->

Теперь параметры вашего ядра установлены, и вы можете пересобрать ядро. Поддержка ALSA для вашей карты будет доступной сразу, после того как вы загрузитесь с новым ядром. Чтобы использовать новое ядро, не забудьте обновить конфигурацию вашего загрузчика. Теперь, чтобы проверить, что всё работает как должно, переходите к разделу [Утилиты ALSA](#).

## Использование пакета драйверов ALSA

Итак, вы решили использовать пакет alsa-driver. Тогда начнём. Нужно выполнить несколько небольших действий для того, чтобы был скомпилирован только необходимый вашей аудио карте драйвер. Хотя это и не требуется, это сократит количество лишних драйверов, которые в противном случае были бы собраны.



Если вы не знаете, какие драйверы для звуковой карты вам могут понадобиться, ознакомьтесь с разделом, посвящённым [lspci](#) этого руководства. Как только вы узнаете имя драйвера (emu10k1 в нашем примере), добавьте переменную ALSA\_CARDS в файл /etc/make.conf.

**Листинг 2.4:** Добавление ALSA\_CARDS в make.conf

(Для одной звуковой карты)

```
ALSA_CARDS="emu10k1"
```

(Для нескольких карт разделите имена пробелами)

```
ALSA_CARDS="emu10k1 via82xx"
```

Если вы уже собирали своё ядро и хотите использовать alsa-driver, пожалуйста, удостоверьтесь в следующем, перед тем как продолжить, иначе alsa-driver, скорее всего, не установится. Следующий перечень поможет вам провести проверку.

**Примечание:** Пользователи genkernel могут продолжить с [установки alsa-driver](#), так как конфигурация их обновлённого ядра по умолчанию соответствует нижеприведённой.

1. CONFIG\_SOUND включён. (Общая поддержка звука включена)
2. CONFIG\_SOUND\_PRIME выключен. (Встроенная поддержка OSS отключена)
3. CONFIG\_SND выключен. (Встроенная поддержка ALSA отключена)
4. Символическая ссылка /usr/src/linux указывает на то ядро, в котором ALSA будет работать.

**Листинг 2.5:** .config checks

(Проверяем, что символическая ссылка указывает на нужное ядро.)

```
# cd /usr/src/linux
```

```
# grep SOUND .config
```

(Первый пункт выполнен)

```
CONFIG_SOUND=y
```

(Второй пункт выполнен)

```
CONFIG_SOUND_PRIME is not set
```

```
# grep SND .config
```

(Третий пункт выполнен)

```
CONFIG_SND is not set
```

Теперь всё, что вам нужно сделать, это набрать магические слова... нет, не абракадабру.

**Листинг 2.6:** Установка alsa-driver

```
# emerge alsa-driver
```

**Важно:** Помните, что вам придётся выполнять emerge alsa-driver каждый раз после (пере)сборки ядра, так как предыдущие драйверы будут удалены. Чтобы упростить эту задачу, вы можете воспользоваться пакетом module-rebuild, который будет вести учёт всем пакетам с модулями ядра и по необходимости пересобирать их. Сначала, чтобы создать список пакетов, выполните module-rebuild populate, а затем после (пере)сборки ядра просто запускайте module-rebuild rebuild, и все внешние модули

ядра будут пересобраны.

### 3. Настройка/тестирование ALSA

#### Утилиты ALSA

Пакет **alsa-utils** составляет неотъемлемую часть ALSA и содержит множество крайне полезных программ, в том числе и сценарий инициализации ALSA. Поэтому мы настоятельно советуем вам установить **alsa-utils**.

Листинг 3.1: Установка alsa-utils  
# emerge alsa-utils

**Примечание:** Если вы включили поддержку ALSA в вашем [ядре](#), а не собрали её в качестве модулей, пожалуйста, перейдите в раздел [сценарий инициализации ALSA](#). Всё, что осталось сделать всем остальным, это настроить ALSA. Сделать это очень просто благодаря утилите, входящей в состав alsa-utils — alsacnf.

#### Настройка

Последние версии udev (>=udev-103) в некоторой степени предоставляют автоматическое конфигурирование вашей аудио карты на уровне ядра. По возможности старайтесь полагаться на автоматическую настройку и позвольте ядру настроить вашу звуковую карту. В противном случае, чтобы настроить карту, используйте alsacnf, как это показано ниже.

**Примечание:** Пожалуйста, завершите все программы, которые могут обратиться к звуковой карте во время работы **alsacnf**.

Чтобы настроить вашу звуковую карту просто наберите alsacnf в командной оболочке с правами root.

Листинг 3.2: Запуск alsacnf  
# alsacnf

Вы увидите изящный управляемый с помощью меню интерфейс программы, которая автоматически исследует ваши устройства и попытается найти вашу звуковую карту. Вас попросят выбрать вашу звуковую карту из списка. Как только это будет сделано, вас спросят разрешения автоматически сделать необходимые изменения в /etc/modules.d/alsa. После этого программа изменит настройки громкости на оптимальный уровень, выполнит update-modules и запустит службу /etc/init.d/alsasound. Как только alsacnf завершит работу, вы сможете продолжить с настройки сценария инициализации ALSA.

#### Сценарий инициализации ALSA

Мы почти завершили настройку. Вне зависимости от выбранного вами способа установки ALSA, вам понадобится что-то, что будет загружать модули или инициализировать ALSA и восстанавливать настройки громкости при загрузке системы. Сценарий инициализации ALSA, называемый alsasound, делает всё это

для вас. Добавьте его в загрузочный уровень исполнения.

**Листинг 3.3:** Добавление ALSA в загрузочный уровень исполнения

```
# rc-update add alsasound boot
```

```
* alsasound added to runlevel boot
```

```
* rc-update complete.
```

Теперь проверьте файл `/etc/conf.d/alsasound` и убедитесь, что переменная `SAVE_ON_STOP` установлена в значение `yes`, тогда ваши настройки звука при выключении системы будут сохраняться.

## Группа audio

Перед тем как начать проверку звука, необходимо сделать одну последнюю важную вещь. Главное правило в операционных системах \*nix гласит: «не запускай ничего с правами `root`, если этого не требуется». И в данной ситуации это правило тоже применимо ;) Каким образом? Пожалуй, большую часть времени вы будете работать под учётной записью пользователя, при этом слушать музыку или иметь доступ к звуковой карте. Чтобы вы могли это делать, вы должны быть в группе `audio`. Тут то мы и добавим пользователей в эту группу, чтобы у них не было проблем с доступом к аудио устройствам. Мы воспользуемся `grasswd`, так что вы должны иметь права `root`, чтобы сделать это.

**Листинг 3.4:** Добавление пользователей в группу `audio`

(Замените `<имя_пользователя>` на необходимое имя)

```
# grasswd -a <имя_пользователя> audio
```

```
Adding user <имя_пользователя> to group audio
```

## Проверка громкости!

Теперь все настройки и необходимые предпосылки выполнены, так что давайте заставим ALSA работать. Если вы запускали утилиту **alsacnf**, то можете пропустить этот шаг, так как `alsacnf` уже всё сделал за вас.

**Листинг 3.5:** Запуск сервиса

```
/etc/init.d/alsasound start
```

Теперь мы позаботились обо всём, что может быть необходимым, нам нужно проверить громкость, так как в определённых случаях звук выключен. Для этого мы воспользуемся `alsamixer`.

**Листинг 3.6:** Запуск `alsamixer`

(Откройте в терминале. Будут отображены только необходимые настройки)

```
# alsamixer
```

**Важно:** Если у вас проблемы с запуском `alsamixer` и вы получаете ошибки, подобные этой: «`alsamixer: function snd_ctl_open failed for default: No such file or directory`», то, скорее всего, проблема в том, как демон `udev` инициализировал устройства. Выполните `killall udevd; udevstart` для перезагрузки устройств в `/dev` и попробуйте снова `alsamixer`. Это должно решить проблему.

Вот так может выглядеть микшер ALSA при первом запуске. Обратите внимание, что уровни каналов Master и PCM занижены и что под ними буквы MM. Это означает, что они выключены. Если вы попытаете что-нибудь проиграть, в то время пока alsamixer в таком состоянии, то вы ничего не услышите из ваших динамиков.

Для начала включим каналы и установим необходимые уровни громкости.

**Предупреждение:** Оба канала Master и PCM должны быть включены и установлены в слышимый уровень громкости, если вы хотите услышать что-нибудь из ваших динамиков.

- Для перемещения между каналами используйте клавиши влево и вправо. (<- и ->).
- Для того чтобы выключить/включить канал, например Master, выберите его и нажмите клавишу m.
- Чтобы увеличить или уменьшить уровень громкости, используйте клавиши вверх и вниз.

**Примечание:** Будьте осторожны при установке значений для Bass и Treble. Обычно для обоих оптимально значение 50. Слишком высокие значения Bass могут вызвать дребезжание неспособных воспроизводить глубокие басы динамиков.

После того как вы всё сделаете, ваш микшер ALSA должен выглядеть, так, как это показано ниже. Заметьте, что вместо MM стоит 00 и уровни громкости в оптимальном значении.

## Проверка звука!

Наконец-то. Какие-нибудь звуки. Если всё прошло успешно, то теперь вы сможете услышать музыку. Быстрый способ проверить — это воспользоваться консольной командой, наподобие media-sound/madplay. Вы также можете использовать что-нибудь более известное, например, mpg123. Если вы поклонник формата OGG, используйте программу ogg123 из пакета media-sound/vorbis-tools. Используйте любой удобный для вас плеер. Как всегда, emerge — всё что вам нужно.

**Листинг 3.7:** Получение программ  
(Установка необходимых приложений)  
**# emerge madplay mpg123**  
(Для проигрывания ogg-файлов)  
**# emerge vorbis-tools**

А теперь проиграем ваш любимый саундтрек...

**Листинг 3.8:** Проигрывание музыки  
**# madplay -v /mnt/shyam/Music/Paul\ Oakenfold\ -\ Dread\ Rock.mp3**  
MPEG Audio Decoder 0.15.2 (beta) - Copyright (C) 2000-2004 Robert Leslie et al.  
Title: Dread Rock  
Artist: Paul Oakenfold  
Album: Matrix Reloaded  
Year: 2003

Genre: Soundtrack  
Soundtrack  
00:04:19 Layer III, 160 kbps, 44100 Hz, joint stereo (MS), no CRC  
  
# ogg123 Paul\ Oakenfold\ -\ Dread\ Rock.ogg  
Audio Device: Advanced Linux Sound Architecture (ALSA) output  
  
Playing: Paul Oakenfold - Dread Rock.ogg  
Ogg Vorbis stream: 2 channel, 44100 Hz  
Genre: Soundtrack  
Transcoded: mp3;160  
Title: Dread Rock  
Artist: Paul Oakenfold  
Date: 2003  
Album: Matrix Reloaded  
Time: 00:11.31 [04:28.75] of 04:40.06 (200.6 kbps) Output Buffer 96.9%

## ALSA и USE

Теперь, для того чтобы приложения, поддерживающие ALSA, были собраны с поддержкой оной, вам нужно добавить USE-флаг `alsa` в файл `/etc/make.conf`. Некоторые архитектуры, например, `x86` и `amd64`, включают этот флаг по умолчанию.

### Проблемы?

Если по каким-либо причинам вы не слышите звука, перво-наперво проверьте настройки [alsamixer](#). 80% всех проблем заключаются в выключенных каналах или низкой громкости. Кроме того, проверьте звуковой апплет вашего оконного менеджера и удостоверьтесь, что громкости каналов установлены на слышимом уровне.

`/proc` — ваш друг. В этом случае, `/proc/asound` — ваш лучший друг. Мы бегло рассмотрим, сколько полезной информации доступно нам здесь.

#### Листинг 3.9: Развлечение с `/proc/asound`

(Прежде всего, если `/proc/asound/cards` отображает вашу карту, то ALSA распознала вашу звуковую карту.)

```
# cat /proc/asound/cards
```

```
0 [Live      ]: EMU10K1 - Sound Blaster Live!  
                Sound Blaster Live! (rev.6, serial:0x80271102) at 0xb800, irq 11
```

(Следующая команда отобразит текущую версию ALSA)

```
# cat /proc/asound/version
```

```
Advanced Linux Sound Architecture Driver Version 1.0.8 (Thu Jan 13 09:39:32 2005 UTC).
```

(Подробности эмуляции OSS в ALSA)

```
# cat /proc/asound/oss/sndstat
```

```
Sound Driver:3.8.1a-980706 (ALSA v1.0.8 emulation code)
```

```
Kernel: Linux airwolf.zion 2.6.11ac1 #2 Wed May 4 00:35:08 IST 2005 i686
```

```
Config options: 0
```

Installed drivers:

```
Type 10: ALSA emulation
```

Card config:

```
Sound Blaster Live! (rev.6, serial:0x80271102) at 0xb800, irq 11
```

Audio devices:  
0: EMU10K1 (DUPLEX)

Synth devices: NOT ENABLED IN CONFIG

Midi devices:  
0: EMU10K1 MPU-401 (UART)

Timers:  
7: system timer

Mixers:

Другой очень распространённой проблемой среди пользователей является странная ошибка «Unknown symbol in module» («Неизвестный символ в модуле»). Пример её появления показан ниже.

**Листинг 3.10:** Ошибка: неизвестный символ в модуле

**# /etc/init.d/alsasound start**

```
* Loading ALSA modules ...
* Loading: snd-card-0 ... [ ok ]
* Loading: snd-pcm-oss ...
WARNING: Error inserting snd_mixer_oss
(/lib/modules/2.6.12-gentoo-r6/kernel/sound/core/oss/snd-mixer-oss.ko): Unknown
symbol in module, or unknown parameter (see dmesg) FATAL: Error inserting
snd_pcm_oss
(/lib/modules/2.6.12-gentoo-r6/kernel/sound/core/oss/snd-pcm-oss.ko): Unknown
symbol in module, or unknown parameter (see dmesg)
[ !! ]
* Loading: snd-mixer-oss ...
FATAL: Error inserting snd_mixer_oss
(/lib/modules/2.6.12-gentoo-r6/kernel/sound/core/oss/snd-mixer-oss.ko): Unknown
symbol in module, or unknown parameter (see dmesg)
[ !! ]
* Loading: snd-seq ... [ ok ]
* Loading: snd-emu10k1-synth ... [ ok ]
* Loading: snd-seq-midi ... [ ok ]
* Restoring Mixer Levels ... [ ok ]
```

И если, как советуют, посмотреть вывод `dmesg`, то, скорее всего, можно увидеть следующее:

**Листинг 3.11: Вывод `dmesg`**

(Показаны только относящиеся к делу части вывода)

**# dmesg | less**

```
ACPI: PCI Interrupt 0000:02:06.0[A] -> Link [APC3] -> GSI 18 (level, low) -> IRQ 209
snd_mixer_oss: Unknown symbol snd_unregister_oss_device
snd_mixer_oss: Unknown symbol snd_register_oss_device
snd_mixer_oss: Unknown symbol snd_mixer_oss_notify_callback
snd_mixer_oss: Unknown symbol snd_oss_info_register
snd_pcm_oss: Unknown symbol snd_unregister_oss_device
snd_pcm_oss: Unknown symbol snd_register_oss_device
snd_pcm_oss: Unknown symbol snd_mixer_oss_ioctl_card
snd_pcm_oss: Unknown symbol snd_oss_info_register
snd_mixer_oss: Unknown symbol snd_unregister_oss_device
snd_mixer_oss: Unknown symbol snd_register_oss_device
snd_mixer_oss: Unknown symbol snd_mixer_oss_notify_callback
snd_mixer_oss: Unknown symbol snd_oss_info_register
```

Эта проблема вызвана переключением с alsa-driver на драйверы, предоставляемые ядром, потому что когда вы удаляете alsa-driver, то файлы модулей сохраняются системой защиты конфигурации. И поэтому, когда вы переходите на встроенные в ядро драйверы, попытка modprobe выдаст вам смесь модулей из alsa-driver и встроенных в ядро, вызывая приведённые выше ошибки.

Решение очень простое. Удалите вызывающий проблемы каталог после удаления alsa-driver. Проверьте, что удаляете модули правильной, но не текущей версии ядра!

**Листинг 3.12:** Удаление модулей alsa-driver

```
# rm -rf /lib/modules/$(uname -r)/alsa-driver
```

Другой причиной подобных сообщений может являться файл в /etc/modules.d, содержащий параметр device\_mode, в то время как он не требуется. Проверьте, так ли это, и выясните, какой файл является источником проблем.

**Листинг 3.13:** Подтверждение проблемы и поиск device\_mode

(Проверим вывод dmesg для идентификации проблемы)

```
# dmesg | grep device_mode
```

```
snd: Unknown parameter `device_mode'
```

(И теперь найдём источник проблемы)

```
# grep device_mode /etc/modules.d/*
```

Обычно это файл с именем alsa, в котором присутствует строка options snd device\_mode=0666. Удалите эту строку и перезапустите службу alsasound. Это должно решить проблему.

## 4. Другие возможности ALSA

### Настройка поддержки MIDI

Сначала проверьте, что у вас включён USE-флаг **midi** в файле **/etc/make.conf**. Если вы до сих пор этого не сделали, то сделайте это сейчас. Помимо этого вам потребуется пересобрать все пакеты ALSA, использующие флаг midi: alsa-lib, alsa-utils и alsa-driver.

Если ваша карта имеет встроенный MIDI синтезатор и вы хотите слушать \*.mid файлы, вам придётся установить пакет awesfx, содержащий основной набор программ для управления драйвером AWE32. Сначала вам надо установить его. Если у вас нет синтезатора, вы можете использовать виртуальный. Обратитесь к разделу [виртуальные синтезаторы](#) для дополнительной информации.

**Листинг 4.1:** Установка awesfx

```
# emerge awesfx
```

**Примечание:** Вам понадобится скопировать файлы банка звуков (SoundFont: SF2) с компакт-диска с драйверами для вашей звуковой карты или установленные в Windows в каталог /usr/share/sounds/sf2/. Например, файл банка звуков для карты Creative SBLive! может называться 8MBGMSFX.SF2.

После копирования файлов банка звуков мы сможем проигрывать midi-файлы. Также, для того чтобы банк звуков загружался каждый раз при загрузке системы, вы можете добавить команду **asfxload** в **/etc/conf.d/local.start**.

**Примечание:** Используемые ниже пути, начинающиеся с /mnt, не применимы для вашего компьютера. Они являются примерами. Пожалуйста, будьте осторожны при изменении путей на соответствующие вашей системе.

#### Листинг 4.2: Загрузка сэмплов

(Сначала копируем файл с банком звуков)

```
# cp /mnt/win2k/Program\ Files/CreativeSBLive2k/SFBank/8MBGMSFX.SF2 /usr/share/sounds/sf2/
```

(Или берём его с компакт-диска SoundBlaster)

```
# cp /mnt/cdrom/AUDIO/ENGLISH/SFBANK/8MBGMSFX.SF2 /usr/share/sounds/sf2/
```

(Загружаем определённый банк звуков)

```
# asfxload /usr/share/sounds/sf2/8MBGMSFX.SF2
```

Теперь вы можете проигрывать midi-файлы, используя программу, подобную **aplaymidi**. Запустите **aplaymidi -l**, чтобы получить список доступных портов, а затем задействуйте один из них для проигрывания файла.

#### Листинг 4.3: Проигрывание MIDI

(Проверка открытых портов)

```
# aplaymidi -l
```

Port	Client name	Port name
64:0	EMU10K1 MPU-401 (UART)	EMU10K1 MPU-401 (UART)
65:0	Emu10k1 WaveTable	Emu10k1 Port 0
65:1	Emu10k1 WaveTable	Emu10k1 Port 1
65:2	Emu10k1 WaveTable	Emu10k1 Port 2
65:3	Emu10k1 WaveTable	Emu10k1 Port 3

(Выбираем порт и проигрываем mid-файл)

```
# aplaymidi --port=65:0 /mnt/shyam/music/midi/mi2.mid
```

## Виртуальные синтезаторы

Если у вашей карты отсутствует встроенный синтезатор, вы можете использовать виртуальный, например, **timidity++**. Установка происходит на одном дыхании.

#### Листинг 4.4: Установка timidity++

```
# emerge timidity++
```

Чтобы **timidity** воспроизводил звуки, ему нужен набор сэмплов (или банк звуков). К счастью, вместе с пакетом устанавливаются несколько банков звуков. Есть ещё несколько пакетов с банками звуков в Portage, например, **timidity-freepats** и **timidity-eawpatches**. Вы можете установить несколько банков звуков, а также можете разместить собственный банк звуков в каталог **/usr/share/timidity/**. Для переключения **timidity** между разными банками звуков используйте **eselect**.

#### Листинг 4.5: Изменение конфигураций

```
# eselect timidity list
```

```
# eselect timidity set eawpatches
```



Не забудьте добавить `timidity` в основной уровень исполнения.

**Листинг 4.6:** Добавление `timidity` в основной уровень исполнения

```
# rc-update add timidity default
# /etc/init.d/timidity start
```

Теперь вы можете попробовать [проиграть MIDI](#) файлы.

## Утилиты и Firmware

Для некоторых звуковых карт могут быть полезны утилиты из пакетов **alsa-tools** и **alsa-firmware**. Вы можете установить любой из этих пакетов, просто запустив `emerge`.

**Листинг 4.7:** Установка утилит ALSA

```
# emerge alsa-tools
```

## Несколько звуковых карт

Вы можете использовать больше одной звуковой карты, при условии что вы собрали ALSA как модули ядра (или из пакета `alsa-driver`). Сначала в файле `/etc/modules.d/alsa` вам следует лишь указать, какая из карт должна быть запущена первой. В этом файле карты идентифицируются по именам своих драйверов. 0 означает первую карту, 1 — вторую, и так далее. Вот пример для системы, в которой присутствуют две звуковые карты.

**Листинг 4.8:** Две карты в файле `/etc/modules.d/alsa`

```
options snd-emu10k1 index=0
options snd-via82xx index=1
```

Или если у вас две карты, использующие один и тот же драйвер, то их следует указать на одной строке, разделяя цифры запятой. Ниже приведён пример системы, в которой установлено три звуковые карты, две из которых являются картами Intel High Definition Audio.

**Листинг 4.9:** Несколько звуковых карт в `/etc/modules.d/alsa`

```
options snd-ymfpci index=0
options snd-hda-intel index=1,2
```

## Модули расширения

Для расширения возможностей ALSA вы можете установить дополнительные расширения. Пакет `alsa-plugins` является собранием полезных модулей расширения, который включает: модуль вывода PulseAudio, конвертер частоты дискретизации, `jack` (аудио сервер с малым временем задержки) и кодек, позволяющий вам выводить шестиканальный звук через цифровой S/PDIF-вывод (оптический или коаксиальный). Вы можете выбрать те расширения, которые вы хотите установить, добавив соответствующие USE-флаги в `/etc/portage/package.use`.

**Листинг 4.10:** Установка `alsa-plugins`

```
# emerge -avt alsa-plugins
```

## Огромное спасибо вам всем...

Всем, кто принимал участие в написании ранних версий руководства ALSA в Gentoo: Vincent Verleye, Grant Goodyear, Arcady Genkin, Jeremy Huddleston, John P. Davis,

Sven Vermeulen, Benny Chuang, Tiemo Kieft и Erwin. А также Dr][аМ за перевод ранней версии руководства на русский язык.

Ссылки

- [Домашняя страница проекта ALSA](#)
- [Linux Sound/MIDI Software](#)

# Руководство по настройке Java в Gentoo

*Ссылка на оригинал:*

<http://www.gentoo.org/doc/ru/java.xml>

*С версии: 1.0.*

## 1. Установка JDK/JRE

### Введение

Gentoo предоставляет несколько JDK и JRE. По умолчанию выбран Blackdown JDK/JRE, так как он свободно доступен без регистрации.

Примечание: Как только kaffe станет легко встраиваемой JRE/JDK, возможно, мы примем её по умолчанию.

И Sun JDK/JRE, и IBM JDK/JRE вообще-то значительно быстрее, но получение их требует намного большего труда, поскольку вам потребуется прочитать и принять их лицензии перед загрузкой (IBM дополнительно требует, чтобы вы зарегистрировались).

Наши ebuild-скрипты для Sun и IBM JDK/JRE уведомят вас относительно того, откуда всё это можно загрузить.

### Установка Sun/IBM JDK/JRE

Если вы запустите `emerge sun-jdk-1.3.1` или `emerge ibm-jdk-1.3.1`, вы будете уведомлены о том, что требуется, чтобы вы загрузили архив фактически самостоятельно. Это происходит из-за ограничений лицензии Sun JDK/JRE (online click-wrap license) и проблем с регистрацией IBM JDK/JRE.

Примечание: Есть также `sun-jdk-1.4.0`, но не все пакеты хорошо работают с Java 1.4, так, что выбор всегда за вами: использовать 1.4.0 JDK или нет.

Вы должны загрузить обозначенные файл(ы) в `/usr/portage/distfiles`. После чего можете повторно запустите команду `emerge`, JDK/JRE будет установлен должным образом в `/opt`.

## 2. Конфигурирование вашей JDK/JRE

### Краткий обзор

Gentoo позволяет бесконфликтно устанавливать несколько JDK и JRE на одной машине. На этот счёт есть несколько предостережений, отмеченных ниже.

Используя инструмент `java-config` и имея `root`-доступ, вы можете установить общесистемное значение по умолчанию. Пользователи также могут использовать `java-config` для установки их собственных значений по умолчанию, отличных от общесистемного значения.

## Выбор основного JDK/JRE

Выполнение команды `java-config --list-available-vms` выдаст список всех доступных JRE/JDK на вашей системе. Что-то вроде:

**Листинг 2.1:** Список доступных VM

```
[%1 ~] java-config --list-available-vms
```

```
[blackdown-jdk-1.3.1] Blackdown JDK 1.3.1 (/etc/env.d/java/20blackdown-jdk-1.3.1)
[blackdown-jre-1.3.1] Blackdown JRE 1.3.1 (/etc/env.d/java/20blackdown-jre-1.3.1)
[ibm-jdk-1.3.0] IBM JDK 1.3.0 (/etc/env.d/java/20ibm-jdk-1.3.0)
[ibm-jdk-1.3.1] IBM JDK 1.3.1 (/etc/env.d/java/20ibm-jdk-1.3.1)
[ibm-jre-1.3.1] IBM JRE 1.3.1 (/etc/env.d/java/20ibm-jre-1.3.1)
[sun-jdk-1.4.0] Sun JDK 1.4.0 (/etc/env.d/java/20sun-jdk-1.4.0)
```

Название в скобках "[ ]" является дескриптором или идентификатором для VM. Используйте идентификатор с `java-config --set-system-vm`, примерно так:

**Листинг 2.2:** Выбор системной VM

```
[#1 ~] java-config --set-system-vm=ibm-jdk-1.3.1
Now using IBM JDK 1.3.1 (/etc/env.d/java/20ibm-jdk-1.3.1)
```

**Примечание:** Вы должны иметь права `root` для запуска `--set-system-vm`

После запуска `java-config --set-system-vm` с VM идентификатором, нужно обновить ваш `/etc/profile.env`, так:

**Листинг 2.3:** Обновление `/etc/profile.env`

```
[#1 ~] env-update
```

После чего нужно повторно войти в систему, чтобы перечитать файл рабочей среды `/etc/profile`.

Как обычный пользователь, вы можете использовать `java-config --set-user-vm`, который создаст `$HOME/.gentoo/java-env` со всеми требуемыми `env` переменными. Обычно источник переменных - это сценарий запуска вашей оболочки (`$HOME/.zshenv` в моём случае).

## Установка значения CLASSPATH

`java-config` также может быть использован для установки системного значения `CLASSPATH`, и конечно пользовательского значения `CLASSPATH`.

Первое, что вам нужно - это список доступных библиотек Java, необходимые из которых нужно поместить в `CLASSPATH`. Приблизительно это выглядит так:

**Листинг 2.4:** Список классов

```
[%1 ~] java-config --list-available-packages
[ant] No description (/usr/share/ant/classpath.env)
```

```
[java-gnome] No description (/usr/share/java-gnome/classpath.env)
[java-gtk] No description (/usr/share/java-gtk/classpath.env)
[log4j] "" (/usr/share/log4j/package.env)
```

**Примечание:** Ни один из этих пакетов не имеет надлежащего описания. Это будет осуществлено в ближайшем будущем.

Снова названия в скобках "[]" являются идентификаторами, которые вы должны передать в `java-config --set-system-classpath`, таким образом:

**Листинг 2.5:** Выбор classpath

```
java-config --set-system-classpath=log4j,java-gtk,java-gnome
```

**Примечание:** Директория в (.) не будет добавлена в classpath, если это не прописано в profile файле root.

Снова вы должны будете выполнить `env-update`, чтобы модифицировать среду вашей системы, и повторно войти чтобы перечитать `/etc/profile`.

Для пользователей `java-config --set-user-classpath` создаст `$HOME/.gentoo/java-env-classpath`, который автоматически будет включён в `$HOME/.gentoo/java-env`.

### 3. Дополнительные ресурсы

#### Off-line ресурсы

`man java-config`

`java-config --help`

Непосредственно сам скрипт `/usr/bin/java-config`

#### Online ресурсы

Архив почтовой рассылки `gentoo-dev`, `gentoo-user`

`#gentoo` на `irc.openprojects.net`

### 4. Предостережения

#### Основное

Если Вы выбираете JRE VM по умолчанию, команда `javac`, будет не доступна до тех пор пока вы не сделаете символическую ссылку.

Jikes будет использоваться для компиляции некоторых больших пакетов. Со временем мы надеемся скомпилировать все java зависимые пакеты с Jikes, так некоторые зависимости от JDK могут быть заменены зависимостью от JRE.

В некоторых версиях Portage при выполнении `emerge --world update` будет установлено JDK, независимо от того, устанавливали ли вы его прежде или нет. Это - известная ошибка, исправленная в более поздних версиях Portage.

# Gentoo и GPRS

## GPRS через USB (Руководство)

Ссылка на оригинал: <http://linuxforum.ru/>

Автор: Nikoli (Linuxforum.ru)

С версии: 1.3

Появилась необходимость в GPRS соединении в качестве страховочного. Телефоны подключаю по USB. Потратил пару часов на настройку, но в итоге получилось довольно простое решение. Здесь для Gentoo такого руководства не выкладывали, надеюсь моё будет полезно. Если считаете необходимым, добавьте его в FAQ.

1) Ядро должно быть собрано с:

```
Device Drivers --->
  USB support --->
    <M> USB Modem (CDC ACM) support
```

2) Должен быть установлен **ppp**

3) В **/etc/conf.d/net** нужно добавить:

```
#*****/-----Nokia with Megafon
config_ppp1=( "ppp" )
link_ppp1="ttyACM0"
pppd_ppp1=(
"debug" # Comment this off, if you don't need more info
#460800 # Serial port line speed
"crtcts" # hardware flow control for cable
"local" # Ignore carrier detect signal from the modem:
"lcp-echo-failure 0"
"lcp-echo-interval 0"
# IP addresses:
":0.0.0.0"
# - accept peers idea of our local address and set address peer as 10.6.6.6
# (any address would do, since IPCP gives 0.0.0.0 to it)
# - if you use the 10. network at home or something and pppd rejects it,
# change the address
"noipdefault" # pppd must not propose any IP address to the peer!
"ipcp-accept-local" # Accept peers idea of our local address
"defaultroute" # Add the ppp interface as default route to the IP routing table
#replacedefaultroute # New route should be our default route to Internet
"usepeerdns" # User DNS returned by server
"noauth" # The phone is not required to authenticate
# Most phone do not support compression, so turn it off.
#noipv6
#novj
#nobsdcomp
#novjccomp
#nopcomp
#noaccomp
# Username and password:
# If username and password are required by the APN, put here the username
```

```

# and put the username-password combination to the secrets file:
# /etc/ppp/pap-secrets for PAP and /etc/ppp/chap-secrets for CHAP
# authentication. See pppd man pages for details.
'user "internet" # Change this
'persist' # Persistent connection
'maxfail 99999' # Retry and retry and retry if failed...
)
chat_ppp1=(
'TIMEOUT' '45'
'ABORT' 'BUSY'
'ABORT' 'NO CARRIER'
'ABORT' 'ERROR'
" 'ATE1'
'OK' 'AT+CGDCONT=1,"IP","internet"'
'OK' 'ATD*99***1#'
'CONNECT' 'c'
)

#*****/-----Motorola with Beeline
config_ppp2=( "ppp" )
link_ppp2="ttyACM0"
pppd_ppp2=(
"debug" # Comment this off, if you don't need more info
#460800 # Serial port line speed
"crtcts" # hardware flow control for cable
"local" # Ignore carrier detect signal from the modem:
"lcp-echo-failure 0"
"lcp-echo-interval 0"
# IP addresses:
":0.0.0.0"
# - accept peers idea of our local address and set address peer as 10.6.6.6
# (any address would do, since IPCP gives 0.0.0.0 to it)
# - if you use the 10. network at home or something and pppd rejects it,
# change the address
"noipdefault" # pppd must not propose any IP address to the peer!
"ipcp-accept-local" # Accept peers idea of our local address
"defaultroute" # Add the ppp interface as default route to the IP routing table
#replacedefaultroute # New route should be our default route to Internet
"usepeerdns" # User DNS returned by server
"noauth" # The phone is not required to authenticate
# Most phone do not support compression, so turn it off.
#noipv6
#novj
#nobsdcomp
#novjccomp
#nopcomp
#noaccomp
# Username and password:
# If username and password are required by the APN, put here the username
# and put the username-password combination to the secrets file:
# /etc/ppp/pap-secrets for PAP and /etc/ppp/chap-secrets for CHAP
# authentication. See pppd man pages for details.
'user "internet.beeline.ru" # Change this
'persist' # Persistent connection
'maxfail 99999' # Retry and retry and retry if failed...
)
chat_ppp2=(
'TIMEOUT' '45'
'ABORT' 'BUSY'
'ABORT' 'NO CARRIER'
'ABORT' 'ERROR'
" 'ATE1'
'OK' 'AT+CGDCONT=1,"IP","INTERNET.MSK"'
'OK' 'ATD*99***1#'
'CONNECT' 'c'

```

)

4) Сделайте ссылки на *net.lo*

```
In -s /etc/init.d/net.lo /etc/init.d/net.ppp1  
In -s /etc/init.d/net.lo /etc/init.d/net.ppp2
```

5) Подключите телефон с помощью USB, появится */dev/ttyACM0*

6) Выполните

```
/etc/init.d/net.ppp1 start
```

Возможно у вас появится соединение с интернетом.

Полезно почитать вывод **tail -f /var/log/messages** :

```
Oct 27 01:18:03 kompyc pppd[9874]: pppd 2.4.4 started by root, uid 0  
Oct 27 01:18:03 kompyc chat[9877]: timeout set to 45 seconds  
Oct 27 01:18:03 kompyc chat[9877]: abort on (BUSY)  
Oct 27 01:18:03 kompyc chat[9877]: abort on (NO CARRIER)  
Oct 27 01:18:03 kompyc chat[9877]: abort on (ERROR)  
Oct 27 01:18:03 kompyc chat[9877]: send (ATE1^M)  
Oct 27 01:18:03 kompyc chat[9877]: expect (OK)  
Oct 27 01:18:03 kompyc chat[9877]: ATE1^M^M  
Oct 27 01:18:03 kompyc chat[9877]: OK  
Oct 27 01:18:03 kompyc chat[9877]: -- got it  
Oct 27 01:18:03 kompyc chat[9877]: send (AT+CGDCONT=1,"IP","internet"^M)  
Oct 27 01:18:04 kompyc chat[9877]: expect (OK)  
Oct 27 01:18:04 kompyc chat[9877]: ^M  
Oct 27 01:18:04 kompyc chat[9877]: AT+CGDCONT=1,"IP","internet"^M^M  
Oct 27 01:18:04 kompyc chat[9877]: OK  
Oct 27 01:18:04 kompyc chat[9877]: -- got it  
Oct 27 01:18:04 kompyc chat[9877]: send (ATD*99***1#^M)  
Oct 27 01:18:04 kompyc chat[9877]: expect (CONNECT)  
Oct 27 01:18:04 kompyc chat[9877]: ^M  
Oct 27 01:18:05 kompyc chat[9877]: ATD*99***1#^M^M  
Oct 27 01:18:05 kompyc chat[9877]: CONNECT  
Oct 27 01:18:05 kompyc chat[9877]: -- got it  
Oct 27 01:18:05 kompyc chat[9877]: send (c^M)  
Oct 27 01:18:05 kompyc pppd[9874]: Serial connection established.  
Oct 27 01:18:05 kompyc pppd[9874]: Using interface ppp1  
Oct 27 01:18:05 kompyc pppd[9874]: Connect: ppp1 <--> /dev/ttyACM0  
Oct 27 01:18:06 kompyc pppd[9874]: local IP address 10.205.145.43  
Oct 27 01:18:06 kompyc pppd[9874]: remote IP address 10.6.6.6  
Oct 27 01:18:06 kompyc pppd[9874]: primary DNS address 10.77.48.2  
Oct 27 01:18:06 kompyc pppd[9874]: secondary DNS address 10.77.48.3
```

Проверял на том, что было. С отключённым сжатием у меня заработало, у вас его возможно нужно выключить. Если у вас есть замечания по улучшению, буду рад.  
По мотивам:

[http://gentoo-wiki.com/Nokia\\_S60\\_and\\_N\\_series](http://gentoo-wiki.com/Nokia_S60_and_N_series)

[http://gentoo-wiki.com/HOWTO\\_Phone\\_sync](http://gentoo-wiki.com/HOWTO_Phone_sync)

[http://www.opennet.ru/base/modem/gprs\\_linux.txt.html](http://www.opennet.ru/base/modem/gprs_linux.txt.html)



## **Настройка GPRS в Linux**

Ссылка на оригинал: <http://ru.gentoo-wiki.com>

С версии: 1.3

### **Введение**

Это руководство поможет настроить мобильный телефон для связи с провайдером через GPRS (на примере beeline). Для того, чтобы настроить мобильный телефон для связи через GPRS, вам понадобится подключить на вашем телефоне услугу GPRS. Обзавестись Mini-USB кабелем. Поправить ручками кое-что в ядре и собрать дополнительные пакеты из набора portage.

### **Подготовка**

Для наших задач нам понадобится изменить несколько файлов настроек.

Я использовал ядро gentoo-sources-2.4.28-gentoo-r5 из portage,

дозвонщик wvdial оттуда же

и ppp-2.4.3-r1 тоже соответственно из portage.

Изменяем файлы:

Открываем /etc/make.conf для редактирования nano -w /etc/make.conf

Добавляем mppe-mppc (Нужно для подключения с шифрованием Microsoft MS-CHAPv2)к вашим USE Файл: /etc/make.conf

**USE="mppe-mppc"**

### **Сборка пакетов**

Собираем rppd:

**emerge /usr/portage/net-dialup/ppp/ppp-2.4.3-r1.ebuild**

собираем wvdial:

**emerge wvdial**

редактируем ядро

## **cd /usr/src/linux && make menuconfig**

(подразумевается что linux это символическая ссылка на исходники ядра)

изменяем:

Network Device Support --->

<M> (PPP point-to-point protocol) support

[ ] PPP multilink support (EXPEREMRNTAL) (Вы же не хотите объединять несколько телефонов в один канал)

[M] PPP filtering

<M> PPP support for async serial ports

<M> PPP support for sync tty ports

<M> PPP Deflate compression

<M> PPP BSD-Compress compression

<M> PPP over Ethernet (EXPEREMENTAL)

**Примечание** - т.к. у меня ядро 2.6.12-gentoo-r6, то выше названные модули я просто включил в ядро, и не возился в дальнейшем с их загрузкой. Пока проблем не было.

ISDN Subsystem --->

<\*> ISDN Support

[\*] Support asynchronus PPP

[\*] PPP filtering for ISDN

[\*] Use VJ-compression with asynchronus PPP

USB supprot --->

[\*] Support for USB

--- USB Host Controller Driver

<\*> ECHI HCD (USB2.0) support (EXPEREMENTAL)

(так как у меня чип материнской платы Intel то)

<\*> UHCI (Intel PIIX4, VIA, ...) support

--- USB Device Class drivers

<M> USB Modem (CDC ACM) support

**Примечание** - для ядра 2.6.12-gentoo-r6 нужно установить:

Device Drivers ---> ISDN subsystem --->

ISDN subsystem --->

[\*] CAPI2.0 Middleware support (Может и не нужно, но с ним работает)

Для C380 в данном разделе остальное у меня отключено.

И по меньшей мере для C380 дополнительно требуется установить:

Device Drivers ---> USB support ---> USB Serial Converter support --->

<\*> USB Serial Converter support

[\*] USB Serial Console device support

Сохраняем конфигурацию ядра и выполняем сборку. После сборки устанавливаем новое ядро

(Процесс установки ядра описан во многих источниках так что не буду повторяться) и перезагружаем машину.

## Настройка

После перезагрузки пытаемся загрузить модули

перед всеми манипуляциями проверить есть ли в /dev устройство ttyACM0, иначе скрипт и терминалка не увидят модема (телефона) по крайней мере в Fedora Core 2.0 если с ttyUSB0(1,2,3,4,5) ничего не вышло наберите находясь в /dev "mknod ttyACM0 c 160 0" (спасибо Pooh) и все должно заработать, естественно что модемом после этого будет устройство ttyACM0 (Yuri г. Barnaul)

- стандартный модуль PPP modprobe ppp\_generic
- Модуль для асинхронной передачи modprobe ppp\_async
- Модуль для нашего телефона modprobe acm

Проверяем загрузились модули **lsmod**

Вывод должен выглядеть примерно так --->

```
ppp_async      6848  0 (unused)
ppp_generic    22628  0 [ppp_async]
slhc           4624  0 [ppp_generic]
acm            45353  0 [xxxxxxxxxxx] (не помню точно что должно быть)
```

Производим первичную конфигурацию wvdial

**cd /etc && wvdialconf wvdial.conf**

открываем для редактирования /etc/wvdial.conf

**nano -w wvdial.conf Файл: /etc/wvdial.conf**

и изменяем следующие строки

```
Init2 = AT+CGDCONT=1,"IP","internet.beeline.ru"
```

```
Phone = *99***1#
```

```
Username = beeline
```

```
Password = beeline
```

сохраняем полученный файл. Если значение Init2 уже занато (для C380 в нем дополнительные команды инициализации), то используйте значение Init3 (Init3 = AT+CGDCONT=1,"IP","internet.beeline.ru").

На некоторых тарифах имя пользователя и пароль пустые, тогда файл для C380 и ядра 2.6.12-r6 будет выглядеть примерно так: Файл: /etc/wvdial.conf

```
Modem = /dev/ttyACM0
Baud = 460800
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0
Init3 = AT+CGDCONT=1,"IP","internet.usi.ru"
ISDN = 0
Modem Type = Analog Modem
Phone = *99***1#
Username = ""
Password = ""
```

## Установка соединения

запускаем wvdial

### wvdial

Потираем ручки и юзаем GPRS на Linux

## Автоматическая установка соединения

Если вы используете udev, можно настроить автоматическое подключение GPRS при подключении телефона к USB.

Для этого создайте такой вот файл: Файл: /etc/udev/rules.d/90-gprs-autodial.rules

```
KERNEL=="ttyACM0", RUN="/usr/bin/wvdial"
```

(немного некрасиво, но работает на ура)

..а при отключении телефона соединение оборвётся и без наших скриптов..

**Примечание:** А вообще, на домашних компьютерах можно использовать и графическую звонилку krrr, там и в командной строке набирать ничего не надо, да и удобнее она чем wvdial.

## Заключение

Данная статья ни в коем случае не претендует на то что это является правильным и единственным способом подключения GPRS. Вся ответственность за действия ложится на ваши плечи. Я описал то что у меня работает. Допускаю что мог в чем то ошибиться так как писал это все на память так что милости прошу ваши комментарии и исправления можно отправлять по email [darkman@rusgate.org](mailto:darkman@rusgate.org)

### *Экономим трафик, с помощью Toonel. (Linux)*

**Ссылка на оригинал:** <http://doc.3dn.ru/publ/9-1-0-12>

**Автор:** Дмитрий а.к.а. doc (<http://doc.3dn.ru>)

**С версии:** 1.3

Бесплатный сервис toonel.net. Клиент написан на java, поэтому необходима виртуальная машина java.

Оф.сайт: [www.toonel.net](http://www.toonel.net)

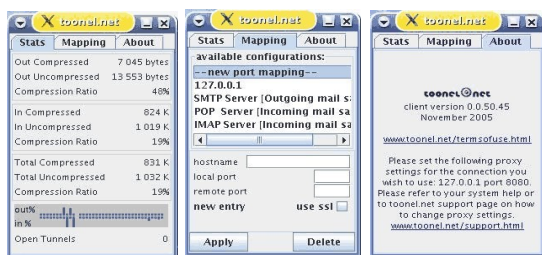
Убедитесь, что у вас установлена виртуальная машина JAVA. Посмотрите в /usr/lib/jre-xxxxx (x – номер версии), если не стоит то установите с диска дистрибутива (практически во всех дистрибутивах JAVA клиент есть), обычно называется j2re-xxxxx (x – номер версии), если Java стоит, то можно приступить к установке toonel.

Прежде чем запускать файл toonel, поменяйте у него права на чтение и запись.

Теперь выполняем простую команду:

**\$ java -jar /указываем путь/toonel.jar**

Появляется окно программы.



Вкладка **Stats** – показывает статистику,  
вкладка **Mapping** – настройка программы,  
вкладка **About** – о программе.

Теперь переходим к настройке, вкладка Mapping.

Чтобы настроить сжатие в браузере жмем на 127.0.0.1

Здесь стоит hostname: 127.0.0.1

localport: 8080

web interface port: 7999

Устанавливаем нужный процент сжатия JPEG.

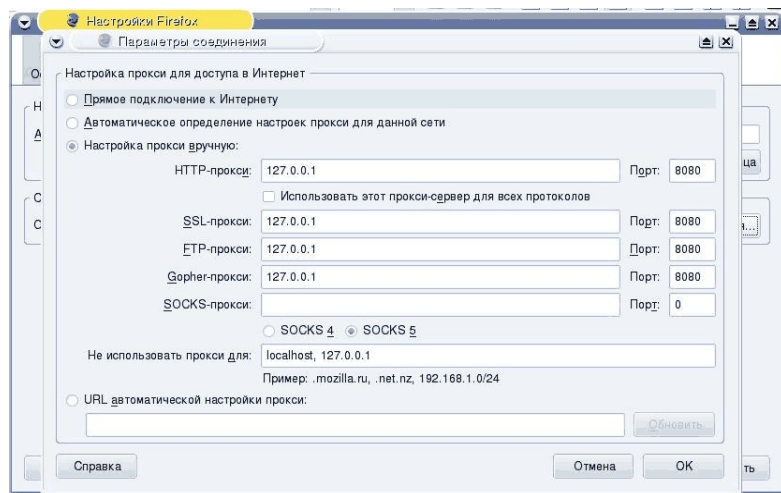
Теперь переходим в браузер, входим в Настройки, Параметры соединения.

Выбираем: Настройка прокси в ручную:

HTTP-прокси: 127.0.0.1 Порт:8080

SSL-прокси: 127.0.0.1 Порт:8080

FTP-прокси: 127.0.0.1 Порт:8080



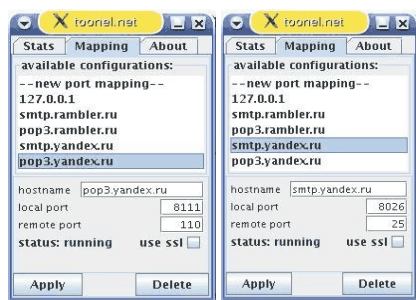
Gopher-прокси:127.0.0.1 Порт:8080

Все теперь сжатие трафика с проводника установлено.

### Сжимаем трафик электронной почты.

В Toonel на вкладке Mapping жмем на – new port mapping – в hostname пишем pop3 вашего почтового сервера, например pop3.yandex.ru , в local port указываем порт для прокси сервера, например 8111 и в remote port указываем порт для входящей почты обычно 110, и жмем кнопку Apply.

Для исходящей почты тоже самое только меняем порты, ну и конечножи сервер на smtp.



Все toonel для сжатия почты готов, теперь переходим в почтовый клиент.

Входим в Настройки, Параметры соединения, выбираем Настройка прокси вручную:

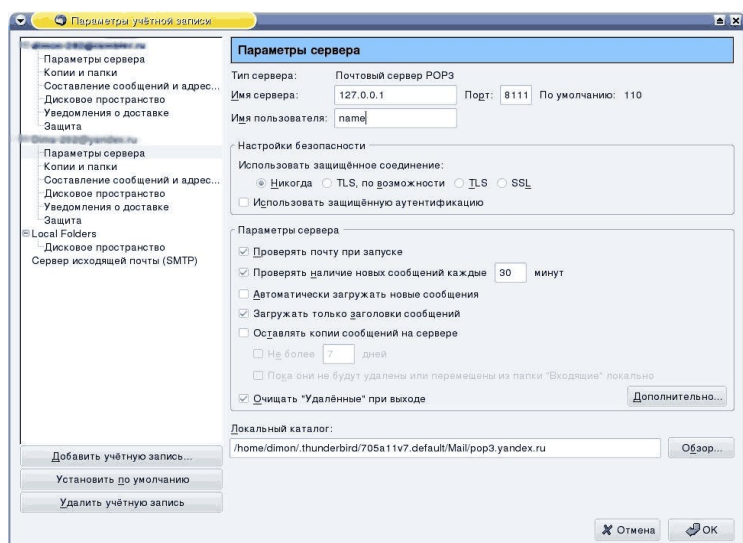
HTTP-прокси: 127.0.0.1 Порт:8080

SSL-прокси: 127.0.0.1 Порт:8080

После настройки прокси, входим в Параметры учетной записи и меняем Имя сервера: на 127.0.0.1 а порт 110, на порт который указали в toonel 8111.

Теперь и сжатие трафика входящей почты установлено.

Для исходящей почты Имя сервера: на 127.0.0.1 а порт 25, на порт который указали в toonel 8026.



Таким образом можно настроить сжатие любой программы которой вы пользуетесь выходя во всемирную паутину.

## **GPRS через KPPP**

**Ссылка на оригинал:**

[http://www.linuxcenter.ru/lib/articles/networking/kppp\\_gprs\\_hints.phtml](http://www.linuxcenter.ru/lib/articles/networking/kppp_gprs_hints.phtml)

**Автор:** Валентин Синицин

**С версии:** 1.3

*Телефон с поддержкой GPRS сегодня есть практически у каждого. Руководств по настройке GPRS-соединения в Linux тоже предостаточно, но большая часть из них ориентирована на конкретных операторов сотовой связи или не выходит за рамки mini-HOWTO. В этой статье мы постарались дать общее представление о том, что представляет собой "GPRS в Linux" и привели примерную схему действий для настройки подключения "стандартными средствами" — через KPPP*

С точки зрения повседневного использования сотовый телефон с поддержкой GPRS не слишком отличается от старого доброго модема на 14400 (да, и по скорости временами тоже). Значит, для подключения к "мобильному Интернету" можно использовать все привычные PPP-утилиты — ничего нового не потребуется.

Впрочем, необходимо учесть кое-какие тонкости — о них мы и собираемся поговорить.

Как и всякое приличное "железо", GPRS-модем идентифицируется в системе файлом устройства в `/dev`. Его конкретное имя может отличаться в зависимости от дистрибутива и способа подключения: для дата-кабеля оно может называться `/dev/ttyUSB0` (необходимо подгрузить драйвер используемого конвертера USB-to-Serial, чем, по сути, и является дата-кабель. Весьма распространенный вариант — `pl2303`, если в вашем дата-кабеле используется другая микросхема — ознакомьтесь с соответствующей секцией конфигуратора ядра Linux или поищите на форумах. В современных дистрибутивах с `udev`, как правило, необходим только скомпилированный модуль — все остальное, в том числе, создание файла устройства происходит автоматически), для IrDA — `/dev/ircomm0`, для Bluetooth — `/dev/rfcomm0`. В двух последних случаях потребует некоторая предварительная настройка, выходящая за рамки данной статьи — все подробности можно найти в соответствующих HOWTO.

Настройку GPRS-подключения, как уже упоминалось выше, можно произвести стандартными утилитами. Конкретные параметры можно найти на сайте вашего оператора сотовой связи, здесь же мы опишем стандартную схему действий на примере KPPP.

- Запустите KPPP и создайте новый модем (Configure->Modems->New). Укажите используемое имя устройства (если нужного вам имени в списке нет, создайте соответствующую символическую ссылку. Лучше делать это прямо из правил `udev`)
- Перейдите на вкладку "Modem" и нажмите кнопку "Modem commands". Введите дополнительную строку инициализации вида `AT +CGDCONT=1,"IP","APN_NAME"`. Точный вид этой строки должен быть приведен на сайте вашего оператора.
- Проверьте модем, нажав на кнопку Query Modem. Если вы не увидите сообщения об ошибке, значит, все прошло, как надо.
- Создайте новое подключение. В качестве телефона введите `*99***1#` — точный вариант, опять же, ищите на сайте оператора.
- Теперь можно задать дополнительные аргументы `pppd`. Рекомендуем добавить `"debug"` — будете хорошо видеть, если что-то пойдет не так. Если оператор требует отключить сжатие данных — введите `"связку" nodeflate + nobsdcomp`; если необходимо избавиться от сжатия заголовков TCP/IP — добавьте `nojob + novjccomp`.
- Отдельного упоминания заслуживает фраза: "имя пользователя и пароль не используются". Дело в том, что KPPP не позволяет оставить имя пользователя и пароль пустыми, если используется PAP-авторизация, а некоторые операторы не принимают никакие реквизиты, кроме пустой строки. Проблема решается следующим образом: откройте файл `/etc/ppp/options` (от имени `root`) и добавьте в него опцию `privgroup <ваша_группа>`. Это позволит вам передавать в командной строке привилегированные опции `pppd`, в частности, `noauth`. Ею и следует воспользоваться.
- Закройте окно настроек. Введите имя пользователя и пароль (в случае, если они "не используются", ими может быть любой набор символов) и нажмите кнопку "Connect". Если все пройдет гладко, то через некоторое время KPPP свернется в трей — это значит, что подключение удалось. В противном случае проверьте файл журнала на предмет сообщений об ошибках.



## ***GPRS через Bluetooth***

Смотрите "[\*\*HOWTO Соединение с Internet !\*\*](#)"

# HOWTO Соединение с Internet

Ссылка на оригинал: <http://ru.gentoo-wiki.com/>

С версии: 1.0

Изменено: 1.3

## Настройка VPN

Это работает для ядер  $\geq 2.6.15$

## Проверка параметров ядра

Если ваш провайдер использует mppe-mprc проверьте следующее:

Linux Kernel Configuration: Параметры ядра

```
Device Drivers --->
  Network support --->
    <M>   PPP (point-to-point protocol) support
    <M>   PPP MPPE compression (encryption) (EXPERIMENTAL)
  Cryptographic options --->
<M>   SHA1 and RC4 algorithms
```

Теперь необходимо собрать наше ядро. Без разницы, как вы скомпилировали эти части. Вы можете вкомпилировать все в ядро или собрать модулями. В конце вы должны подгрузить модули:

- **libcrc32c**
- **ip\_gre**
- **crc\_ccitt**
- **ppp\_async**
- и конечно **ppp\_mppe\_mprc**.

А затем запустите **modules-config**.

## Установка пакетов

Это работает для ядер  $\geq 2.6.15$

Установка очень проста:

Code: Установка

```
emerge sys-kernel/gentoo-dev-sources
emerge ppp
для ppp должны быть включены флаги ram,atp,dhcrp (не достоверно, у меня
работает без atp и dhcrp)
emerge pptpclient
```

## Настройка клиента

Если все удалось, то теперь вам необходимо сконфигурировать клиента. Вот пример конфигурации `/etc/ppp/peers/myvpn`:

```
pty                "/usr/sbin/pptp xxx.xxx.xxx.xxx --nolaunchpppd" #xxx...это адрес
сервера
name               myname
remotename         myvpn
lock
dump
noipx
#debug            #включите эту опцию, если хотите видеть отладочные сообщения (пригодиться
если ничего не работает)
ipcp-accept-local
ipcp-accept-remote
lcp-echo-failure 4
lcp-echo-interval 30
asynmap 0
crtscts
require-mppe
refuse-eap
nobsdcomp
nodeflate
proxyarp
ipparam myvpn
```

Осталось только дозвониться до провайдера: `pon myvpn` Для вывода отладочных сообщений на этой же консоли введите: `pon myvpn debug dump logfd 2`

Теперь необходимо настроить таблицы маршрутизации. Обычно это делается так:  
`route add default dev ppp0`

## Настройка с помощью pptpconfig

Для графической (GTK+) настройки PPTP используйте

- `net-dialup/pptpconfig`

он пока под маской "`~x86`"

Code: установка pptpconfig

`ACCEPT_KEYWORDS=~x86 emerge pptpconfig`

Или более правильно в файл `/etc/portage/package.keywords` добавить следующие строки:

```
net-dialup/pptpconfig ~x86
```

```
dev-php4/php-gtk ~x86
```

А затем

Code: установка pptpconfig

`emerge pptpconfig`

**Примечание:** Для нормальной сборки необходимы флаги USE=cli pcntl pcre posix session

Наберите от рута, например в xterm: pptpconfig &

Создайте новое соединение и настройте в соответствии с параметрами указанными выше, плюс поправка на рекомендации провайдера.

Для настройки маршрутизации по умолчанию используется вкладка **Routing**, на ней используйте значение **All to tunnel**.

Выберите созданное соединение и нажмите Start. Все :)

**Примечание:** если возникнут проблемы с соединением на вкладке "Miscellanyous" в опциях для rppd напишите

```
noipdefault 50
```

## Проверяем соединение

Теперь все готово. Можно навестить [Gentoo-Wiki](#) ;)

Проверить соединение можно командой:

```
# ifconfig ppp0
ppp0  Link encap:Point-to-Point Protocol
       inet addr:xxx.xxx.xxx.xxx  P-t-P:xxx.xxx.xxx.xxx  Mask:xxx.xxx.xxx.xxx
       UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
       RX packets:6 errors:0 dropped:0 overruns:0 frame:0
       TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:3
       RX bytes:72 (72.0 b)  TX bytes:78 (78.0 b)
```

[Настройка VPN-соединения](#) для Gentoo

[Настройка маршрутизации](#) для VPN-соединений

**Примечание (JohnBat26):** Я использую perl скрипт: **pptp-command** для работы с VPN. Также можно воспользоваться **Webmin**. Взято с сайта провайдера [2Com](#):

/usr/sbin/pptp-command (появится после установки pptp-client)

Надо запустить этот скрипт из под пользователя root.

Затем выбрать:

Setup -> Manage CHAP secrets -> Add a New CHAP secret

Local Name: ВАШ\_ЛОГИН

Remote Name [PPTP]: 2Com

Password: ВАШ\_ПАРОЛЬ

Затем выбрать Quit

Затем выбрать:

Add a NEW PPTP Tunnel -> Other

Tunnel Name: 2Com

```
Server IP:      192.168.X.2    (здесь X - номер вашей сети)
route:          add default ppp0
route:
Local Name:     ВАШ_ЛОГИН
Remote Name [PPTP]: 2Com
```

Затем выбрать Quit

**ВНИМАНИЕ!!!!**

проверьте наличие файла /etc/ppp/options.pptp

Файл /etc/ppp/options.pptp должен выглядеть след. образом:

```
lock
noauth
nobsdcomp
nodeflate
novj
novjccomp
nорсomp
noaccomp
```

Для установки VPN соединения надо запустить скрипт /usr/sbin/pptp-command

Затем выбрать:

```
start -> 2Com
```

Для разрыва VPN соединения надо запустить скрипт /usr/sbin/pptp-command

Затем выбрать:

```
stop
```

## Через pppoe

Code: Устанавливаем pppoe

**emerge pppoe**

В /etc/conf.d/net прописываем

```
modules=( "ifconfig" )
config_eth0=( "null" )
config_ppp0=( "ppp" )
link_ppp0="eth0"
plugins_ppp0=("pppoe")
username_ppp0='login'
password_ppp0='passwd'
pppd_ppp0=(
    "defaultroute" # Make this PPP interface the default route
    "usepeerdns"   # Use the DNS settings provided by PPP
    "lcp-echo-interval 15" # Send a LCP echo every 15 seconds
    "lcp-echo-failure 3"   # Make peer dead after 3 consecutive
)
```

```
#Если собираемся раздавать интернет в локалку с помощью маскардинга то нужно
добавить такую строчку в IPTABLES:
#iptables -I FORWARD 1 -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-
pmtu
```

Code: Делаем симлинк

```
ln -s /etc/init.d/net.lo /etc/init.d/net.ppp0
```

Code: Запускаем

```
/etc/inet.d/net.ppp0 start
```

## ***Настройка выхода через DIAL-UP***

### **Подключение модема**

#### ***Последовательный порт***

Подключение данного типа модема заключается в физическом "втыкании" в один из COM портов дата кабеля модема, включении модема в розетку и подсоединения к нему телефонной линии. Все это можно прочитать в руководстве к модему. Как известно последовательные порты в Линукс именуются */dev/ttySx* (где x - номер порта, /dev/ttyS0 = COM1 и т.д.) Вам нужно будет знать имя порта на который Вы "повесили свой модем"...скорее всего у вас их два или даже один, поэтому ошибиться здесь довольно сложно. Либо /dev/ttyS0, либо /dev/ttyS1. На этом подключение модема к последовательному порту можно считать законченным.

#### ***Подключение через USB***

Linux Kernel Configuration: Установка пакетов

```
"USB support" >>
<*>"USB Modem (CDC ACM) support"
```

устройство будет называться /dev/ttyACM0

#### ***Win модемы***

Для работы Win-модема нужно установить драйвер. Не все win-модемы поддерживаются поэтому Вам остается надеяться на лучшее. За драйверами идем на <http://www.linmodems.org/> Там можно взять отличную софтинку под названием scanModem которая опросит Ваш модем и выдаст (или не выдаст) URL откуда взять драйвер. Тянем. Распаковываем. Читаем INSTALL. Ставим. Еще можно почитать эту статью: [linmodems](http://www.linmodems.org/)

#### ***Для всех типов модемов***

Для удобства можно создать симлинк вида */dev/modem* на соответствующее модему устройство:

```
ln -s /dev/ttyS0 /dev/modem
```

и в дальнейшем в настройках использовать устройство ( [/dev/modem](#) )...хотя никто и не запрещает использовать устройство модема без симлинка [/dev/ttySx](#)

### **Проверка работоспособности, возможные грабли**

После подключения модема и установки всех необходимых драйверов необходимо проверить его работоспособность. Я делаю это так: запускаю утилиту minicom и даю команду AT. Если все в порядке, я в ответ получаю OK, и не получаю если что-то не так. Если OK не получен, то самое первое, что нужно сделать - это проверить настройки minicom. Если модем ответил "OK", можно попробовать позвонить провайдеру:

ATDP 554554 - для пульсового набора  
или  
ATDT 554554 - для тонового набора

если модемы соединились и на экране появилось приглашение ввести логин или посыпался мусор в виде фигурных скобок ({}), можете себя поздравить и переходить к настройке программного обеспечения, модем у вас работает.

## **Настройка программного обеспечения**

### **Включение поддержки PPP в ядре**

в ядре должны быть включены опции отвечающие за ppp:

Linux Kernel Configuration: Установка пакетов

```
Device Drivers----->
Networking support----->
  <*>PPP (point-to-point protocol) support
  <*>PPP support for async serial ports
  <*>PPP Deflate compression
  <*>PPP BSD compression
```

### **Установка необходимых пакетов**

**во-первых** нужно установить пакет который реализует поддержку протокола точка-точка (point-to-point protocol)

```
emerge ppp
```

**во-вторых** нужна звонилка или скрипт для автоматизирования процесса установления соединения.

*Обзор звонилок и примеры скриптов смотри ниже...*

### **Настройка соединения с помощью chat-script**

Прежде чем писать скрипт, необходимо настроить pppd: в [/etc/ppp/pap-secrets](#) пропишем свои Логин/Пароль

Code: [/etc/ppp/pap-secrets](#)

```
login * password *
```

где login и password выданные Вашим провайдером Вам логин и пароль соответственно...

в `/etc/ppp/options` пропишем:

```
Code: /etc/ppp/options
```

```
nodetach
defaultroute
lock
crtscts
```

в `/etc/resolv.conf` прописываем один или несколько ip-адресов DNS серверов (спросить у провайдера адрес его сервера)

```
Code: /etc/resolv.conf
```

```
nameserver xx.xx.xx.xx
```

xx.xx.xx.xx - заменить на ip-адресс DNS сервера

В каталоге `/usr/share/doc/ppp-*/scripts` лежат примеры скриптов, нам нужны:

```
ppp-on
ppp-off
ppp-on-dialer
```

подправляем в них логин, номер телефона, пароль...и вуаля...ppp-on поднимает соединение, ppp-off разрывает его, а ppp-on-dialer -вспомогательный скрипт для осуществления дозвона.

замечу только что для тонового набора используется AT-команда ATDT, а для пульсового - ATDP.

## Обзор звонилки

Хорошая звонилка избавит Вас от процесса написания скриптов, вникания в опции rpprd, и сделает настройку соединения довольно тривиальной

### chestnut-dialer

Замечательная звонилка которая к сожалению не поставляется в стандартных портах gentoo. ebuild можно взять [здесь](#)

Из плюсов

- Настройка тривиальна
- Умеет работать в консоли
- Умеет работать в GUI (имеет GTK+ и Qt интерфейсы)
- Умеет call back
- Нетребавательна к ресурсам
- "Местного разлива" - можно пообщаться с автором на родном языке, что для многих немаловажно.

Из минусов

- Не входит в официальное дерево портов



## Настройка Call back

pppd должен быть собран с поддержкой CALLBACK. В gentoo это делается по умолчанию, так что вам нужно просто собрать pppd штатными средствами.

Самый простой скрипт для CALLBACK был обнаружен в `/usr/doc/ppp-2.4.2/README.cbcr`:

```
#!/bin/sh
pppd debug nodetach /dev/modem 115200 crtscts modem \
callback СВОЙ_НОМЕР name ЛОГИН \
connect 'chat -v "" atz OK atdtНОМЕР_ПРОВАЙДЕРА CONNECT "~--"'
sleep 1
pppd debug nodetach /dev/modem 115200 crtscts modem \
name ЛОГИН defaultroute \
connect 'chat -v RING ATA CONNECT "\c"'
```

ЛОГИН должен соответствовать записи в `/etc/ppp/chap-secrets` (`/etc/ppp/pap-secrets`). В `/etc/resolv.conf` нужно прописать адрес DNS-сервера (можно добавить опцию `usepeerdns` в `/etc/ppp/options` для получения списка DNS серверов от провайдера).

Также можно использовать `kppp` или `chestnut-dialer` для установки callback соединения. Настройки их достаточно тривиальны.

Самым сложным может оказаться правильно подобрать строку инициализации для модема. Рекомендуют отключить `CallerID`. Приведу несколько строк инициализации для примера, но лучше берите документацию от модема (часто идет только на сидироме хотя иногда бывает и в бумажном виде), смотрите что каждая из AT команд означает и экспериментируйте.

- Для модемов Lucent:

```
ats0=1s38=0s37=17-v90=0
ats0=1s38=0s37=17-v90=0+vcid=0
```

значение регистра `s37` можна понижать до 14

- Для модемов GVC Vector K2D:

```
ATS0=1+VCID=0&U1
```

Я лично подобрав подходящую строку инициализации, сохранил ее прямо в профиле модема...

## Настройка выхода через GPRS

Если вы подключаетесь через GPRS с помощью звонилки (`kppp`, `chestnut-dialer`), может быть важным использование опции `pppd - noipdefault` !!!

## Требуемая аппаратура

Для соединения по GPRS потребуется следующее:

- Мобильный телефон, поддерживающий технологию GPRS
- Кабель COM или USB для подключения к компьютеру ("data-кабель") либо IrDA-интерфейс.

## Подключение с помощью USB-кабеля

Убедитесь, что в конфигурационном файле Вашего ядра присутствуют следующие опции:

```
"Device Drivers" >>
[*]"Networking support" >>
  [*] "Network device support"
  <M> "PPP (point-to-point protocol) support"
  <M> "PPP support for async serial ports"
  <M> "PPP support for sync tty ports"
  <M> "PPP Deflate compression"
  <M> "PPP BSD-Compress compression"
```

Для data-кабелей на основе микросхем Profilic 2303 (используются для подключения большинства телефонов, в том числе Siemens, Motorola, Nokia и др.) проверьте наличие:

```
"USB support" >>
  "USB Serial Converter support" >>
    <M> "USB Prolific 2303 Single Port Serial Driver"
```

После компиляции ядра проверьте модуль:

```
modprobe pl2303
```

Теперь кабель представлен в системе как устройство /dev/usb/lps/0 (возможно /dev/ttyUSB0), которое может быть использовано в качестве обычного модема.

Для телефонов Motorola, которые подключаются кабелем mini-USB (например C380,C390) достаточно включить

```
"USB support" >>
  <*>"USB Modem (CDC ACM) support"
```

устройство будет называться /dev/ttyACM0

## Подключение с помощью COM-кабеля

Тут все просто, "втыкаем" и используем...как любой модем на последовательном порту.

## Подключение через Bluetooth

Предполагается что вы уже настроили соединение телефона с компьютером через bluetooth,например по этой статье:[Работа с мобильными телефонами через bluetooth](#).Итак устройства сопряжены.Теперь нам необходимо подключить телефон как модем.Для этого мы выполним привязку bluetooth-сервиса 1(то есть Dial-Up) к устройству /dev/rfcomm0.Начнем по порядку.

- Поищем телефон

```
#hcitool scan
Scanning ...
    00:60:57:D5:3B:BE          Gtbear
```

- Подключим его к rfcomm0

```
#rfcomm bind 0 00:60:57:D5:3B:BE 1
```

Этой командой мы подключили в rfcomm0 канал 1. Теперь телефон готов к работе.Осталось настроить соединения в конфигах rppd.

- Создадим файл /etc/ppp/peers/gprs.В него запишем следующее:

```

lcp-echo-failure 0
lcp-echo-interval 0
/dev/rfcomm0
connect /etc/ppp/peers/suct-connect # Это скрипт соединения, такое имя выбрано в
честь моего оператора:
115200 #Южно-Уральский Сотовый Телефон :)
crtscts
local
ipcp-accept-local
noauth
usepeerdns
user suct
defaultroute
noipdefault

```

- Теперь сделаем скрипт соединения:

```

#!/bin/bash
exec chat -vS \
'' \rAT \
TIMEOUT 12 \
OK ATH \
OK ATE1 \
OK 'AT+CGDCONT=1,"IP","internet.usi.ru"' \
OK ATD*99***1# \
TIMEOUT 22 \
SAY "\nWaiting for connect...\n" \
CONNECT "" \
SAY "\nGPRS connected.Enjoy.."

```

- Впринципе все готово.Для запуска соединения можно использовать команду **pppd call gprs nodetach**

## Подключение через IrDA

Пакеты:

```
emerge irda-utils
```

Linux Kernel Configuration: Ядро 2.6

```

Device Drivers --->
[*] Networking support
  <*> IrDA (infrared) subsystem support --->
  <*> IrCOMM protocol
  Infrared-port device drivers --->
    <*> IrDA USB dongles
    <*> SigmaTel STIr4200 bridge (EXPERIMENTAL)

```

Когда ядро будет готово можно запускать команду:

```
irattach irda0 -s
```

Теперь если поднести телефон к инфракрасному порту он должен определиться...  
Дальше можно поднимать rrr как для обычного модема

## Настройка выхода через CDMA (SKYLINK)

### Подключение аппаратуры

- [Модем AnyDATA ADU-E100A](#)

### Настройка программного обеспечения

- Устанавливаем пакет который реализует поддержку протокола точка-точка (point-to-point protocol)

```
# emerge ppp
```

- Редактируем файл `/etc/ppp/chap-secrets`

```
Code: /etc/ppp/chap-secrets
```

```
# Secrets for authentication using CHAP
# client server secret IPaddresses
"mobile" * "internet" *
```

- Создаем файл `/etc/ppp/peers/cdma`

```
Code: /etc/ppp/peers/cdma
```

```
debug
/dev/ttyUSB0
921600
noipdefault
defaultroute
ipcp-accept-local
lcp-echo-interval 60
lcp-echo-failure 5
usepeerdn
nocomp
noauth
noaccomp
nodetach
user "mobile"
connect "/usr/sbin/chat -s -S -V -t 5 -f /etc/ppp/cdma.chat"
```

- Создаем файл `/etc/ppp/cdma.chat`

```
Code: /etc/ppp/cdma.chat
```

```
" " # "Это не кавычки, а два знака, который на клавиатуре слева от единички
" 'ATZ' # "Это не кавычки, а два знака, который на клавиатуре слева от единички
'OK' 'ATI'
'OK' 'ATDT#777'
'CONNECT' 'ATO'
```

- Пробуем

```
# pon cdma
```

# Power Management Guide (Англ.)

*Ссылка на оригинал:*

<http://www.gentoo.org/doc/ru/kde-config.xml>

*С версии: 1.0.*

## 1. Introduction

Capacity and lifetime of laptop batteries have improved much in the last years. Nevertheless modern processors consume much more energy than older ones and each laptop generation introduces more devices hungry for energy. That's why Power Management is more important than ever. Increasing battery run time doesn't necessarily mean buying another battery. Much can be achieved applying intelligent Power Management policies.

### A Quick Overview

Please notice that this guide describes Power Management for laptops. While some sections might also suite for servers, others do not and may even cause harm. Please do not apply anything from this guide to a server unless you really know what you are doing.

As this guide has become rather long, here's a short overview helping you to find your way through it.

The Prerequisites chapter talks about some requirements that should be met before any of the following device individual sections will work. This includes BIOS settings, kernel configuration and some simplifications in user land. The following three chapters focus on devices that typically consume most energy - processor, display and hard drive. Each can be configured seperately. CPU Power Management shows how to adjust the processor's frequency to save a maximum of energy without losing too much performance. A few different tricks prevent your hard drive from working unnecessarily often in Disk Power Management (decreasing noise level as a nice side effect). Some notes on graphics cards, Wireless LAN and USB finish the device section in Power Management For Other Devices while another chapter is dedicated to the (rather experimental) sleep states. Last not least Troubleshooting lists common pitfalls.

### Power Budget For Each Component

Nearly every component can operate in different states - off, sleep, idle, active to name a few - consuming a different amount of energy. Major parts are consumed by the LCD display, CPU, chipset and hard drives. Often one is able to activate OS-independent Power Management in the BIOS, but an intelligent setup in the operating system adapting to different situations can achieve much more.

## 2. Prerequisites

Before discussing the details of making individual devices Power Management aware, make sure certain requirements are met. After controlling BIOS settings, some kernel options want to be enabled - these are in short ACPI, sleep states and CPU frequency scaling. As power saving most of the time comes along with performance loss or increased latency, it should only be enabled when running on batteries. That's where a new runlevel battery comes in handy.

### The BIOS Part

First have a look into your BIOS Power Management settings. The best way is to combine BIOS and operating system policies, but for the moment it's better to disable most of the BIOS part. This makes sure it doesn't interfere with your policies. Don't forget to re-check BIOS settings after you configured everything else.

### Setting USE Flags

Please check that the acpi USE flag is set in `/etc/make.conf`. Other USE flags that might be interesting for your system are `apm`, `lm_sensors`, `nforce2`, `nvidia`, `pmu`. See `/usr/portage/profiles/use*.desc` for details. If you forgot to set one of these flags, you can recompile affected packages using the `--newuse` flag in `emerge`, see `man emerge`.

### Configuring The Kernel

ACPI (Advanced Configuration and Power Interface) support in the kernel is still work in progress. Using a recent kernel will make sure you'll get the most out of it.

There are different kernel sources in Portage. I'd recommend using `gentoo-sources` or `suspend2-sources`. The latter contains patches for Software Suspend 2, see the chapter about sleep states for more details. When configuring the kernel, activate at least these options:

#### Code Listing 2.1: Minimum kernel setup for Power Management (Kernel 2.6)

Power Management Options --->

- ☒ Power Management Support
- ☐ Software Suspend

ACPI( Advanced Configuration and Power Interface ) Support --->

- ☒ ACPI Support
- ☐ Sleep States
- ☐ /proc/acpi/sleep (deprecated)
- ☒ AC Adapter
- ☒ Battery
- <M> Button
- <M> Video
- ☐ Generic Hotkey
- <M> Fan
- <M> Processor
- <M> Thermal Zone
- < > ASUS/Medion Laptop Extras
- < > IBM ThinkPad Laptop Extras
- < > Toshiba Laptop Extras
- (0) Disable ACPI for systems before Jan 1st this year
- ☐ Debug Statements

```
[*] Power Management Timer Support
< > ACPI0004,PNP0A05 and PNP0A06 Container Driver (EXPERIMENTAL)
```

```
CPU Frequency Scaling --->
[*] CPU Frequency scaling
[] Enable CPUfreq debugging
< > CPU frequency translation statistics
[] CPU frequency translation statistics details
    Default CPUFreq governor (userspace)
<*> 'performance' governor
<*> 'powersave' governor
<*> 'ondemand' cpufreq policy governor
<*> 'conservative' cpufreq governor
<*> CPU frequency table helpers
<M> ACPI Processor P-States driver
<*> CPUFreq driver for your processor
```

Decide yourself whether you want to enable Software Suspend, and Sleep States (see below). If you own an ASUS, Medion, IBM Thinkpad or Toshiba laptop, enable the appropriate section.

The kernel has to know how to enable CPU frequency scaling on your processor. As each type of CPU has a different interface, you've got to choose the right driver for your processor. Be careful here - enabling Intel Pentium 4 clock modulation on a Pentium M system will lead to strange results for example. Consult the kernel documentation if you're unsure which one to take.

Compile your kernel, make sure the right modules get loaded at startup and boot into your new ACPI-enabled kernel. Next run `emerge sys-power/acpid` to get the acpi daemon. This one informs you about events like switching from AC to battery or closing the lid. Make sure the modules are loaded if you didn't compile them into the kernel and start acpid by executing `/etc/init.d/acpid start`. Run `rc-update add acpid default` to load it on startup. You'll soon see how to use it.

#### **Code Listing 2.2:** Installing acpid

```
# emerge sys-power/acpid
# /etc/init.d/acpid start
# rc-update add acpid default
```

## **Creating A "battery" Runlevel**

The default policy will be to enable Power Management only when needed - running on batteries. To make the switch between AC and battery convenient, create a runlevel battery that holds all the scripts starting and stopping Power Management.

Note: You can safely skip this section if you don't like the idea of having another runlevel. However, skipping this step will make the rest a bit trickier to set up. The next sections assume a runlevel battery exists.

#### **Code Listing 2.3:** Creating a battery runlevel

```
# cd /etc/runlevels
# cp -a default battery
```

Finished. Your new runlevel battery contains everything like default, but there is no automatic switch between both yet. Time to change it.

## Reacting On ACPI Events

Typical ACPI events are closing the lid, changing the power source or pressing the sleep button. An important event is changing the power source, which should cause a runlevel switch. A small script will take care of it.

First you need a script which changes the runlevel to default respectively battery depending on the power source. The script uses the `on_ac_power` command from `sys-power/powermgmt-base` - make sure the package is installed on your system.

**Code Listing 2.4:** Installing `powermgmt-base`  
`# emerge powermgmt-base`

You are now able to determine the power source by executing `on_ac_power && echo AC available || echo Running on batteries` in a shell. The script below is responsible for changing runlevels. Save it as `/etc/acpi/actions/pmg_switch_runlevel.sh`.

**Code Listing 2.5:** `/etc/acpi/actions/pmg_switch_runlevel.sh`

```
#!/bin/bash

# BEGIN configuration
RUNLEVEL_AC="default"
RUNLEVEL_BATTERY="battery"
# END configuration

if [ ! -d "/etc/runlevels/${RUNLEVEL_AC}" ]
then
    logger "${0}: Runlevel ${RUNLEVEL_AC} does not exist. Aborting."
    exit 1
fi

if [ ! -d "/etc/runlevels/${RUNLEVEL_BATTERY}" ]
then
    logger "${0}: Runlevel ${RUNLEVEL_BATTERY} does not exist. Aborting."
    exit 1
fi

if on_ac_power
then
    if [[ "$(</var/lib/init.d/softlevel)" != "${RUNLEVEL_AC}" ]]
    then
        logger "Switching to ${RUNLEVEL_AC} runlevel"
        /sbin/rc ${RUNLEVEL_AC}
    fi
elif [[ "$(</var/lib/init.d/softlevel)" != "${RUNLEVEL_BATTERY}" ]]
then
    logger "Switching to ${RUNLEVEL_BATTERY} runlevel"
    /sbin/rc ${RUNLEVEL_BATTERY}
fi
```

Dont forget to run `chmod +x /etc/acpi/actions/pmg_switch_runlevel.sh` to make the script executable. The last thing that needs to be done is calling the script whenever the power source changes. That's done by catching ACPI events with the help of `acpid`. First you



need to know which events are generated when the power source changes. The events are called `ac_adapter` and `battery` on most laptops, but it might be different on yours.

**Code Listing 2.6:** Determining ACPI events for changing the power source  
`# tail -f /var/log/acpid | grep "received event"`

Run the command above and pull the power cable. You should see something like this:

**Code Listing 2.7:** Sample output for power source changes  
[Tue Sep 20 17:39:06 2005] received event "ac\_adapter AC 00000080 00000000"  
[Tue Sep 20 17:39:06 2005] received event "battery BAT0 00000080 00000001"

The interesting part is the quoted string after received event. It will be matched by the event line in the files you are going to create below. Don't worry if your system generates multiple events or always the same. As long as any event is generated, runlevel changing will work.

**Code Listing 2.8:** `/etc/acpi/events/pmg_ac_adapter`  
# replace "ac\_adapter" below with the event generated on your laptop  
# For example, `ac_adapter.*` will match `ac_adapter AC 00000080 00000000`  
`event=ac_adapter.*`  
`action=/etc/acpi/actions/pmg_switch_runlevel.sh %e`

**Code Listing 2.9:** `/etc/acpi/events/pmg_battery`  
# replace "battery" below with the event generated on your laptop  
# For example, `battery.*` will match `battery BAT0 00000080 00000001`  
`event=battery.*`  
`action=/etc/acpi/actions/pmg_switch_runlevel.sh %e`

Finally `acpid` has to be restarted to recognize the changes.

**Code Listing 2.10:** Finishing runlevel switching with `acpid`  
`# /etc/init.d/acpid restart`

Give it a try: Plug AC in and out and watch `syslog` for the "Switching to AC mode" or "Switching to battery mode" messages. See the Troubleshooting section if the script is not able to detect the power source correctly.

Due to the nature of the event mechanism, your laptop will boot into runlevel default regardless of the AC/battery state. This is fine when running from AC, but we'd like to boot into the battery runlevel otherwise. One solution would be to add another entry to the boot loader with the parameter `softlevel=battery`, but it's likely to forget choosing it. A better way is faking an ACPI event in the end of the boot process and letting `pmg_switch_runlevel.sh` script decide whether a runlevel change is necessary. Open `/etc/conf.d/local.start` in your favourite editor and add these lines:

**Code Listing 2.11:** Runlevel adjustment at boot time by editing `local.start`  
# Fake acpi event to switch runlevel if running on batteries  
`/etc/acpi/actions/pmg_switch_runlevel.sh "battery/battery"`

Prepared like this you can activate Power Management policies for individual devices.

### 3. CPU Power Management

Mobile processors can operate at different frequencies. Some allow changing voltage as well. Most of the time your CPU doesn't need to run at full speed and scaling it down will save much energy - often without any performance decrease.

#### Some Technical Terms

CPU frequency scaling brings up some technical terms that might be unknown to you. Here's a quick introduction.

First of all, the kernel has to be able to change the processor's frequency. The CPUfreq processor driver knows the commands to do it on your CPU. Thus it's important to choose the right one in your kernel. You should already have done it above. Once the kernel knows how to change frequencies, it has to know which frequency it should set. This is done according to the policy which consists of a CPUfreq policy and a governor. A CPUfreq policy are just two numbers which define a range the frequency has to stay between - minimal and maximal frequency. The governor now decides which of the available frequencies in between minimal and maximal frequency to choose. For example, the powersave governor always chooses the lowest frequency available, the performance governor the highest one. The userspace governor makes no decision but chooses whatever the user (or a program in userspace) wants - which means it reads the frequency from `/sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed`.

This doesn't sound like dynamic frequency changes yet and in fact it isn't. Dynamics however can be accomplished with various approaches. For example, the ondemand governor makes its decisions depending on the current CPU load. The same is done by various userland tools like cpudyn, cpufreqd, powernowd and many more. ACPI events can be used to enable or disable dynamic frequency changes depending on power source.

#### Setting The Frequency Manually

Decreasing CPU speed and voltage has two advantages: On the one hand less energy is consumed, on the other hand there is thermal improvement as your system doesn't get as hot as running on full speed. The main disadvantage is obviously the loss of performance. Decreasing processor speed is a trade off between performance loss and energy saving.

Note: Not every laptop supports frequency scaling. If unsure, have a look at the list of supported processors in the Troubleshooting section to verify yours is supported.

It's time to test whether CPU frequency changing works. Let's install another tool which is very handy for debugging purposes: `sys-power/cpufrequtils`

**Code Listing 3.1:** Checking CPU frequency

```
# emerge cpufrequtils
# cpufreq-info
```

Here is an example output:

### Code Listing 3.2: Sample output from cpufreq-info

```
cpufrequtils 0.3: cpufreq-info (C) Dominik Brodowski 2004
Report errors and bugs to linux@brodo.de, please.
analyzing CPU 0:
  driver: centrino
  CPUs which need to switch frequency at the same time: 0
  hardware limits: 600 MHz - 1.40 GHz
  available frequency steps: 600 MHz, 800 MHz, 1000 MHz, 1.20 GHz, 1.40 GHz
  available cpufreq governors: conservative, ondemand, powersave, userspace, performance
  current policy: frequency should be within 924 MHz and 1.40 GHz.
    The governor "performance" may decide which speed to use
    within this range.
  current CPU frequency is 1.40 GHz.
```

Now play around with `cpufreq-set` to make sure frequency switching works. Run `cpufreq-set -g ondemand` for example to activate the ondemand governor and verify the change with `cpufreq-info`. If it doesn't work as expected, you might find help in the Troubleshooting section in the end of this guide.

While adjusting the frequency to the current load looks simple at a first glance, it's not such a trivial task. A bad algorithm can cause switching between two frequencies all the time or wasting energy when setting frequency to an unnecessary high level.

Which one to choose? If you have no idea about it, try `cpufreqd`:

### Code Listing 3.3: Installing cpufreqd # emerge cpufreqd

`cpufreqd` can be configured by editing `/etc/cpufreqd.conf`. The default one that ships with `cpufreqd` may look a bit confusing. I recommend replacing it with the one from former Gentoo developer Henrik Brix Andersen (see below). Please notice that you need `cpufreqd-2.0.0` or later. Earlier versions have a different syntax for the config file.

### Code Listing 3.4: /etc/cpufreqd.conf (cpufreqd-2.0.0 and later)

```
[General]
pidfile=/var/run/cpufreqd.pid
poll_interval=3
enable_plugins=acpi_ac, acpi_battery
enable_remote=1
remote_group=wheel
verbosity=5
[/General]
```

```
[Profile]
name=ondemand
minfreq=0%
maxfreq=100%
policy=ondemand
[/Profile]
```

```
[Profile]
name=conservative
minfreq=0%
maxfreq=100%
policy=conservative
[/Profile]
```

```
[Profile]
```

```

name=powersave
minfreq=0%
maxfreq=100%
policy=powersave
[/Profile]

[Profile]
name=performance
minfreq=0%
maxfreq=100%
policy=performance
[/Profile]

[Rule]
name=battery
ac=off
profile=conservative
[/Rule]

[Rule]
name=battery_low
ac=off
battery_interval=0-10
profile=powersave
[/Rule]

[Rule]
name=ac
ac=on
profile=ondemand
[/Rule]

```

Now you can start the `cpufreqd` daemon. Add it to the default and battery runlevel as well.

**Code Listing 3.5:** Starting `cpufreqd`

```

# rc-update add cpufreqd default battery
# rc

```

Sometimes it can be desirable to select another policy than the daemon chooses, for example when battery power is low, but you know that AC will be available soon. In that case you can turn on `cpufreqd`'s manual mode with `cpufreqd-set manual` and select one of your configured policies (as listed by `cpufreqd-get`). You can leave manual mode by executing `cpufreqd-set dynamic`.

Warning: Do not run more than one of the above programs at the same time. It may cause confusion like switching between two frequencies all the time.

## Verifying the result

The last thing to check is that your new policies do a good job. An easy way to do so is monitoring CPU speed while working with your laptop:

**Code Listing 3.6:** Monitoring CPU speed

```

# watch grep \"cpu MHz\" /proc/cpuinfo

```

If `/proc/cpuinfo` doesn't get updated (see Troubleshooting), monitor the CPU frequency with:

**Code Listing 3.7:** Alternative CPU speed monitoring  
**# watch x86info -mhz**

Depending on your setup, CPU speed should increase on heavy load, decrease on no activity or just stay at the same level. When using `cpufreqd` and verbosity set to 5 or higher in `cpufreqd.conf` you'll get additional information about what's happening reported to `syslog`.

## 4. LCD Power Management

As you can see in figure 1.1, the LCD display consumes the biggest part of energy (might not be the case for non-mobile CPU's). Thus it's quite important not only to shut the display off when not needed, but also to reduce it's backlight if possible. Most laptops offer the possibility to control the backlight dimming.

### Standby settings

The first thing to check is the standby/suspend/off timings of the display. As this depends heavily on your windowmanager, I'll let you figure it out yourself. Just two common places: Blanking the terminal can be done with `setterm -blank <number-of-minutesM>`, `setterm -powersave on` and `setterm -powerdown <number-of-minutesM>`. For X.org, modify `/etc/X11/xorg.conf` similar to this:

**Code Listing 4.1:** LCD suspend settings in X.org and Xfree86

```
Section "ServerLayout"
    Identifier [...]
    [...]
    Option "BlankTime" "5" # Blank the screen after 5 minutes (Fake)
    Option "StandbyTime" "10" # Turn off screen after 10 minutes (DPMS)
    Option "SuspendTime" "20" # Full suspend after 20 minutes
    Option "OffTime" "30" # Turn off after half an hour
    [...]
EndSection

[...]

Section "Monitor"
    Identifier [...]
    Option "DPMS" "true"
    [...]
EndSection
```

This is the same for XFree86 and `/etc/X11/XF86Config`.

### Backlight dimming

Probably more important is the backlight dimming. If you have access to the dimming settings via a tool, write a small script that dims the backlight in battery mode and place it in your battery runlevel. The following script should work on most IBM Thinkpads and Toshiba laptops. You've got to enable the appropriate option in your kernel (IBM Thinkpads only). For Toshiba laptops, install `app-laptop/acpitool` and skip configuration of `ibm_acpi` as described below.

**Warning:** Support for setting brightness is marked experimental in ibm-acpi. It accesses hardware directly and may cause severe harm to your system. Please read the ibm-acpi website

To be able to set the brightness level, the ibm\_acpi module has to be loaded with the experimental parameter.

**Code Listing 4.2:** automatically loading the ibm\_acpi module

(Please read the warnings above before doing this!)

```
# echo "options ibm_acpi experimental=1" >> /etc/modules.d/ibm_acpi
# /sbin/modules-update
# echo ibm_acpi >> /etc/modules.autoload.d/kernel-2.6
# modprobe ibm_acpi
```

This should work without error messages and a file /proc/acpi/ibm/brightness should be created after loading the module. An init script will take care of choosing the brightness according to the power source.

**Code Listing 4.3:** /etc/conf.d/lcd-brightness

```
# See /proc/acpi/ibm/brightness for available values
# Please read /usr/src/linux/Documentation/ibm-acpi.txt

# brightness level in ac mode. Default is 7.
BRIGHTNESS_AC=7

# brightness level in battery mode. Default is 4.
BRIGHTNESS_BATTERY=4
```

**Code Listing 4.4:** /etc/init.d/lcd-brightness

```
#!/sbin/runscript

set_brightness() {
    if on_ac_power
    then
        LEVEL=${BRIGHTNESS_AC:-7}
    else
        LEVEL=${BRIGHTNESS_BATTERY:-4}
    fi

    if [ -f /proc/acpi/ibm/brightness ]
    then
        ebegin "Setting LCD brightness"
        echo "level ${LEVEL}" > /proc/acpi/ibm/brightness
        eend $?
    elif [[ -e /usr/bin/acpitool && -n $(acpitool -T | grep "LCD brightness") ]]
    then
        ebegin "Setting LCD brightness"
        acpitool -l $LEVEL >/dev/null || ewarn "Unable to set lcd brightness"
        eend $?
    else
        ewarn "Setting LCD brightness is not supported."
        ewarn "For IBM Thinkpads, check that ibm_acpi is loaded into the kernel"
        ewarn "For Toshiba laptops, you've got to install app-laptop/acpitool"
    fi
}

start() {
```

```

    set_brightness
}

stop () {
    set_brightness
}

```

When done, make sure brightness is adjusted automatically by adding it to the battery runlevel.

**Code Listing 4.5:** Enabling automatic brightness adjustment

```

# chmod +x /etc/init.d/lcd-brightness
# rc-update add lcd-brightness battery
# rc

```

## 5. Disk Power Management

Hard disks consume less energy in sleep mode. Therefore it makes sense to activate power saving features whenever the hard disk is not used for a certain amount of time. I'll show you two alternative possibilities to do it. First, laptop-mode will save most energy due to several measures which prevent or at least delay write accesses. The drawback is that due to the delayed write accesses a power outage or kernel crash will be more dangerous for data loss. If you don't like this, you have to make sure that there are no processes which write to your hard disk frequently. Afterwards you can enable power saving features of your hard disk with hdparm as the second alternative.

### Increasing idle time - laptop-mode

Recent kernels (2.6.6 and greater, recent 2.4 ones and others with patches) include the so-called laptop-mode. When activated, dirty buffers are written to disk on read calls or after 10 minutes (instead of 30 seconds). This minimizes the time the hard disk needs to be spun up.

**Code Listing 5.1:** Automated start of laptop-mode

```

# emerge laptop-mode-tools

```

laptop-mode-tools has its configuration file in /etc/laptop-mode/laptop-mode.conf. Adjust it the way you like it, it's well commented. Run rc-update add laptop\_mode battery to start it automatically.

Recent versions (1.11 and later) of laptop-mode-tools include a new tool lm-profiler. It will monitor your system's disk usage and running network services and suggests to disable unneeded ones. You can either disable them through laptop-mode-tools builtin runlevel support (which will be reverted by Gentoo's /sbin/rc) or use your default/battery runlevels (recommended).

**Code Listing 5.2:** Sample output from running lm-profiler

```

# lm-profiler

```

```

Profiling session started.
Time remaining: 600 seconds

```

```
[4296896.602000] amarokapp
Time remaining: 599 seconds
[4296897.714000] sort
[4296897.970000] mv
Time remaining: 598 seconds
Time remaining: 597 seconds
[4296900.482000] reiserfs/0
```

After profiling your system for ten minutes, Im-profiler will present a list of services which might have caused disk accesses during that time.

**Code Listing 5.3:** Im-profiler suggests to disable some services

```
Program:    "atd"
Reason:     standard recommendation (program may not be running)
Init script: /etc/init.d/atd (GUESSED)
```

**Do you want to disable this service in battery mode? [y/N]: n**

To disable atd as suggested in the example above, you would run `rc-update del atd battery`. Be careful not to disable services that are needed for your system to run properly - Im-profiler is likely to generate some false positives. Do not disable a service if you are unsure whether it's needed.

## Limiting Write Accesses

If you don't want to use laptop-mode, you must take special care to disable services that write to your disk frequently - `syslogd` is a good candidate, for example. You probably don't want to shut it down completely, but it's possible to modify the config file so that "unnecessary" things don't get logged and thus don't create disk traffic. `Cups` writes to disk periodically, so consider shutting it down and only enable it manually when needed.

**Code Listing 5.4:** Disabling cups in battery mode  
**# rc-update del cupsd battery**

You can also use Im-profiler from laptop-mode-tools (see above) to find services to disable. Once you eliminated all of them, go on with configuring `hdparm`.

## hdparm

The second possibility is using a small script and `hdparm`. Skip this if you are using laptop-mode. Otherwise, create `/etc/init.d/pmg_hda`:

**Code Listing 5.5:** Using `hdparm` for disk standby  
**#!/sbin/runscript**

```
depend() {
after hdparm
}

start() {
ebegin "Activating Power Management for Hard Drives"
hdparm -q -S12 /dev/hda
eend $?
}
```



```
stop () {
begin "Deactivating Power Management for Hard Drives"
hdparm -q -S253 /dev/hda
end $?
}
```

See man hdparm for the options. If your script is ready, add it to the battery runlevel.

**Code Listing 5.6:** Automate disk standby settings

```
# chmod +x /etc/init.d/pmg_hda
# /sbin/depscan.sh
# rc-update add pmg_hda battery
```

**Important:** Be careful with sleep/spin down settings of your hard drive. Setting it to small values might wear out your drive and lose warranty.

## Other tricks

Another possibility is to deactivate swap in battery mode. Before writing a swapon/swapoff switcher, make sure there is enough RAM and swap isn't used heavily, otherwise you'll be in big problems.

If you don't want to use laptop-mode, it's still possible to minimize disk access by mounting certain directories as tmpfs - write accesses are not stored on a disk, but in main memory and get lost with unmounting. Often it's useful to mount /tmp like this - you don't have to pay special attention as it gets cleared on every reboot regardless whether it was mounted on disk or in RAM. Just make sure you have enough RAM and no program (like a download client or compress utility) needs extraordinary much space in /tmp. To activate this, enable tmpfs support in your kernel and add a line to /etc/fstab like this:

**Code Listing 5.7:** Editing /etc/fstab to make /tmp even more volatile

```
none /tmp tmpfs size=32m 0 0
```

Warning: Pay attention to the size parameter and modify it for your system. If you're unsure, don't try this at all, it can become a performance bottleneck easily. In case you want to mount /var/log like this, make sure to merge the log files to disk before unmounting. They are essential. Don't attempt to mount /var/tmp like this. Portage uses it for compiling...

## 6. Power Management For Other Devices

### Graphics Cards

In case you own an ATI graphics card supporting PowerPlay (dynamic clock scaling for the the graphics processing unit GPU), you can activate this feature in X.org. Open /etc/X11/xorg.conf and add (or enable) the DynamicClocks option in the Device section. Please notice that this feature will lead to crashes on some systems.

**Code Listing 6.1:** Enabling ATI PowerPlay support in X.org

```
Section "Device"
[...]
```

```
Option "DynamicClocks" "on"
EndSection
```

## Wireless Power Management

Wireless LAN cards consume quite a bit of energy. Put them in Power Management mode in analogy to the `pmg_hda` script.

Note: This script assumes your wireless interface is called `wlan0`; replace this with the actual name of your interface.

### Code Listing 6.2: WLAN Power Management automated

```
#!/sbin/runscript
start() {
    ebegin "Activating Power Management for Wireless LAN"
    iwconfig wlan0 power on
    eend $?
}

stop () {
    ebegin "Deactivating Power Management for Wireless LAN"
    iwconfig wlan0 power off
    eend $?
}
```

Starting this script will activate power saving features for `wlan0`. Save it as `/etc/init.d/pmg_wlan0` and add it to the battery runlevel like the disk script above. See `man iwconfig` for details and more options like the period between wakeups or timeout settings. If your driver and access point support changing the beacon time, this is a good starting point to save even more energy.

### Code Listing 6.3: Power Management for WLAN

```
# chmod +x /etc/init.d/pmg_wlan0
# /sbin/depscan.sh
# rc-update add pmg_wlan0 battery
```

## USB Power Management

There are two problems with USB devices regarding energy consumption: First, devices like USB mice, digital cameras or USB sticks consume energy while plugged in. You cannot avoid this (nevertheless remove them in case they're not needed). Second, when there are USB devices plugged in, the USB host controller periodically accesses the bus which in turn prevents the CPU from going into sleep mode. The kernel offers an experimental option to enable suspension of USB devices through driver calls or one of the power/state files in `/sys`.

### Code Listing 6.4: Enabling USB suspend support in the kernel

```
Device Drivers
USB support
[*] Support for Host-side USB
[*] USB suspend/resume (EXPERIMENTAL)
```

## 7. Sleep States: *sleep, standby, and suspend to disk*

ACPI defines different sleep states. The more important ones are

S1 aka Standby

S3 aka Suspend to RAM aka Sleep

S4 aka Suspend to Disk aka Hibernate

They can be called whenever the system is not in use, but a shutdown is not wanted due to the long boot time.

### Sleep (S3)

The ACPI support for these sleep states is marked experimental for good reason. APM sleep states seem to be more stable, however you can't use APM and ACPI together.

**Code Listing 7.1:** Kernel configuration for the various suspend types

```
Power Management Options --->
```

```
[*] Power Management support
```

```
ACPI (Advanced Configuration and Power Interface) Support --->
```

```
[*] ACPI Support
```

```
[*] Sleep States
```

Once your kernel is properly configured, you can use the hibernate-script to activate suspend or sleep mode. Let's install that first.

**Code Listing 7.2:** Installing the hibernate-script

```
# emerge hibernate-script
```

Some configuration has to be done in /etc/hibernate. The default package introduces two configuration files hibernate.conf and ram.conf.

To configure sleep, edit ram.conf in /etc/hibernate. UseSysfsPowerState mem is already setup correctly, but you have to go through the rest of the configuration file and set it up for your system. The comments and option names will guide you. If you use nfs or samba shares over the network, make sure to shutdown the appropriate init scripts to avoid timeouts.

Ready? Now is the last chance to backup any data you want to keep after executing the next command. Notice that you probably have to hit a special key like Fn to resume from sleep.

**Code Listing 7.3:** Calling sleep

```
# hibernate-ram
```

If you're still reading, it seems to work. You can also setup standby (S1) in a similar way by copying ram.conf to standby.conf and creating a symlink /usr/sbin/hibernate-standby pointing to /usr/sbin/hibernate. S3 and S4 are the more interesting sleep states due to greater energy savings however.

## Hibernate (S4)

This section introduces hibernation, where a snapshot of the running system is written to disk before powering off. On resume, the snapshot is loaded and you can go on working at exactly the point you called hibernate before.

Warning: Don't exchange non hot-pluggable hardware when suspended. Don't attempt to load a snapshot with a different kernel image than the one it was created with. Shutdown any NFS or samba server/client before hibernating.

There are two different implementations for S4. The original one is swsusp, then there is the newer suspend2 with a nicer interface (including fbsplash support). A feature comparison is available at the suspend2 Homepage. There used to be Suspend-to-Disk (pmdisk), a fork of swsusp, but it has been merged back.

Suspend2 is not included in the mainline kernel yet, therefore you either have to patch your kernel sources with the patches provided by [suspend2.net](http://suspend2.net) or use `sys-kernel/suspend2-sources`.

The kernel part for both swsusp and suspend2 is as follows:

### Code Listing 7.4: Kernel configuration for the various suspend types

```
Power Management Options --->
(hibernate with swsusp)
[*] Software Suspend
    (replace /dev/SWAP with your swap partition)
    (/dev/SWAP)    Default resume partition

(hibernate with suspend2)
Software Suspend 2
--- Image Storage (you need at least one writer)
[*]  File Writer
[*]  Swap Writer
--- General Options
[*]  LZF image compression
    (replace /dev/SWAP with your swap partition)
    (swap:/dev/SWAP) Default resume device name
[]   Allow Keep Image Mode
```

The configuration for swsusp is rather easy. If you didn't store the location of your swap partition in the kernel config, you can also pass it as a parameter with the `resume=/dev/SWAP` directive. If booting is not possible due to a broken image, use the `noresume` kernel parameter. The `hibernate-cleanup` init script invalidates swsusp images during the boot process.

### Code Listing 7.5: Invalidating swsusp images during the boot process

```
# rc-update add hibernate-cleanup boot
```

To activate hibernate with swsusp, use the `hibernate` script and set `UseSysfsPowerState` disk in `/etc/hibernate/hibernate.conf`.

Warning: Backup your data before doing this. Run `sync` before executing one of the commands to have cached data written to disk. First try it outside of X, then with X running, but not logged in.

If you experience kernel panics due to uhci or similar, try to compile USB support as module and unload the modules before sending your laptop to sleep mode. There are configuration options for this in hibernate.conf

**Code Listing 7.6:** Hibernating with swsusp

```
# nano -w /etc/hibernate.conf  
(Make sure you have a backup of your data)  
# hibernate
```

The following section discusses the setup of suspend2 including fbsplash support for a nice graphical progress bar during suspend and resume.

The first part of the configuration is similar to the configuration of swsusp. In case you didn't store the location of your swap partition in the kernel config, you have to pass it as a kernel parameter with the resume2=swap:/dev/SWAP directive. If booting is not possible due to a broken image, append the noresume2 parameter. Additionally, the hibernate-cleanup init script invalidates suspend2 images during the boot process.

**Code Listing 7.7:** Invalidating suspend2 images during the boot process

```
# rc-update add hibernate-cleanup boot
```

Now edit /etc/hibernate/hibernate.conf, enable the suspend2 section and comment everything in the sysfs\_power\_state and acpi\_sleep sections. Do not enable the fbsplash part in global options yet.

**Code Listing 7.8:** Hibernating with suspend2

```
# nano -w /etc/hibernate.conf  
(Make sure you have a backup of your data)  
# hibernate
```

Please configure fbsplash now if you didn't do already. To enable fbsplash support during hibernation, the sys-apps/suspend2-userui package is needed. Additionally, you've got to enable the fbsplash USE flag.

**Code Listing 7.9:** Installing suspend2-userui

```
# mkdir -p /etc/portage  
# echo "sys-apps/suspend2-userui fbsplash" >> /etc/portage/package.use  
# emerge suspend2-userui
```

The ebuild tells you to make a symlink to the theme you want to use. For example, to use the livecd-2005.1 theme, run the following command:

**Code Listing 7.10:** Using the livecd-2005.1 theme during hibernation

```
# ln -sf /etc/splash/livecd-2005.1 /etc/splash/suspend2
```

If you don't want a black screen in the first part of the resume process, you have to add the suspend2ui\_fbsplash tool to your initrd image. Assuming you created the initrd image with splash\_geninitramfs and saved it as /boot/fbsplash-emergence-1024x768, here's how to do that.

**Code Listing 7.11:** Adding suspend2ui\_fb splash to an initrd image

```
# mount /boot
# mkdir ~/initrd.d
# cp /boot/fbsplash-emergence-1024x768 ~/initrd.d/
# cd ~/initrd.d
# gunzip -c fbsplash-emergence-1024x768 | cpio -idm --quiet -H newc
# rm fbsplash-emergence-1024x768
# cp /usr/sbin/suspend2ui_fb splash sbin/
# find . | cpio --quiet --dereference -o -H newc | gzip -9 > /boot/fbsplash-suspend2-emergence-1024x768
```

Afterwards adjust grub.conf respectively lilo.conf so that your suspend2 kernel uses /boot/fbsplash-suspend2-emergence-1024x768 as initrd image. You can now test a dry run to see if everything is setup correctly.

**Code Listing 7.12:** Test run for fbsplash hibernation

```
# suspend2ui_fb splash -t
```

Afterwards open /etc/hibernate/hibernate.conf again and activate the fbsplash options. Execute hibernate and enjoy.

## 8. Troubleshooting

Q: I'm trying to change the CPU frequency, but /sys/devices/system/cpu/cpu0/cpufreq/scaling\_governor does not exist.

A: Make sure your processor supports CPU frequency scaling and you chose the right CPUFreq driver for your processor. Here is a list of processors that are supported by cpufreq (kernel 2.6.7): ARM Integrator, ARM-SA1100, ARM-SA1110, AMD Elan - SC400, SC410, AMD mobile K6-2+, AMD mobile K6-3+, AMD mobile Duron, AMD mobile Athlon, AMD Opteron, AMD Athlon 64, Cyrix Media GXm, Intel mobile PIII and Intel mobile PIII-M on certain chipsets, Intel Pentium 4, Intel Xeon, Intel Pentium M (Centrino), National Semiconductors Geode GX, Transmeta Crusoe, VIA Cyrix 3 / C3, UltraSPARC-III, SuperH SH-3, SH-4, several "PowerBook" and "iBook2" and various processors on some ACPI 2.0-compatible systems (only if "ACPI Processor Performance States" are available to the ACPI/BIOS interface).

Q: My laptop supports frequency scaling, but /sys/devices/system/cpu/cpu0/cpufreq/ is empty.

A: Look for ACPI related error messages with dmesg | grep ACPI. Try to update the BIOS, especially if a broken DSDT is reported. You can also try to fix it yourself (which is beyond the scope of this guide).

Q: My laptop supports frequency scaling, but according to /proc/cpuinfo the speed never changes.

A: Probably you have activated symmetric multiprocessing support (CONFIG\_SMP) in your kernel. Deactivate it and it should work. Some older kernels had a bug causing this. In that case, run emerge x86info, update your kernel as asked and check the current frequency with x86info -mhz.

Q: I can change the CPU frequency, but the range is not as wide as in another OS.

A: You can combine frequency scaling with ACPI throttling to get a lower minimum frequency. Notice that throttling doesn't save much energy and is mainly used for thermal management (keeping your laptop cool and quiet). You can read the current throttling state with `cat /proc/acpi/processor/CPU/throttling` and change it with `echo -n "0:x" > /proc/acpi/processor/CPU/limit`, where `x` is one of the `Tx` states listed in `/proc/acpi/processor/CPU/throttling`.

Q: When configuring the kernel, powersave, performance and userspace governors show up, but that ondemand thing is missing. Where do I get it?

A: The ondemand governor is only included in recent kernel sources. Try updating them.

Q: Battery life time seems to be worse than before.

A: Check your BIOS settings. Maybe you forgot to re-enable some of the settings.

Q: My battery is charged, but KDE reports there would be 0% left and immediately shuts down.

A: Check that battery support is compiled into your kernel. If you use it as a module, make sure the module is loaded.

Q: My system logger reports things like "logger: ACPI group battery / action battery is not defined".

A: This message is generated by the `/etc/acpi/default.sh` script that is shipped with `acpid`. You can safely ignore it. If you like to get rid of it, you can comment the appropriate line in `/etc/acpi/default.sh` as shown below:

Code Listing 8.1: Disabling warnings about unknown acpi events

```
*)    # logger "ACPI action $action is not defined"
```

Q: I have a Dell Inspiron 51XX and I don't get any ACPI events.

A: This seems to be a kernel bug. Read on here.

Q: I activated the DynamicClocks option in `xorg.conf` and now X.org crashes / the screen stays black / my laptop doesn't shutdown properly.

A: This happens on some systems. You have to disable DynamicClocks.

Q: I want to use `suspend2`, but it tells me my swap partition is too small. Resizing is not an option.

A: If there is enough free space on your system, you can use the `filewriter` instead of the `swapwriter`. The `hibernate-script` supports it as well. More information can be found in `/usr/src/linux/Documentation/power/suspend2.txt`.

Q: I just bought a brand new battery, but it only lasts for some minutes! What am I doing wrong?

A: First follow your manufacturer's advice on how to charge the battery correctly.

Q: The above didn't help. What should I do then?

A: Some batteries sold as "new" are in fact old ones. Try the following:

Code Listing 8.2: Querying battery state

```
$ grep capacity /proc/acpi/battery/BAT0/info
```

```
design capacity: 47520 mWh
```

```
last full capacity: 41830 mWh
```

If the "last full capacity" differs significantly from the design capacity, your battery is probably broken. Try to claim your warranty.

Q: My problem is not listed above. Where should I go next?

A: Don't fear to contact me, Dennis Nienhüser, directly. The Gentoo Forums are a good place to get help as well. If you prefer IRC, try the `#gentoo-laptop` channel at `irc.freenode.net`.



# Руководство по udev в Gentoo

Ссылка на оригинал: <http://www.gentoo.org/doc/ru/udev-guide.xml>

С версии: 1.0

## 1. Что такое udev?

### Каталог /dev

Когда пользователи Linux разговаривают о начинке своих систем среди людей, полагающих, что Linux — это какой-то вид вируса или марка кофе, пользуясь оборотами вроде «слэш дев слэш что—нибудь» (/dev/foo), на них наверняка смотрят косо. Но для счастливицков (естественно, включая вас), /dev/hda1 — просто быстрый способ упоминания первого раздела ведущего диска IDE на первичном канале. Разве нет?

Все мы знаем, что такое файл устройства. Некоторые даже знают, почему рядом с ними видны специальные числа, если посмотреть на результат `ls -l`, запущенной в каталоге /dev. Но мы принимаем за данность, что первичный ведущий диск IDE называется /dev/hda. У вас может быть и не так, но это всего лишь пример.

Представим себе устройства «горячей замены» вроде USB, IEEE1394, заменяемой PCI и т.д. Какое из них является первым по счету? Надолго ли? Как изменятся названия других устройств, если первое отключить? Как это повлияет на текущие операции? Было бы забавно, если бы задание печати вдруг переключалось с вашего супер навороченного лазерного принтера на старый полудохлый матричный из-за того, что ваша мама решила выдернуть разъем лазерного принтера, а тот оказался первым в списке, не так ли?

Перейдем к udev. Цели проекта udev как занимательны, так и актуальны:

- запуск драйверов в режиме пользователя

- динамическое создание и удаление файлов устройств

- согласованное назначение имен устройств

- предоставление интерфейса программирования (API) для режима пользователя

Разработка udev состоит из трех отдельных проектов: namedev, libsysfs и, конечно, udev.

### namedev

Namedev позволяет отделить именованье устройств от программы udev. Это дает возможность гибкого использования правил и схем именования, разработанных отдельно. Подсистема именования предоставляет стандартный интерфейс, который может использоваться udev.

Сейчас, через namedev представлена только одна схема именования, предоставляемая LANANA. Она используется на большинстве Linux—систем, и

поэтому хорошо подходит основной массе пользователей Linux.

В `namedev` для определения имени конкретного устройства используется последовательность из пяти шагов. Если на одном из шагов имя установлено, оно и используется. Вот эти шаги:

1. по метке или серийному номеру
2. по номеру устройства шины
3. по топологии шины
4. статическое имя
5. имя, назначенное в ядре

По метке или серийному номеру — проверяется, есть ли у устройства уникальный идентификатор. Например, у устройств USB есть уникальный серийный номер USB, а у устройств SCSI — уникальный UUID. Если `namedev` находит соответствие уникального номера и определенного конфигурационного файла, используется имя, указанное в конфигурационном файле.

По номеру устройства шины — проверяется номер устройства, подключаемого к шине. В среде, где нет возможности «горячей замены», этого шага достаточно для идентификации устройства (например, номер на шине PCI редко меняется за время существования системы). Опять же, если `namedev` находит соответствие номера устройства шины и определенного конфигурационного файла, используется имя, указанное в конфигурационном файле.

Подобным образом, по топологии шины — это, скорее, статический способ идентификации устройств, подходящий до тех пор, пока пользователь не начинает их переключать. Когда позиция устройства совпадает со значением, заданным пользователем, используется указанное имя.

Четвёртый шаг, статическое имя — это простая подстановка строки. Замененное имя присваивается, когда имя, назначенное в ядре (по умолчанию) совпадает с заданной строкой замены.

Последний шаг (имя, назначенное в ядре) покрывает все остальные случаи: при этом берётся имя по умолчанию, присвоенное в ядре. В большинстве случаев этого достаточно, поскольку имя соответствует принятому в современных Linux-системах.

## **libsysfs**

`udev` взаимодействует с ядром через псевдофайловую систему `sysfs`. Проект `libsysfs` предлагает обобщенный интерфейс программирования для доступа к сведениям, предоставляемым файловой системой `sysfs`. Это даёт возможность опроса разнородных устройств без необходимости знания их типа.

## **udev**

Каждый раз, когда ядро обнаруживает обновление структуры устройств, оно вызывает программу `/sbin/hotplug`. `Hotplug`, в свою очередь, запускает приложения, указанные в каталоге `/etc/hotplug.d/default`, где также находится символическая ссылка на приложение `udev`. `Hotplug` направляет сведения, данные ядром, в программу `udev`, которая производит необходимые действия над структурой `/dev` (создавая или удаляя файлы устройств).

## 2. Как использовать udev в Gentoo

### Требования к системе

Файловая система udev должна использоваться вместе с ядром версии 2.6 (пакеты `vanilla-sources` или `gentoo-sources`, и профилем 2005.0). Если вы используете такое ядро, убедитесь, что у вас есть самая последняя версия `sys-apps/baselayout`. Вот и всё, что потребуется.

**Листинг 2.1:** Установка udev  
**# emerge udev**

**udev** установит пакет `hotplug-base`, как одну из зависимостей. Не нужно устанавливать пакет `hotplug` до тех пор, пока вы хотите, чтобы модули автоматически загружались при подключении устройств. `hotplug` также управляет автоматическим включением сетевых устройств и загрузкой встроенного ПО.

**Листинг 2.2:** Установка дополнительных сценариев hotplug  
**# emerge hotplug**

Если хотите, чтобы загружались модули для устройств, подключаемых до загрузки, используйте пакет `coldplug`:

**Листинг 2.3:** Установка пакета coldplug  
**# emerge coldplug**

И не забудьте добавить `coldplug` на загрузочный уровень запуска (`boot`):

**Листинг 2.4:** Добавление coldplug на загрузочный уровень запуска  
**# rc-update add coldplug boot**

Что же касается ядра, не забудьте выбрать следующие параметры:

**Листинг 2.5:** Требуемые параметры ядра  
General setup --->  
[\*] Support for hot-pluggable devices

File systems --->  
Pseudo filesystems --->  
[\*] /proc file system support  
[\*] Virtual memory file system support (former shm fs)

По вашему усмотрению, можете оставить поддержку файловой системы `/dev file system support (OBSOLETE)` активной, но обязательно убедитесь в том, что параметр «Automatically mount at boot» выключен:

**Листинг 2.6:** Автоматически не монтировать devfsd  
File systems --->  
Pseudo Filesystems --->  
[\*] /dev file system support (OBSOLETE)  
[ ] Automatically mount at boot

Если вы используете `genkernel`, не забудьте запустить её с параметром `--udev`, для включения всех необходимых параметров конфигурации ядра. Конфигурация по умолчанию, даваемая этим обращением к `genkernel`, достаточна.

## Конфигурация

Если вы хотите использовать улучшения `udev`, добавленные Gentoo для более удобного использования, не читайте дальше. Gentoo будет использовать `udev`, сохраняя неизменной структуру `/dev`, так что вы никогда не потеряете специальные файлы устройств. Сценарии инициализации Gentoo не будут запускать демон `devfsd` и деактивируют `devfs` при загрузке.

Но если вы «крепкий орешек» и хотите запустить любыми средствами систему только с `udev` и без улучшений, к чему и стремится разработка `udev` (включая сложности в отсутствии узлов устройств, потому, что `udev` их ещё не поддерживает), тогда читайте дальше :)

Мы деактивируем правила, которые сохраняют специальные файлы устройств: отредактируйте переменную `RC_DEVICE_TARBALL` в `/etc/conf.d/rc` и установите её значение на `no`:

**Листинг 2.7:** `/etc/conf.d/rc`  
`RC_DEVICE_TARBALL="no"`

Если вы включили поддержку `devfs` в вашем ядре, вы можете деактивировать её в конфигурации загрузчика: добавьте `gentoo=nodevfs` как параметр ядра. Если вы хотите использовать `devfs` и деактивировать `udev`, добавьте `gentoo=noudev`.

## 3. Известные проблемы

### Отсутствие специальных файлов устройств при загрузке

Если ваша машина не может успешно загрузиться и выдаётся ошибка о том, что путь `/dev/null` не найден или отсутствует первоначальная консоль, то проблема в том, что отсутствуют некоторые файлы устройств, которые должны быть доступны до того, как `/dev` смонтирован и обработан `udev`. Это часто встречается на машинах, где Gentoo установлен со старых носителей.

Если вы используете `sys-apps/baselayout-1.8.12` или более позднюю версию, эта проблема облегчается, так как процесс загрузки должен управлять до полного завершения. Однако, чтобы избавиться от этих раздражающих предупреждений, вам следует создать отсутствующие специальные файлы, как описано далее.

Выполните следующие команды для того, чтобы увидеть, какие специальные файлы присутствуют до обработки `udev` в `/dev`:

**Листинг 3.1:** Вывод списка специальных файлов устройств, доступных при загрузке

```
# mkdir test
# mount --bind / test
# cd test/dev
# ls
```

Устройства `/dev/null` и `/dev/console` необходимы для успешной загрузки. Если они не были показаны в предыдущем тесте, вы должны создать их самостоятельно. Выполните следующие команды в каталоге `test/dev/`:

**Листинг 3.2:** Создание необходимых специальных файлов устройств

```
# mknod -m 660 console c 5 1
# mknod -m 660 null c 1 3
```

Когда закончите, не забудьте отмонтировать каталог `test/`:

**Листинг 3.3:** Отмонтирование каталога `test/`

```
# cd ../../
# umount test
# rmdir test
```

## udev и nvidia

Если вы используете коммерческий драйвер от nVidia и сервер X не запускается только в системе с udev, убедитесь что у вас:

название модуля nvidia указано в файле `/etc/modules.autoload.d/kernel-2.6`  
версия nvidia-kernel больше или равна `media-video/nvidia-kernel-1.0.5336-r2`  
версия baselayout как минимум `sys-apps/baselayout-1.8.12`

Если `xorg-x11` отказывается запускаться, возможно, отсутствует файл устройства `/dev/nvidia`. В этом случае запустите `/sbin/NVmakedevices.sh`, чтобы его создать или пересоздать.

## Не появляются имена LVM2

Если вы используете вместе udev и Logical Volume Manager 2 (LVM—менеджер логических разделов), вы можете заметить, что созданные вами группы разделов и логические разделы исчезли. Что ж, на самом деле нет, но они, к несчастью, названы `/dev/dm-#`, где `#`—это 0, 1, ... и т.д.

Чтобы это исправить, отредактируйте файл `/etc/udev/rules.d/50-udev.rules` и уберите комментарий со следующей строки:

**Листинг 3.4:** Как убрать комментарий со строки из файла `/etc/udev/rules.d/50-udev.rules`

```
KERNEL="dm-[0-9]*", PROGRAM="/sbin/devmap_name %M %m", NAME="%k",
SYMLINK="%c"
```

Далее, установите пакет `sys-fs/multipath-tools`, который содержит программу `devmap_name`.

**Листинг 3.5:** Установка пакета `multipath-tools`

(На момент написания, пакет `multipath-tools` был доступен только в тестовой ветке дерева Portage)

```
# echo "sys-fs/multipath-tools-0.4.2 ~x86" >>
/etc/portage/package.keywords
# emerge multipath-tools
```

## Отличающиеся названия в DevFS и udev

Даже если наша цель—иметь совместимую схему именований между этими динамическими системами управления устройствами, иногда случаются различия в именованиях.

Сообщалось о коллизии в работе контроллера HP Smart Array 5i (более точно - модуль ядра cciss). При использовании udev, устройства именовались как `/dev/cciss/cXdYpZ`, где X, Y и Z - числа. При использовании devfs, устройства именовались `/dev/hostX/targetY/partZ` или являлись символическими ссылками на `/dev/cciss/cXdY`.

В этом случае, не забудьте обновить файл `/etc/fstab` и конфигурационные файлы загрузчика соответственно.

То же самое происходит со всеми символическими ссылками, которые существуют в каталоге `/dev`, такими как `/dev/mouse`, которые udev больше не создаёт. Обязательно проверьте настройку графического сервера X и посмотрите секцию Device, указывает ли там строка настройки на существующий файл устройства.

Другая проблема заключается в различном именовании терминалов между devfs и udev. Если в devfs терминалы называются tty, то в udev они называются vc. Это может привести к проблемам если возможность входа root с консоли ограничивается с помощью `/etc/securetty`. Чтобы обеспечить возможность входа root с консоли, нужно обязательно изменить tty1 на vc/1 в `/etc/securetty`.

## Другие проблемы

Если файлы устройств не создаются, когда соответствующий модуль загружается из `/etc/modules.autoload.d/kernel-2.6`, но появляются когда вы загружаете этот модуль вручную при помощи `modprobe`, тогда вам стоит произвести обновление до `sys-apps/baselayout-1.8.12` или более поздней версии.

Поддержка устройств кадрового буфера (`/dev/fb/*`) включена в ядро с версии 2.6.6-rc2.

В ядрах старше, чем 2.6.4, вам следует явно включить поддержку для файловой системы `/dev/pts`.

**Листинг 3.6:** Включение поддержки файловой системы `/dev/pts`

```
File systems --->
```

```
Pseudo filesystems --->
```

```
[*] /dev/pts file system for Unix98 PTYs
```

## 4. Ссылки на ресурсы и благодарности

Разговор об udev начался на симпозиуме по Linux в Оттаве в 2003 году (Linux Symposium, Ottawa, Ontario Canada - 2003) Гером Кроа-Хартманом (Greg Kroah-Hartman) из корпорации IBM, который дал ясное понимание для применения udev.

Decibel's UDEV Primer - документ о udev и Gentoo.

Написание правил для udev члена сообщества разработки Gentoo Дэниела Дрэйка

(Daniel Drake) - отличный документ для того, чтобы научиться управлять установкой udev.

## Gentoo LDAP-DNS Guide (Англ.)

Ссылка на оригинал: <http://www.gentoo.org/doc/en/ldapdns-guide.xml>

С версии: 1.0

### 1. Introduction

LDAPDNS is a small server that provides DNS services to your network. With DNS services, you can manage your hostnames in a central manner deprecating the tedious /etc/hosts updates every time a system is added/removed or has changed its host name.

The LDAPDNS package uses an LDAP service to store all DNS records (which is, simply explained, a line that contains the mapping between a hostname and an IP address). LDAP is a standard protocol to obtain information from a hierarchically represented knowledge base (directories). The most well-known LDAP service for Linux is OpenLDAP, a free LDAP implementation.

With this small guide at hand, you should be able to set up DNS services on your network with as little effort as possible.

### 2. Configuring LDAP-DNS

First, install net-dns/ldapdns.

**Code Listing 2.1:** Installing ldapdns  
**# emerge net-dns/ldapdns**

Next, configure ldapdns to host the DNS records for your network. We use ldapdns-conf which uses the following syntax:

**Code Listing 2.2:** Syntax for ldapdns-conf

```
ldapdns-conf acct logacct /path yourip ldaphost dn [suffix]
Keyword      Explanation      Example
acct         Username as which ldapdns will run      ldapdns
logacct      Username as which the ldapdns logging will run  dnslog
/path        Chrooted home directory for ldapdns      /var/lib/ldapdns
yourip       IP address to listen to      127.0.0.1
ldaphost     Address of the LDAP service      ldapi://%2fvar%2frun%2fopenldap%2fslapd.sock
dn           LDAP login credentials      cn=Manager,dc=yourdomain
suffix       Optional default ldap path to add on all queries  ou=Machines,dc=yourdomain
```

You should substitute the 127.0.0.1 IP address with an IP address that all hosts can reach and don't forget to use the domain name you want instead of yourdomain.

**Code Listing 2.3:** Example ldapdns-conf run

```
# ldapdns-conf ldapdns dnslog /var/lib/ldapdns 127.0.0.1 ldapi://%%2fvar%2frun%2fopenldap%2fslldapd.sock  
cn=Manager,dc=yourdomain ou=Machines,dc=yourdomain
```

Now set the LDAP login password:

**Code Listing 2.4:** Setting the LDAP login password

```
# echo YourSecretPassword > /var/lib/ldapdns/root/password  
# chmod 0400 /var/lib/ldapdns/root/password
```

Now configure ldapdns to use the simple authentication method and the cosine LDAP schemas:

**Code Listing 2.5:** Configuring ldapdns

```
# echo simple > /var/lib/ldapdns/env/LDAP_AUTH  
# echo cosine > /var/lib/ldapdns/env/SCHEMA
```

You can optionally set an e-mail address for the service:

**Code Listing 2.6:** Setting a host master e-mail address

```
# echo YourMail@address > /var/lib/ldapdns/env/HOSTMASTER
```

Add the service to the supervise scan and run the service supervisor if you haven't started it already:

**Code Listing 2.7:** Managing supervise services

```
# ln -s /var/lib/ldapdns /service  
# /etc/init.d/svscan start  
# rc-update add svscan default
```

### 3. Configuring OpenLDAP

Now we need to configure OpenLDAP with the DNS schema. Open up /etc/openldap/sldap.conf with your favorite editor and make sure the following three lines are listed:

**Code Listing 3.1:** Editing /etc/openldap/sldap.conf

```
include /etc/openldap/schema/cosine.schema  
include /etc/openldap/schema/inetorgperson.schema  
include /etc/openldap/schema/nis.schema
```

Bootstrap LDAP with the base dn you defined previously with ldapdns-conf. To accomplish this, we first create a file called bootstrap.ldif (it is just a name) which we fill up with DNS information.

As an example, we provide a bootstrap.ldif file for the fictitious domain cherchetoujours.org.



Now bootstrap your LDAP with this information:

**Code Listing 3.2:** Bootstrapping LDAP

```
# ldapadd -x -D "cn=Manager,dc=yourdomain" -W -f bootstrap.ldif
```

## ***4. Testing the Installation***

Now you are all set. Test your setup using nslookup (part of net-dns/bind-tools):

**Code Listing 4.1:** Testing the ldapdns configuration

```
# nslookup ns1.yourdomain
```

Note: nslookup is deprecated and may be removed from future releases.  
Consider using the 'dig' or 'host' programs instead. Run nslookup with  
the '-sil[ent]' option to prevent this message from appearing.

```
Server:      127.0.0.1
```

```
Address:     127.0.0.1#53
```

```
Name:  ns1.example.com
```

```
Address: (Your server IP address)
```

# Руководство по настройке Samba в режиме PDC с использованием LDAP

**Ссылка на оригинал:**

[http://ru.gentoo-wiki.com/Samba\\_PDC\\_LDAP](http://ru.gentoo-wiki.com/Samba_PDC_LDAP)

**С версии: 1.2**

**смотрите также:**

[Настройка и использование централизованного управления сервисами с использованием сервера LDAP.](#)

## Устанавливаемые пакеты

Установим нужные пакеты:

```
# USE="ldap acl ldapsam pam gdbm samba ssl tcpd winbind" emerge samba openldap  
acl nss_ldap pam_ldap
```

**ВАЖНО:** Версия nss\_ldap должна быть не ниже 250-r1 (>=sys-auth/nss\_ldap-250-r1)

## OpenLDAP

### Конфигурирование OpenLDAP

Для примера используется домен amber.global.com который является дочерним доменом домена global.com, который работает под управлением Win2003 Server.

**Примечание:** Не решенной задачей осталось создание доверительных отношений между доменами и включение домена amber в лес global.com

**Файл:** /etc/openldap/slapd.conf

```
include /etc/openldap/schema/core.schema  
include /etc/openldap/schema/cosine.schema  
include /etc/openldap/schema/inetorgperson.schema  
include /etc/openldap/schema/misc.schema  
include /etc/openldap/schema/nis.schema  
include /etc/openldap/schema/openldap.schema  
include /etc/openldap/schema/samba.schema  
pidfile /var/run/openldap/slapd.pid  
argsfile /var/run/openldap/slapd.args  
  
access to dn.base=""  
by self write  
by * auth  
access to attr=userPassword  
by self write
```

```

        by * auth
access to attr=shadowLastChange
        by self write
        by * read
access to *
        by * read
        by anonymous auth

#loglevel 1
database        ldbm
suffix          "dc=amber,dc=global,dc=com"
rootdn          "cn=Manager,dc=amber,dc=global,dc=com"
# Пароль rootpw лучше всего указывать в зашифрованном виде.
# Для генерации шифрованного пароля используйте утилиту slappasswd
# Например: slappasswd -h {MD5}
# rootpw = secret
rootpw          {MD5}Xr4ilOzQ4PCOq3aQ0qbuaQ==
directory       /var/lib/openldap-ldbm

index objectClass      eq
index cn               eq,subinitial
index sn               eq,subinitial
index uid              eq,subinitial
index displayName      eq,subinitial
index uidNumber        eq
index gidNumber        eq
index memberUID        eq
index sambaSID         eq
index sambaPrimaryGroupSID eq
index sambaDomainName  eq

```

**Файл: /etc/openldap/ldap.conf**

```

HOST 127.0.0.1
BASE dc=sanaa,dc=global,dc=com

```

Проверяем разрешения на каталоги **/var/lib/openldap-\***

```
ls -la /var/lib/
```

### ВАЖНО:

```

drwx----- 2 ldap ldap 104 Июн  8 18:31 openldap-data
drwx----- 2 ldap ldap  72 Июн  8 18:31 openldap-ldbm
drwx----- 2 ldap ldap  72 Июн  8 18:31 openldap-slurp

```

## Запуск сервера OpenLDAP

Пока работаем без шифрования трафика, так как сервер LDAP и SAMBA работают на одном сервере, подправляем конфиг чтобы LDAP слушал 389 порт только на localhost.

**Файл: /etc/conf.d/slaped**

```

# conf.d file for the openldap-2.1 series
#
# To enable both the standard unciphpered server and the ssl encrypted
# one uncomment this line or set any other server starting options
# you may desire.
#
# OPTS="-h 'ldaps:// ldap:// ldapi://%2fvar%2frun%2fopenldap%2fslaped.sock'"

```

```
OPTS="-h 'ldap://127.0.0.1'"
```

и запускаем OpenLDAP

```
/etc/init.d/slapd start
```

## SAMBA

### Миграция OpenLDAP

Существует 2 пути создания учетных записей пользователей: домен уже существует и создание нового домена. В первом случае самбу нужно настроить в режиме BDC, перенести все учетные записи, и потом изолировав PDC, перезапустить самбу в режиме PDC. Во втором случае все еще проще, самбу сразу запускаем в режиме PDC и создаем стандартные учетные записи при помощи замечательного пакета smbldap-tools.

emerge	smbldap-tools
/etc/init.d/samba	start

Code: **configure.pl**

```
-----
      smbldap-tools script configuration
      -----
Before starting, check
. if your samba controller is up and running.
. if the domain SID is defined (you can get it with the 'net getlocalsid')

. you can leave the configuration using the Ctrl-c key combination
. empty value can be set with the "." character
-----
Looking for configuration files...

Samba Configuration File Path [/etc/samba/smb.conf] >
The default directory in which the smbldap configuration files are stored is shown.
If you need to change this, enter the full directory path, then press enter to
continue.
Smbldap-tools Configuration Directory Path [/etc/smbldap-tools/] >
-----
Let's start configuring the smbldap-tools scripts ...

. workgroup name: name of the domain Samba act as a PDC
  workgroup name [amber] >
. netbios name: netbios name of the samba controler
  netbios name [neptun] >
. logon drive: local path to which the home directory will be connected
(for NT Workstations). Ex: 'H:'
  logon drive [U:] >
. logon home: home directory location (for Win95/98 or NT Workstation).
(use %U as username) Ex: '\\neptun\%U'
  logon home (press the "." character if you don't want homeDirectory) [\\%L\users\%U]
>
. logon path: directory where roaming profiles are stored. Ex: '\\neptun\profiles\%U'
  logon path (press the "." character if you don't
want roaming profile) [\\%L\Profiles\%a\%U] >
. home directory prefix (use %U as username) [/home/%U] >
```

```
. default users' homeDirectory mode [700] >
. default user netlogon script (use %U as username) [] >
  default password validation time (time in days) [45] > 900
. ldap suffix [dc=amber,dc=global,dc=com] >
. ldap group suffix [ou=Groups] >
. ldap user suffix [ou=Users] >
. ldap machine suffix [ou=Users] >
. Idmap suffix [ou=Idmap] >
. sambaUnixIdPool: object where you want to store the next uidNumber
  and gidNumber available for new users and groups
  sambaUnixIdPool object (relative to ${suffix}) [sambaDomainName=amber] >
. ldap master server: IP address or DNS name of the master (writable) ldap server
  ldap master server [127.0.0.1] >
. ldap master port [389] >
. ldap master bind dn [cn=Manager,dc=amber,dc=global,dc=com] >
. ldap master bind password [] >
. ldap slave server: IP address or DNS name of the slave ldap server: can also
  be the master one
  ldap slave server [127.0.0.1] >
. ldap slave port [389] >
. ldap slave bind dn [cn=Manager,dc=amber,dc=global,dc=com] >
. ldap slave bind password [] >
. ldap tls support (1/0) [0] >
. SID for domain amber: SID of the domain (can be obtained with 'net getlocalsid
  neptun')
  SID for domain amber [S-1-5-21-1918777035-593721947-2697221154] >
. unix password encryption: encryption used for unix passwords
  unix password encryption (CRYPT, MD5, SMD5, SSHA, SHA) [SSHA] > MD5
. default user gidNumber [513] >
. default computer gidNumber [515] >
. default login shell [/bin/bash] >
. default skeleton directory [/etc/skel] >
. default domain name to append to mail address [] >
-----
backup old configuration files:
  /etc/smbldap-tools/smbldap.conf->/etc/smbldap-tools/smbldap.conf.old
  /etc/smbldap-tools/smbldap_bind.conf->/etc/smbldap-tools/smbldap_bind.conf.old
writing new configuration file:
  /etc/smbldap-tools/smbldap.conf done.
  /etc/smbldap-tools/smbldap_bind.conf done.
```

Если мы не хотим, что бы у всех пользователей профили были перемещаемыми, то в файле `/etc/smbldap-tools/smbldap.conf` установим следующее значение:

**Файл:** `/etc/smbldap-tools/smbldap.conf`

```
...
userProfile=""
...
```

Инициализируем каталоги самбы в LDAP:

```
smbldap-populate -a Administrator -k 0 -m 0
```

## Конфигурация SAMBA

**Файл:** `/etc/samba/smb.conf`

```

[global]
    workgroup = amber
    netbios name = neptun
    realm = amber.global.com
    nt acl support = yes
    acl compatibility = win2k
    map acl inherit = yes
    server string = Samba Server %v
    interfaces = eth0
    bind interfaces only = yes
    hosts allow = 192.168.7. 127.
    log file = /var/log/samba/log.%m
    debug level = 9
    max log size = 500
    socket options = TCP_NODELAY SO_SNDBUF=8192 SO_RCVBUF=8192
    security = user
    os level = 250
    passdb backend = ldapsam:"ldap://127.0.0.1/"
    enable privileges = yes

    passwd program = /usr/sbin/smbldap-passwd "%u"
    passwd chat = *new*password* %n\n *new*password* %n\n *successfully*
    passdb expand explicit = no
    unix password sync = no
    ldap passwd sync = no

    ldap suffix = dc=amber,dc=global,dc=com
    ldap admin dn = cn=Manager,dc=amber,dc=global,dc=com
    ldap user suffix = ou=Users
    ldap group suffix = ou=Groups
# Т.к. для самбы компьютеры и пользователи - одно и то же,
# и искать она в дальнейшем записи компьютеров будет в пользователях,
# то для избежания дальнейших проблем при добавлении рабочих станций
# к домену мы вместо следующей строки
#
#     ldap machine suffix = ou=Computers
#
# напишем другую:
    ldap machine suffix = ou=Users
    ldap idmap suffix = ou=Idmap
    idmap backend = ldapsam:ldap://127.0.0.1/
    idmap uid = 10000-20000
    idmap gid = 10000-20000

    ldap delete dn = Yes
    ldap ssl = no

    add user script = /usr/sbin/smbldap-useradd -n -a "%u"
    delete user script = /usr/sbin/smbldap-userdel "%u"

    add group script = /usr/sbin/smbldap-groupadd -p "%g"
    delete group script = /usr/sbin/smbldap-userdel "%g"

    add user to group script = /usr/sbin/smbldap-groupmod -m "%u" "%g"
    delete user from group script = /usr/sbin/smbldap-groupmod -x "%u" "%g"
    set primary group script = /usr/sbin/smbldap-usermod -g "%g" "%u"

    add machine script = /usr/sbin/smbldap-useradd -w "%u"

#PDC
domain master = yes
preferred master = yes

```

```

#BDC
# domain master = no
# preferred master = no
# domain logons = Yes

logon script =
# Если хотите, что бы профили всех пользователей были перемещаемыми
# и хранились на сервере (со всеми гигабайтами фильмов и личных фотографий)
# то укажите такое значение следующего параметра:
#
# logon path = \\%L\Profiles\%a\%U
#
# Если вы не хотите гонять профили по сети, оставьте значение пустым,
# (но ни в коем случае не комментируйте эту строку, она просто получит
# значение по умолчанию), вот так:
# logon path =
# logon drive = U:
# logon home = \\%L\users\%U

#===== Share Definitions =====
[netlogon]
comment = Network Logon Service
path = /var/lib/samba/netlogon
browseable = yes
guest ok = yes
writable = no
share modes = no

[Profiles]
admin users = admin
create mode = 600
directory mode = 700
path = /var/lib/samba/profiles
browseable = yes
guest ok = yes
writable = yes

[homes]
comment = Home Directories
browseable = no
read only = no

[public]
path = /pub
guest ok = yes
read only = no

[users]
path = /home/users
writable = yes
printable = no

```

Добавим запуск winbind с самбой (если нужно):

**Файл:** /etc/conf.d/samba

```

...
daemon_list="smbd nmbd winbind"
...

```

Введём пароль рабочей станции:

```
smbpasswd -w secret
```

Введём контроллер домена, собственно в домен

```
net rpc join -S neptun -U Administrator
```

## **Настройка системы на авторизацию в LDAP**

--ladserg 14:05, 28 июля 2006 (UTC) У меня честно говоря не получилось сделать авторизацию пользователей samba через LDAP без настройки поддержки авторизации системных пользователей в LDAP, пришлось настраивать и это.

Сначала поправим файл /etc/ldap.conf, приведя его примерно к следующему виду:

**Файл: /etc/ldap.conf**

```
host 127.0.0.1
base dc=amber,dc=global,dc=com
ldap_version 3
rootbinddn cn=Manager,dc=amber,dc=global,dc=com
bind_timelimit 10
bind_policy soft
pam_filter objectClass=posixAccount
pam_password exop
nss_base_passwd ou=Users,dc=tty,dc=perm,dc=ru?one
nss_base_shadow ou=Users,dc=tty,dc=perm,dc=ru?one
nss_base_group ou=Groups,dc=tty,dc=perm,dc=ru?one
nss_base_hosts ou=Hosts,dc=tty,dc=perm,dc=ru?one
nss_base_services ou=Services,dc=tty,dc=perm,dc=ru?one
nss_base_networks ou=Networks,dc=tty,dc=perm,dc=ru?one
nss_base_protocols ou=Protocols,dc=tty,dc=perm,dc=ru?one
nss_base_rpc ou=Rpc,dc=tty,dc=perm,dc=ru?one
nss_base_ethers ou=Ethers,dc=tty,dc=perm,dc=ru?one
nss_base_netmasks ou=Networks,dc=tty,dc=perm,dc=ru?one
nss_base_bootparams ou=Ethers,dc=tty,dc=perm,dc=ru?one
nss_base_aliases ou=Aliases,dc=tty,dc=perm,dc=ru?one
nss_base_netgroup ou=Netgroup,dc=tty,dc=perm,dc=ru?one
ssl off
nss_reconnect_tries 4
nss_reconnect_sleeptime 1
nss_reconnect_maxsleeptime 16
nss_reconnect_maxcontries 2
```

Сим мы скажем nss\_ldap где и как искать записи пользователей и групп.

Теперь создадим файл /etc/ldap.secret и при помощи любого текстового редактора в plain/text виде занесём туда пароль пользователя, который выше у нас указан в опции rootbinddn, например пароль secret:

**Файл: /etc/ldap.secret**

```
secret
```

Затем непременно установим на него нужные права:

**Code: Установка прав на файл /etc/ldap.secret**

```
#chmod 600 /etc/ldap.secret
```



```
#chown root:wheel /etc/ldap.secret
```

Далее приведём файл `/etc/pam.d/system-auth` к следующему виду:

**Файл: `/etc/pam.d/system-auth`**

```
auth        required      pam_env.so
auth        sufficient     pam_unix.so likeauth nullok
auth        sufficient     pam_ldap.so use_first_pass
auth        required      pam_deny.so

account     sufficient     pam_ldap.so
account     required      pam_unix.so

password    required      pam_cracklib.so difok=2 minlen=8 dcredit=2 ocredit=2
retry=3
password    sufficient     pam_unix.so nullok md5 shadow use_authtok
password    sufficient     pam_ldap.so use_authtok
password    required      pam_deny.so

session     required      pam_limits.so
session     required      pam_unix.so
session     required      pam_mkhomedir.so skel=/etc/skel/ umask=077
session     optional      pam_ldap.so
```

Обратите внимание на строку:

```
session required pam_mkhomedir.so skel=/etc/skel/ umask=077
```

Она заставляет систему создавать домашние каталоги для тех пользователей у которых они ещё не созданы, при этом в новый каталог помещается содержимое директории `/etc/skel/` и задаётся маска каталога `0x700`

После чего правим файл `/etc/nsswitch.conf`, приводя его к следующему виду:

**Файл: `/etc/nsswitch.conf`**

```
passwd:      files ldap
shadow:      files ldap
group:       files ldap

hosts:       files dns
networks:    files dns

services:    db files
protocols:   db files
rpc:         ldap [NOTFOUND=return] db files
ethers:      ldap [NOTFOUND=return] db files
netmasks:    files
netgroup:    ldap [NOTFOUND=return] files
bootparams:  files

automount:   files
aliases:     files
```

**Предупреждение:** Ни в коем случае не добавляйте значение `ldap` к следующим базам: `hosts`, `networks`, `protocols`, `services`. Иначе вы рискуете не дожидаться следующей загрузки системы.

Всё, теперь мы указали системе брать пользователей как из системных файлов, так и из LDAP.

Перезагрузим наш компьютер, дабы убедиться что система грузится нормально.

Если система останавливается на загрузке udev, то смотрите ошибки в файле /etc/nsswitch.conf, может вы указали использовать ldap не в той базе.

## Управление пользователями

Ранее мы установили пакет smbldap-tools, теперь рассмотрим возможность управления пользователями с его помощью.

### Создание пользователя

```
smbldap-useradd [-o] [-a] [-b] [-w] [-i] [-u uid] [-g gid] [-G groups,,,]
[-n] [-d home] [-s shell] [-c gecos] [-m [-k]] [-t] [-P] [-A 0|1] [-B 0|1]
[-C sambaHomePath] [-D sambaHomeDrive] [-E sambaLogonScript] [-F
sambaProfilePath]
[-H sambaAcctFlags] [-N surname] [-S family name] [-M local mailAddress,,,]
[-T mailToAddress] [-?] user
```

Где:

```
user - системное имя создаваемого пользователя
-o - add the user in the organizational unit (relative to the user suffix)
-a - is a Windows User (otherwise, Posix stuff only)
-b - is a AIX User
-w - is a Windows Workstation (otherwise, Posix stuff only)
-i - is a trust account (Windows Workstation)
-u - uid
-g - gid
-G - список групп пользователя, разделённых запятой.
-n - do not create a group
-d - домашний каталог пользователя (по умолчанию /home/имя_пользователя)
-s - оболочка пользователя (по умолчанию /bin/false)
-c - отображаемое в Windows имя пользователя
-m - создать домашний каталог и скопировать в него файлы из /etc/skel
-k - указать иной каталог, из которого будут копироваться файлы при
создании домашнего каталога пользователя (используется с ключём -m)
-t - time. Wait 'time' seconds before exiting (when adding Windows
Workstation)
-P - ends by invoking smbldap-passwd
-A - возможность менять пароль пользователем, значение 0 если нет, 1 если да
-B - пользователь должен поменять пароль, значение 0 если нет, 1 если да
-C - домашний каталог samba (например '\\PDC-SRV\homes')
-D - буква диска для монтирования домашнего каталога samba (например 'H:')
-E - скрипт, выполняемый при входе в систему
-F - каталог профиля пользователя (например '\\PDC-SRV\profiles\foo')
-H - sambaAcctFlags (samba account control bits like '[NDHTUMWSLKI]')
-N - настоящее имя пользователя (для русских ещё и отчество)
-S - фамилия пользователя
-M - local mailAddress (comma seperated)
-T - mailToAddress (forward address) (comma seperated)
-? - отобразить помощь
```

Например создание пользователя ladserg:

```
smbldap-useradd -a -c 'Serg Alex Lad' -N 'Serg Alex' -S 'Lad' -s /bin/bash ladserg
```

К сожалению подружить smbldap-tools с русским мне не удалось, даже при использовании кодировки UTF-8.

Итак, в приведённом выше примере будет создан пользователь с системным именем ladserg, фамилией Lad, именем Serg Alex, оболочкой /bin/bash, домашним каталогом /home/ladserg. Флаг -a укажет, что пользователь также будет являться

пользователем домена.

## Изменение пароля

```
smbldap-passwd [-s] [-u] [-h] username
```

Где:

```
username      - имя пользователя
-h, -?, --help - показать помощь
-s            - обновить только samba пароль
-u            - обновить только UNIX пароль
```

Например:

```
smbldap-passwd ladserg
```

После чего дважды будет запрошен пароль.

Теперь можно попробовать зайти в систему под учётной записью только что созданного пользователя.

## Модификация пользователя

```
smbldap-usermod [-a] [-c comment] [-d home_dir] [-e expiration_date]
[-g initial_group] [-r new_login_name] [-p passwd] [-s shell] [-u uid [ -o]] [-x]
[-A canchange] [-B mustchange] [-C smbhome] [-D homedrive] [-E scriptpath]
[-F profilepath] [-G group[,...]] [-H acct-flags] [-N canonical_name]
[-S surname] [-P] login
```

Где:

```
-c      - Полное имя
-d      - Домашний каталог
-r      - новое имя пользователя (cn, sn и dn будут обновлены)
-u      - uid
-o      - uid может быть не уникальным
-g      - gid
-G      - список групп пользователя, разделённых запятой.
-s      - оболочка
-N      - настоящее имя пользователя (для русских ещё и отчество)
-S      - фамилия пользователя
-P      - ends by invoking smbldap-passwd
```

For samba users:

```
-a      - add sambaSAMAccount objectclass
-e      - expire date ("YYYY-MM-DD HH:MM:SS")
-A      - возможность менять пароль пользователем, значение 0 если нет, 1 если да
-B      - пользователь должен поменять пароль, значение 0 если нет, 1 если да
-C      - домашний каталог samba (например '\\PDC-SRV\homes')
-D      - буква диска для монтирования домашнего каталога samba (например 'H:')
-E      - скрипт, выполняемый при входе в систему
-F      - каталог профиля пользователя (например '\\PDC-SRV\profiles\foo')
-H      - sambaAcctFlags (samba account control bits like '[NDHTUMWSLKI]')
-I      - disable an user. Can't be used with -H or -J
-J      - enable an user. Can't be used with -H or -I
-M      - mailAddresses (comma seperated)
-T      - mailToAddress (forward address) (comma seperated)
-?|-h   - отобразить помощь
```

Например команда:

```
smbldap-usermod -A 1 ladserg
```

Позволит пользователю ladserg менять пароль. А команда:

```
smbldap-usermod -a slad-adm
```

Добавит к учётной записи пользователя slad-adm класс sambaSAMAccount, что сделает его пользователем samba.

## Удаление пользователя

```
smbldap-userdel [-r|-R|-?] username
```

Где:

```
-r      удалить домашний каталог
-R      удалить домашний каталог с запросом на удаление каждого файла
-?      отобразить помощь
```

Например команда:

```
smbldap-userdel -r slad-adm
```

удалит пользователя slad-adm, и его домашний каталог.

## Управление пользователями в оффтопике

Если вы вводите в домен компьютеры под управлением Windows, то вам пригодятся пара утилит, архив которых можно скачать отсюда:

<ftp://ftp.microsoft.com/Softlib/MSLFILES/SRVTOOLS.EXE>

## Источники

- <http://us4.samba.org/samba/docs/man/Samba-Guide/happy.html>
- [http://www.opennet.ru/base/net/ldap\\_spama\\_pdc.txt.html](http://www.opennet.ru/base/net/ldap_spama_pdc.txt.html)

# Создание виртуальной почтовой системы

Ссылка на оригинал: <http://www.gentoo.org/doc/ru/virt-mail-howto.xml>

С версии: 1.0

## 1. Введение

Для большинства пользователей gentoo простого почтового клиента и fetchmail (для сбора почты) достаточно. Однако, если вы размещаете на своей системе домен, вам необходим полноценный MTA (Mail Transfer Agent). И если вы размещаете на своей системе несколько доменов, тогда вам точно необходимо что-то мощное для обработки всей почты ваших пользователей. Эта система была разработана для элегантного решения этой проблемы.

Виртуальная почтовая система должна быть способна обработать почту от многочисленных доменов с множеством пользователей через разнообразные интерфейсы. С этим связано несколько проблем, которые надо решить. Для примера, что если вы имеете двух пользователей в разных доменах, которые хотят иметь одинаковые имена? Если вы предоставляете сервисы imap и smtp-авторизации, как вы комбинируете различные демоны авторизации в одной системе? Как обеспечиваете безопасность многочисленных компонентов которая содержит система? Как вы управляете этим всем?

Это howto покажет вам, как настроить гибкую систему обработки почты из стольких доменов, сколько ваша поддерживает система, как создавать виртуальных пользователей не требующих наличия аккаунта в системе, иметь доменные имена пользователей, как авторизовать пользователей через веб, imap, smtp и pop3 снова таки из одной базы данных, использовать ssl-транспорт для безопасности, как обрабатывать списки рассылки для любого домена на машине, и контролировать все хорошей и простой базой данных mysql.

Конечно существует множество путей настройки виртуальной почтовой системы. Другой способ возможно окажется более подходящим вашим нуждам. Дополнительную информацию вы можете найти на <http://www.qmail.org/> и <http://www.exim.org/>

Мы будем использовать следующие программы: apache, courier-imap, pam\_mysql, postfix, mod\_php, phpmyadmin, squirrelmail, cyrus-sasl, mysql, php, и mailman.

Убедитесь что вы добавили в файле /etc/make.conf в переменную USE следующие пакеты: USE="mysql imap libwww maildir sasl ssl". Иначе, возможно, вам придется перекомпилировать пакеты, для поддержки всех необходимых протоколов. Затем, хорошей идеей будет отключить любые другие почтовые и сетевые программы которые вам не нужны, вроде IPv6.

Важно: Это howto написано для postfix-2.0.x. Если вы используете postfix версии < 2 некоторые переменные используемые в этом документе могут отличаться. В таком случае рекомендуется обновить postfix. Некоторые пакеты используемые в этом документе зависят от версии. Вы поступите мудро, если прочтете документацию

поставляемую с пакетами, при возникновении вопросов.

**Важно:** В нашем документе мы будем использовать apache-1.3.x. В портежах Apache-2 был помечен stable, но как бы то ни было существуют еще некоторые проблемы интеграции с php. Пока php-поддержка в apache-2.0.x не будет отмечена stable, в этом документе будем продолжать использовать apache-1.3.x.

**Важно:** Вам необходимо доменное имя для работы общедоступного почтового сервера, или по крайней мере иметь MX-записи для домена. В идеале вы должны иметь контроль над двумя доменами, для извлечения пользы из функциональности новой виртуальной системы.

**Важно:** Убедитесь что в /etc/hostname указано правильное имя сервера, также убедитесь что в /etc/hosts нет конфликтных значений.

**Примечание:** Рекомендуется прочесть документ полностью и ознакомиться со всеми шагами настройки, перед тем как начинать инсталляцию. Если вы столкнетесь с проблемами при любом из шагов, проверьте troubleshooting guide в конце этого документа. Также не все упомянутые пакеты необходимы, просто настройка, описанная здесь, достаточна гибка. Для примера, если вам не требуется веб-интерфейса, вы свободны пропустить соответствующий раздел squirrelmail.

## 2. Начальная настройка postfix

**Листинг 2.1:** Инсталляция postfix  
# emerge postfix

**Предупреждение:** Проверьте чтобы у вас не были проинсталлированы любые другие МТА, например ssmtp, exim или qmail, иначе у вас могут быть БОЛЬШИЕ проблемы.

После инсталляции postfix, время его настроить. Измените следующие настройки в /etc/postfix/main.cf:

**Листинг 2.2:** /etc/postfix/main.cf

```
myhostname = $host.domain.name
mydomain = $domain.name
inet_interfaces = all
mydestination = $myhostname, localhost.$mydomain $mydomain
mynetworks = my.ip.net.work/24, 127.0.0.0/8
home_mailbox = .maildir/
local_destination_concurrency_limit = 2
default_destination_concurrency_limit = 10
```

Следующие изменения следуют для /etc/postfix/master.cf. Они включают режим подробного протоколирования для отладки:

**Листинг 2.3:** /etc/postfix/master.cf

```
# service type private unpriv chroot wakeup maxproc command + args
#          (yes) (yes) (yes) (never) (50)
```

```
#
=====
smtp inet n - n - - smtpd -v

(просто добавьте -v после smtpd)
```

Дальше правим /etc/mail/aliases для добавления локальных псевдонимов. По крайней мере туда должен быть включен псевдоним для root типа: root: your@email.address.

**Листинг 2.4:** Запуск postfix'a в первый раз

```
# /usr/bin/newaliases
(это создаст новые псевдонимы. Это требуется делать)
(когда вы создаете или обновляете файл псевдонимов.)
# /etc/init.d/postfix start
```

Теперь postfix запущен, запустите вашего любимого консольного почтового клиента и пошлите самому себе письмо. Я использую mutt для всей переписки из консоли. Примечание: Настоятельно рекомендуется проверить, чтобы postfix доставлял почту локальным пользователям, перед тем как перейти к следующему шагу.

### 3. Courier-imap

**Листинг 3.1:** Инсталляция courier-imap

```
# emerge courier-imap
```

**Листинг 3.2:** Настройка courier-imap

```
# cd /etc/courier-imap
(если вы хотите использовать возможности ssl в courier-imap или pop3, )
(вам необходимо создать сертификаты)
(если вы не хотите использовать ssl, просто пропустите следующий шаг )

# nano -w pop3d.cnf
# nano -w imapd.cnf
(измените значения C, ST, L, CN и адреса e-mail)

# mkpop3dcert
# mkimapdcert
```

**Листинг 3.3:** Запуск нужных вам сервисов courier

```
# /etc/init.d/courier-imapd start
# /etc/init.d/courier-imapd-ssl start
# /etc/init.d/courier-pop3d start
# /etc/init.d/courier-pop3d-ssl start
```

Снова запустите ваш любимый почтовый клиент и проверьте что сервисы начали принимать и посылать почту. Теперь когда основная часть заработала, перейдем к связыванию всех компонентов, для получения спокойно работающей системы. Снова, проверьте, что все что мы сделали работает, перед тем как переходить к следующему шагу.

## 4. Cyrus-sasl

Следующим шагом будет инсталляция cyrus-sasl. Sasl на самом деле играет роль передатчика авторизационных переменных к pam (Pluggable Authentication Modules), который передаст эту информацию mysql, для авторизации smtp пользователей. Мы не будем проверять работоспособность sasl, пока не настроим mysql, и не создадим тестовых пользователей. В конечном результате, он будет авторизовать пользователей в mysql.

Примечание: По некоторым причинам, sasl плохо работает с pam используя shadow. Для меня долгое время это была большая проблема. Если кто-нибудь знает почему sasl не авторизует пользователей из /etc/shadow в текущем релизе gentoo, пожалуйста напишите мне об этом, я буду очень рад услышать решение этой проблемы. E-mail.

**Листинг 4.1:** Инсталляция и настройка cyrus-sasl

```
# USE='-ldap -mysql' emerge cyrus-sasl
```

(мы не используем ldap и мы не используем возможности sasl-mysql, поэтому мы отключаем их )

Дальше правим /usr/lib/sasl2/smtp.conf.

**Листинг 4.2:** Запуск sasl

```
# nano -w /usr/lib/sasl2/smtp.conf
```

```
pwcheck_method: saslauthd
```

```
mech_list: LOGIN PLAIN
```

(важно выключить метод авторизации который мы не используем.

Они могут привести к граблям на некоторых почтовых клиентах)

```
# /etc/init.d/saslauthd start
```

## 5. SSL-сертификаты для Postfix и Apache

Дальше делаем ssl-сертификаты для postfix и apache.

**Листинг 5.1:**

```
# cd /etc/ssl/
```

```
# nano -w openssl.cnf
```

(измените следующие значения для вашего домена:)

```
countryName_default
```

```
stateOrProvinceName_default
```

```
localityName_default
```

```
O.organizationName_default
```

```
commonName_default
```

```
emailAddress_default.
```

(если каких-то переменных нет, просто добавьте их в любое удобное место)

```
# cd misc
```

```
# nano -w CA.pl
```

(нам необходимо добавить -nodes в код строк "# create a certificate" и

"# create a certificate request" таким образом чтобы наши новые ssl-сертификаты загрузились без пароля. Иначе после перезагрузки системы ssl-сертификаты будут не доступны )

(найдите эти строки в файле и измените их:)



#### **# create a certificate**

```
system ("$REQ -new -nodes -x509 -keyout newreq.pem -out newreq.pem $DAYS");
```

#### **# create a certificate request**

```
system ("$REQ -new -nodes -keyout newreq.pem -out newreq.pem $DAYS");
```

(далее создаем сертификат для postfix)

```
# ./CA.pl -newca
```

```
# ./CA.pl -newreq
```

```
# ./CA.pl -sign
```

```
# cp newcert.pem /etc/postfix
```

```
# cp newreq.pem /etc/postfix
```

```
# cp demoCA/cacert.pem /etc/postfix
```

(а теперь такой же для apache)

```
# openssl req -new > new.cert.csr
```

```
# openssl rsa -in privkey.pem -out new.cert.key
```

```
# openssl x509 -in new.cert.csr -out new.cert.cert -req -signkey new.cert.key -days 365
```

(оставьте получившиеся сертификаты, мы их проинсталлируем когда будем настраивать apache )

## **6. Добавим SSL и SASL поддержку в Postfix**

Теперь поправим конфигурацию postfix чтобы включить совместимость с sasl и ssl. Добавьте следующие параметры в конец файла, где их легко потом можно будет найти.

**Листинг 6.1:** /etc/postfix/main.cf

```
# nano -w /etc/postfix/main.cf
```

```
smtpd_sasl_auth_enable = yes
smtpd_sasl2_auth_enable = yes
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
smtpd_sasl_local_domain =
```

(опция broken\_sasl\_auth\_clients и метод авторизации используется только для outlook и outlook express, они не документированы. Опция smtpd\_sasl\_local\_domain добавляет имя домена для клиентов использующих smtp-авторизацию. Убедитесь что postfix отвергает пустое или просто имя пользователя, и они не могут авторизоваться )

```
smtpd_recipient_restrictions =
    permit_sasl_authenticated,
    permit_mynetworks,
    reject_unauth_destination
```

```
smtpd_use_tls = yes
#smtpd_tls_auth_only = yes
smtpd_tls_key_file = /etc/postfix/newreq.pem
smtpd_tls_cert_file = /etc/postfix/newcert.pem
smtpd_tls_CAfile = /etc/postfix/cacert.pem
smtpd_tls_loglevel = 3
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
tls_random_source = dev:/dev/urandom
```

(опция smtpd\_tls\_auth\_only закомментирована для простоты тестирования, вы можете включить ее позже, если захотите)

## # postfix reload

Теперь мы попробуем проверить работоспособность postfix'a, и то что он прочел файлы конфигурации.

**Листинг 6.2:** Проверка поддержки sasl и tls

```
# telnet localhost 25

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mail.domain.com ESMTP Postfix
EHLO domain.com
250-mail.domain.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN PLAIN
250-XVERP
250 8BITMIME
^]
telnet> quit
```

Проверьте, что ответ postfix'a содержит линии AUTH и STARTTLS. Как уже замечалось выше, AUTH пока не будет работать, потому что sasl пытается авторизовать из sasldb, вместо shadow по неизвестным причинам. Итак мы просто пропускаем это и дальше настраиваем mysql для принятия нашей авторизации и информации о виртуальных доменах.

## 7. MySQL

Для инсталляции mysql нам необходим dump-файл genericmailsql.sql

**Листинг 7.1:** Инсталляция и настройка MySQL

```
# emerge mysql
# /usr/bin/mysql_install_db
(далее следуйте указаниям на экране, для добавления пароля root в
mysql, не mysqladmin, иначе ваша база будет открыта всем ветрам )
```

```
# /etc/init.d/mysql start
# mysqladmin -u root -p create mailsql
# mysql -u root -p mailsql < genericmailsql.sql
```

```
# mysql -u root -p mysql
mysql> GRANT SELECT,INSERT,UPDATE,DELETE
-> ON mailsql.*
-> TO mailsql@localhost
-> IDENTIFIED BY '$password';

-> quit
```

(проверьте, что новый mailsql пользователь, может подключаться к mysql серверу )

```
# mysql -u mailsql -p mailsql
```

Ваша новая база имеет значения по умолчанию и таблицы для двух доменов. Вложены следующие таблицы:

- **alias** — локальные e-mail и информация о псевдонимах для mailman
- **relocated** — информация о адреса перемещенных пользователей
- **transport** — информация о почтовых транспортах для всех доменов размещающихся у вас
- **users** — информация о всех пользователях
- **virtual** — информация о псевдонимах для виртуальных доменов

**Листинг 7.2:** пример таблицы псевдонимов

```
id alias destination
1 root foo@bar.com
2 postmaster foo@bar.com
```

**Листинг 7.3:** пример таблицы пользователей

(приведем для ясности строку)

```
id email clear name uid gid homedir \
maildir quota postfix
10 foo@virt-bar.org $password realname virtid virtid /home/vmail \
/home/vmail/virt-bar.org/foo/.maildir/ y
13 foo@bar.com $password realname localid localid /home/foo \
/home/foo/.maildir/ y
```

**Листинг 7.4:** пример таблицы транспортов

```
id domain destination
1 bar.com local:
2 virt-bar.org virtual:
```

**Листинг 7.5:** пример таблицы псевдонимов для виртуальных доменов

```
id email destination
3 root@virt-bar.org other@email.address
```

## 8. Apache и phpMyAdmin

Итак, следующим шагом мы настроим apache и создадим интерфейс для еще более простого взаимодействия с базой данных.

**Листинг 8.1:** Настройка apache и phpmyadmin

```
# emerge apache mod_php phpmyadmin
```

Существует множество руководств по настройке apache с поддержкой php. Например, <http://www.linuxguruz.org/z.php?id=31>. Также многочисленные сообщения на <http://forums.gentoo.org> где рассматриваются решения проблем возникших в ходе инсталляции (поиск по "apache php"). Итак, я не стараюсь раскрыть эту тему здесь. Настройте apache и php, затем продолжим вместе настройку. Теперь слово для умных: .htaccess положите в директорию к phpmyadmin. Если вы не сделаете этого, поисковые системы проиндексируют страницы phpmyadmin и каждый сможет

получить к нему доступ с помощью google, и изменить ваши базы, что не есть хорошо. Существует много howto как это сделать. <http://docs.csoft.net/micro/blackhtaccess.html>.

Теперь мы переходим к инсталляции сертификатов для apache, созданных нами ранее. Директивы apache которые вам требуется изменить для этого:

```
SSLCertificateFile /path/to/certs/new.cert.cert
```

```
SSLCertificateKeyFile /path/to/certs/new.cert.key
```

**Листинг 8.2:** Инсталляция Apache SSL сертификатов

```
# cp /etc/ssl/misc/new.cert.cert /etc/apache/conf/ssl/
```

```
# cp /etc/ssl/misc/new.cert.key /etc/apache/conf/ssl/
```

```
# nano -w /etc/apache/conf/vhosts/ssl.default-vhost.conf
```

(измените следующие параметры)

```
ServerName host.domain.name
```

```
ServerAdmin your@email.address
```

```
SSLCertificateFile /etc/apache/conf/ssl/new.cert.cert
```

```
SSLCertificateKeyFile /etc/apache/conf/ssl/new.cert.key
```

```
# /etc/init.d/apache restart
```

**Примечание:** Если у вас apache уже проинсталлирован, вам вероятно придется сделать перезагрузку сервера. Проверьте логи системы на предмет корректного запуска apache.

Дальше настраиваем phpMyAdmin.

**Листинг 8.3:** Настройка phpMyAdmin

```
# nano -w /home/httpd/htdocs/phpmyadmin/config.inc.php
```

(измените следующие параметры)

```
$cfg['Servers'][$i]['host'] = 'localhost'; // MySQL hostname
```

```
$cfg['Servers'][$i]['controluser'] = 'mailsql'; // MySQL настройки системного аккаунта  
// (этот аккаунт должен иметь read-only
```

```
$cfg['Servers'][$i]['controlpass'] = '$password'; // доступ к таблицам "mysql/user"  
// и "mysql/db" tables)
```

```
$cfg['Servers'][$i]['user'] = 'mailsql'; // MySQL пользователь
```

```
$cfg['Servers'][$i]['password'] = '$password'; // MySQL пароль
```

Теперь введите адрес phpmyadmin страницы и просмотрите ваши таблицы в базе. Вы можете добавить локальные псевдонимы, поправить таблицу пользователей и добавить тестового пользователя, изменить таблицу транспортов для добавления информации о ваших доменах. Значений по умолчанию которые устанавливаются вместе с dump-файлом должно быть достаточно для примера, чтобы помочь вам настроить систему. Убедитесь что ввели в базу корректную информацию. Для примера, убедитесь что директории локальных пользователей существуют и указаны корректные uid/gid. Почтовые директории пользователей, должны быть созданы postfix, при первой принятой почте для пользователя. Будет неплохо, если вы пошлете "Добро пожаловать!", чтобы убедиться, что .maildir создан.

## 9. Vmail-пользователь

Здесь вы можете быть удивлены, узнав, что для виртуальных акаунтов,

используются пользователи и директории, и будете правы.

**Листинг 9.1:** Adding the vmail user

```
# adduser -d /home/vmail -s /bin/false vmail
# uid=`cat /etc/passwd | grep vmail | cut -f 3 -d :`
# groupadd -g $uid vmail
# mkdir /home/vmail
# chown vmail. /home/vmail
```

Теперь, когда вы настраиваете виртуальный акаунт, используйте vmail uid, gid , и его домашнюю папку. Когда вы создаете локальный акаунт, используйте uid, gid и домашнюю папку нового пользователя, а не vmail. Мы имеем ввиду, что если вы захотите создать php-страничку для администрирования пользователей, не забывайте, что phpmyadmin в целом справляется с этой работой очень неплохо.

## 10. Настройка MySQL авторизации и виртуальных доменов

Дальше мы перенастроим нашу авторизацию, на использование mailsqldb базы в courier-imap и postfix. Во всех следующих примерах, замените \$password паролем, который вы задали пользователю mailsqldb для mysql.

**Листинг 10.1:**

```
# emerge /usr/portage/sys-libs/pam_mysql/pam_mysql-$currentversion.ebuild
(этот пакет здесь задан маской, которую вы должны заменить на текущую
версию пакета. версию пакета вы можете посмотреть в portage )
```

```
# nano -w /etc/pam.d/imap
```

(закомментируйте существующие строки настройки авторизации, и добавьте указанные ниже)

```
#auth    required    pam_nologin.so
#auth    required    pam_stack.so service=system-auth
#account  required    pam_stack.so service=system-auth
#session  required    pam_stack.so service=system-auth

auth    optional    pam_mysql.so host=localhost db=mailsqldb user=mailsqldb \
passwd=$password table=users usercolumn=email passwdcolumn=clear crypt=0
account required    pam_mysql.so host=localhost db=mailsqldb user=mailsqldb \
passwd=$password table=users usercolumn=email passwdcolumn=clear crypt=0
```

```
# nano -w /etc/pam.d/pop3
```

```
# nano -w /etc/pam.d/smtp
```

(сделайте такие же изменения в pop3 и smtp файлах)

Далее нам нужно поправить конфигурацию авторизации courier.

**Листинг 10.2:**

```
# nano -w /etc/courier-imap/authdaemonrc
authmodulelist="authmysql authpam"
```

```
# nano -w /etc/courier-imap/authdaemond.conf
```

```
AUTHDAEMOND="authdaemond.mysql"
```

```
# nano -w /etc/courier-imap/authmysqlrc
```

```
MYSQL_SERVER      localhost
MYSQL_USERNAME     mailsqldb
MYSQL_PASSWORD     $password
```

```

MYSQL_DATABASE      mailsql
MYSQL_USER_TABLE    users
#MYSQL_CRYPT_PWFIELD crypt (эта строка должна быть закомментирована)
MYSQL_CLEAR_PWFIELD clear
MYSQL_UID_FIELD      uid
MYSQL_GID_FIELD      gid
MYSQL_LOGIN_FIELD    email
MYSQL_HOME_FIELD     homedir
MYSQL_NAME_FIELD     name
MYSQL_MAILDIR_FIELD  maildir

```

```

# /etc/init.d/authdaemond restart
# /etc/init.d/saslauthd restart

```

Мы уже почти подошли к тому что вам обещал. Далее мы настроим необходимые конфиги postfix'a для связки с базой данных для всех необходимых необходимых транспортных.

**Листинг 10.3:** /etc/postfix/mysql-aliases.cf  
**# nano -w /etc/postfix/mysql-aliases.cf**  
**# mysql-aliases.cf**

```

user      = mailsql
password  = $password
dbname    = mailsql
table     = alias
select_field = destination
where_field = alias
hosts     = unix:/var/run/mysqld/mysqld.sock

```

**Листинг 10.4:** /etc/postfix/mysql-relocated.cf  
**# nano -w /etc/postfix/mysql-relocated.cf**  
**# mysql-relocated.cf**

```

user      = mailsql
password  = $password
dbname    = mailsql
table     = relocated
select_field = destination
where_field = email
hosts     = unix:/var/run/mysqld/mysqld.sock

```

**Листинг 10.5:** /etc/postfix/mysql-transport.cf (необязательно)  
**# nano -w /etc/postfix/mysql-transport.cf**  
**# mysql-transport.cf**

```

user      = mailsql
password  = $password
dbname    = mailsql
table     = transport
select_field = destination
where_field = domain
hosts     = unix:/var/run/mysqld/mysqld.sock

```

**Листинг 10.6:** /etc/postfix/mysql-virtual-gid.cf (необязательно)  
**# nano -w /etc/postfix/mysql-virtual-gid.cf**  
**#mysql-virtual-gid.cf**

```

user      = mailsql

```

```
password      = $password
dbname        = mailsql
table         = users
select_field  = gid
where_field   = email
additional_conditions = and postfix = 'y'
hosts         = unix:/var/run/mysqld/mysqld.sock
```

**Листинг 10.7:** /etc/postfix/mysql-virtual-maps.cf

```
# nano -w /etc/postfix/mysql-virtual-maps.cf
# mysql-virtual-maps.cf
```

```
user          = mailsql
password      = $password
dbname        = mailsql
table         = users
select_field  = maildir
where_field   = email
additional_conditions = and postfix = 'y'
hosts         = unix:/var/run/mysqld/mysqld.sock
```

**Листинг 10.8:** /etc/postfix/mysql-virtual-uid.cf (необязательно)

```
# nano -w /etc/postfix/mysql-virtual-uid.cf
# mysql-virtual-uid.cf
```

```
user          = mailsql
password      = $password
dbname        = mailsql
table         = users
select_field  = uid
where_field   = email
additional_conditions = and postfix = 'y'
hosts         = unix:/var/run/mysqld/mysqld.sock
```

**Листинг 10.9:** /etc/postfix/mysql-virtual.cf

```
# nano -w /etc/postfix/mysql-virtual.cf
# mysql-virtual.cf
```

```
user          = mailsql
password      = $password
dbname        = mailsql
table         = virtual
select_field  = destination
where_field   = email
hosts         = unix:/var/run/mysqld/mysqld.sock
```

И последнее, правим /etc/postfix/main.cf еще один раз.

**Листинг 10.10:** /etc/postfix/main.cf

```
# nano -w /etc/postfix/main.cf
alias_maps = mysql:/etc/postfix/mysql-aliases.cf
relocated_maps = mysql:/etc/postfix/mysql-relocated.cf
```

```
local_transport = local
local_recipient_maps = $alias_maps $virtual_mailbox_maps unix:passwd.byname
```

```
virtual_transport = virtual
virtual_mailbox_domains =
    virt-bar.com,
```

\$other-virtual-domain.com

```
virtual_minimum_uid = 1000
virtual_gid_maps = static:$vmail-gid
virtual_mailbox_maps = mysql:/etc/postfix/mysql-virtual-maps.cf
virtual_alias_maps = mysql:/etc/postfix/mysql-virtual.cf
virtual_uid_maps = static:$vmail-uid
virtual_mailbox_base = /
#virtual_mailbox_limit =
```

Здесь видно чем значительно отличается postfix 2.0.x от 1.1.x. Сильнее всего заметно отсутствие необходимости в таблицах транспорта, virtual-gid и virtual-uid, хотя эти таблицы и вкладываются, чтобы вы могли их использовать при необходимости.

**Примечание:** Рекомендуем прочесть VIRTUAL\_README идущий с postfix, для большего количества информации.

**Листинг 10.11:**  
**# postfix reload**

Теперь, если все прошло хорошо, вы должны иметь работающий почтовый сервер. Пользователи должны быть способны авторизоваться в sql базе, использовать свой полный почтовый адрес для pop3, imap, и smtp. Настоятельно рекомендую проверить, что всё это действительно работает. Если вы столкнулись с проблемами, проверьте раздел troubleshooting в конце документа.

## 11. Squirrelmail

**Листинг 11.1:**  
**# emerge squirrelmail**  
(добавим ссылку на htdocs для более короткого пути)  
**# ln -s /home/httpd/htdocs/squirrelmail/ /home/httpd/htdocs/mail**  
**# cd /home/httpd/htdocs/mail/conf**  
**# ./conf.pl**

(измените настройки Organization, Server, и Folder для squirrelmail)  
(теперь вы имеете возможность залогиниться в squirrelmail, снова с вашим полным email адресом, и использовать ваш новый webmail setup)

## 12. Mailman

Последний шаг — mailman. Новая версия имеет великолепную поддержку виртуальных доменов, поэтому я использую его, не говоря уже о том что это просто великолепная программа. Настоятельно рекомендую прочесть документацию mailman, включающую README.POSTFIX.gz, для более полного понимания.

Одно замечание, текущая версия mailman устанавливается в /usr/local/mailman. Если вы хотите изменить директорию инсталляции, вы можете изменить в ebuild файле переменную INSTALLDIR.



**Листинг 12.1:** /usr/portage/net-mail/mailman/mailman-\$ver.ebuild  
**# nano -w /usr/portage/net-mail/mailman/mailman-\$ver.ebuild**  
MAILGID="280"  
(задайте MAILGID для группы mailman взамен nobody)  
(это необходимо для интеграции с postfix)

**Листинг 12.2:**

**# emerge mailman**

(имя этого пакета маскировано, поэтому вам надо указать emerge путь к ebuild. После инсталляции, следуйте инструкциям в README.gentoo.gz \*исключая\* — не добавляйте псевдонимы в /etc/mail/aliases. Вместо этого, мы укажем postfix использовать псевдонимы из базы)

**# zless /usr/share/doc/mailman-\$ver/README.gentoo.gz**

**Листинг 12.3:** Настройки по умолчанию: Mailman/Defaults.py

**# nano -w /var/mailman/Mailman/Defaults.py**

(измените значения приведенные ниже на ваши, дальше будем настраивать виртуальные домены)

DEFAULT\_EMAIL\_HOST = 'domain.com'

DEFAULT\_URL\_HOST = 'www.domain.com'

**Листинг 12.4:** настройка mailman: mm\_cfg.py

**# nano -w /var/mailman/Mailman/mm\_cfg.py**

MTA = "Postfix"

POSTFIX\_STYLE\_VIRTUAL\_DOMAINS = ['virt-domain.com', 'virt.domain2.com']

add\_virtualhost('www.virt.domain.com', 'virt.domain.com')

add\_virtualhost('www.virt.domain2.com', 'virt.domain2.com')

(это нужно для работы mailman с виртуальными доменами)

**Листинг 12.5:**

(создадим свой первый список рассылки)

**# su mailman**

**# cd ~**

**# bin/newlist test**

Enter the email of the person running the list: your@email.address

Initial test password:

Hit enter to continue with test owner notification...

(листы виртуального домена могут быть указаны в виде list@domain.com)

**# bin/genaliases**

(теперь ваши псевдонимы сгенерированы, проверьте, что они добавлены правильно)

**# nano -w data/aliases**

**# STANZA START: test**

**# CREATED:**

test: "/var/mailman/mail/mailman post test"

test-admin: "/var/mailman/mail/mailman admin test"

test-bounces: "/var/mailman/mail/mailman bounces test"

test-confirm: "/var/mailman/mail/mailman confirm test"

test-join: "/var/mailman/mail/mailman join test"

test-leave: "/var/mailman/mail/mailman leave test"

test-owner: "/var/mailman/mail/mailman owner test"

test-request: "/var/mailman/mail/mailman request test"

test-subscribe: "/var/mailman/mail/mailman subscribe test"

test-unsubscribe: "/var/mailman/mail/mailman unsubscribe test"

**# STANZA END: test**

**# /etc/init.d/mailman start**

### **# rc-update add mailman default**

(для запуска mailman при каждой загрузке)

**Листинг 12.6:** Добавление поддержки псевдонимов mailman в postfix

#### **# nano -w /etc/postfix/main.cf**

```
owner_request_special = no
```

```
recipient_delimiter = +
```

(прочтите README.POSTFIX.gz для более детальной информации)

```
alias_maps =
```

```
    hash:/var/mailman/data/aliases,
```

```
    mysql:/etc/postfix/mysql-aliases.cf
```

```
virtual_alias_maps =
```

```
    hash:/var/mailman/data/virtual-mailman,
```

```
    mysql:/etc/postfix/mysql-virtual.cf
```

(это добавит поддержку файлов псевдонимов mailman'a в postfix, конечно

вы можете использовать таблицу в mysql для этого, но я ненавижу делать это руками.

Также если вы не используете виртуальных доменов, добавление псевдонимов, может привести к проблемам)

Теперь вы можете создавать листы рассылок для каждого домена в вашей системе. Последнее замечание, убедитесь, что mailman запущен из под пользователя mailman (su mailman) иначе у вас будут проблемы с разрешениями. прочтите документацию mailman для более детальной информации по управлению листами рассылки.

## **13. Фильтрация содержимого и Anti-Virus**

Скоро будет,... это уже существует, но я должен еще немного разобраться в perl'e и протестировать результат. Если вы хотите помочь с этим, свяжитесь со мной.

## **14. Окончание**

Итак,. Вы все сделали, теперь поправьте /etc/postfix/master.cf и отключите режим verbose. Возможно вы захотите доавать сервис в автозагрузку. Убедитесь что вы добавили в автозагрузку все сервисы которые используете - apache, mysql, saslauthd, postfix, courier-imapd, courier-imapd-ssl, courier-pop3d, и courier-pop3d-ssl, все зависит от вашего решения какие сервисы предоставлять. Обычно я разрешаю все сервисы.

**Листинг 14.1:** Окончание

```
# postfix reload
```

```
# rc-update add $service default
```

Вобщем инджой и хэв фан!

## 15. Troubleshooting

### Введение

Troubleshooting: Это небольшой список решений для проблем, наиболее часто возникающих в ходе установки. Это не исчерпывающая информация, но вы можете начать поиск решения граблей отсюда. Такая запутанная настройка и инсталляция как в этом документе, может привести к тому что некоторые компоненты будут работать неправильно. Вообще советую сделать несколько следующих шагов. Запустите систему в базовой конфигурации, добавляя компонент за компонентом, чтобы выяснить какой из них сбоит.

### Шаг 1: Проверьте ваши файлы конфигурации.

Опечатки — главный враг, особенно когда идет речь о системе авторизации. Проверьте ваши конфигурационные файлы и базу данных на предмет опечаток. Если вы делаете изменения настроек для сервиса, убедитесь, что вы перезапустили сервиса, дабы изменения вступили в силу.

#### Листинг 15.1:

```
# /etc/init.d/service restart
```

### Шаг 2: Все необходимые сервисы запущены и работают?

Если не запущены, запустите. Крайне сложно отлаживать программу если она не запущена. Иногда сервис запускается, но не функционирует. Иногда, когда используется неправильный файл конфигурации сервис может использовать другой порт предназначенный другому процессу. Иногда вы можете воспользоваться `netstat`. Или если возможно, перезагрузите ваш сервер, это очистит любой зависший сервис.

#### Листинг 15.2:

```
# /etc/init.d/$service status  
# netstat -a | grep $service (или $port)
```

### Шаг 3: Все сервисы используют правильные файлы конфигурации?

Если вы внесли изменения в конфигурационные файлы сервиса, убедитесь что он использует новые параметры. Некоторые программы, вроде postfix могут показывать конфигурацию их параметров.

#### Листинг 15.3:

```
# apachectl fullstatus (необходим lynx)  
# apachectl configtest (проверка нормальной конфигурации)  
# postconf -n (показывает текущие параметры используемые postfix)  
# /etc/init.d/$service restart
```

### Шаг 4: Проверьте логи.

На то логи и существуют, чтобы помочь нам выяснить причину, а иногда и

подсказать решение проблемы. Просмотрите все логи, которые имеют отношение к программе, ее сбою.

#### **Листинг 15.4:**

```
# kill -USR1 `ps -C metalog -o pid=` (для выключения metalog buffering)
# nano -w /var/log/mail/current
# cat /var/log/mysql/mysql.log
# tail /var/log/apache/error_log
```

Вы также можете найти параметр `debug_peer` в `main.cf` который может быть полезен. Включение этого параметра, увеличит подробность выводящихся сообщений в логи.

#### **Листинг 15.5:** добавление поддержки `debug_peer`

```
# nano -w /etc/postfix/main.cf
debug_peer_level = 5
debug_peer_list = $host.domain.name
(раскомментируйте одну из нужных строк)
```

### **Шаг 5: Попробуйте связаться с сервисом.**

SMTP, IMAP, и POP3 отвечают на telnet сессию, как вы видели немного раньше при конфигурации postfix. Иногда помогает открыть telnet сессию с сервисом, для того чтобы выяснить что случилось.

#### **Листинг 15.6:**

```
# telnet localhost $port
(SMTP на 25-м, IMAP, на 143-м, POP3 на 110-м порту. по крайней мере вы
должны увидеть слово OK. Так вы будете знать что сервис запущен, и готов к ответу.)
```

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
* OK Courier-IMAP ready. Copyright 1998-2002 Double Precision, Inc.
```

### **Шаг 6: Иногда только большая пушка может дать нам информацию: strace.**

Вы должны иметь эту программу всегда под рукой. Это бесценный инструмент для отлаживания программ. Вы можете запустить из командной строки `strace` и наблюдать все системные вызовы происходящие в программе. Часто дампы содержит огромное количество информации, так что вы можете либо наблюдать в реальном времени, за транзакцией приводящей к краху программы, или записать его в файл, для неторопливого просмотра позже.

#### **Листинг 15.7:**

```
# emerge strace
# strace $command
# strace -p `ps -C $service -o pid=`
```

# Виртуальная почтовая система на основе qmail/vpopmail.

Ссылка на оригинал: <http://www.gentoo.org/doc/ru/qmail-howto.xml>

С версии: 1.0

## 1. Введение

Не важно для чего вы хотите организовать систему электронной почты, только лишь для сервисов, как самостоятельный сервер, или для нескольких виртуальных доменов, qmail может быть легко настроен для любых ваших нужд. Это руководство поможет вам установить и настроить qmail, для любого из этих вариантов с уклоном на удаленный доступ, в том числе через зашифрованные каналы данных.

Это руководство поможет вам освоить qmail, courier-imap, vpopmail, и horde/imp. Но вместе с ними так же вступят в "игру" daemontools, ucspi-tcp, mysql, apache, и mod\_php. qmail предоставляет функции MTA (Mail Transfer Agent), courier-imap - предоставляет удаленный сбор почты, (pop3/imap сервер. прим.пер.) vpopmail предоставляет функции управления виртуальными доменами, и horde/imp - это почтовый веб-клиент.

До установки (emerging) чего-либо, вы должны корректно установить переменную USE. Если у вас уже были установлены какие-нибудь из этих пакетов, то, возможно, вам придется их переустановить. USE="maildir ssl imap mysql". Также, если вы хотите использовать horde/imp как web-клиент, то так же надо добавить USE="nls" до установки mod\_php.

**Примечание:** Это руководство написано "по-шагам" Если в какой-то момент вы почувствуете что все то, что нужно готово, то можете не продолжать.

Существует достаточно пакетов, с помощью которых можно построить систему электронной почты. Сейчас самое время осмыслить и решить для вас ли qmail. У нас так же есть прекрасное руководство основанное на Postfix, или вы можете взглянуть на exim. Выбирайте решение для себя, а мы покажем вам, как использовать qmail.

## 2. qmail ("Разговаривая с собой")

Листинг 2.1: Emerge qmail  
`# emerge qmail`

**Важно:** Эта часть построена вокруг qmail-1.03-r13 или старше. Будет ли нижеизложенное работать с ранними версиями? Может быть. Должны ли вы обновиться? Да, если вы хотите быть уверены, что это сработает.

**Предупреждение:** Если вы получили сообщение the virtual/mta package conflicts with another package тогда вам необходимо удалить из системы любой другой MTA. Чтобы узнать что это за пакет, просто запустите emerge qmail -p.

Установка qmail так же "потянет" за собой установку ucspi-tcp и daemontools. О них, если хотите, вы можете прочитать на ucspi-tcp и на daemontools. Обычно, daemontools отвечает за управление qmail'ом как сервисом, тогда как ucspi-tcp отвечает за управление входящими TCP соединениями для сервиса qmail.

Для начала проведем несколько пост-инсталляционных мероприятий. :)

**Листинг 2.2:** Конфигурирование qmail

(Измените под свою конфигурацию)

```
# nano /var/qmail/control/servercert.cnf
# ebuild /var/db/pkg/mail-mta/qmail-1.03-r*/qmail-1.03-r*.ebuild config
```

Qmail проектировался так, чтобы полностью соответствовать требованиям безопасности. Почта никогда не посылается пользователю 'root'. Поэтому вам надо выбрать пользователя на машине, который будет получать почту предназначенную 'root'. С этого момента я предполагаю, что этот пользователь 'vapier'.

**Листинг 2.3:** Установка не-root акаунта для e-mail.

```
# cd /var/qmail/alias
# echo vapier > .qmail-root
# echo vapier > .qmail-postmaster
# echo vapier > .qmail-mailer-daemon
```

Теперь, поднимем сервис доставки qmail.

**Листинг 2.4:** Запуск qmail delivery service

```
# rc-update add svscan default
# /etc/init.d/svscan start
# cd /service
# ln -s /var/qmail/supervise/qmail-send qmail-send
```

Надо убедиться, что qmail работает корректно, вот небольшой тест.

**Листинг 2.5:** Test delivery service

```
# ssh vapier@localhost
# maildirmake .maildir
# qmail-inject root << EOF
test root e-mail!
EOF
# qmail-inject postmaster << EOF
test postmaster e-mail!
EOF
# qmail-inject vapier << EOF
test vapier e-mail!
EOF
# mutt
```

(Теперь в вашем ящике должно быть 3 письма)

Вот и все! Теперь у вас стоит почтовая система способная удовлетворить, потребности локальной машины и системных демонов/пользователей.

Предупреждение: Если вы не получили никаких писем или видите странные ошибки в лог-файлах (проверьте /var/log/qmail/) включающие 'localhost.localhost', это означает что ваша domain/dns информация не корректна. По умолчанию, qmail использует информацию от hostname --fqdn. Если на вашей машине, эта команда выводит 'localhost', тогда проверьте ваши файлы /etc/hostname, /etc/hosts, и настройки dns чтобы убедиться что все правильно. Отредактируйте конфигурационные файлы в каталоге /var/qmail/control/. Используйте их примеры, (там же) если вам нужна помощь.

**Листинг 2.6:** Пример /var/qmail/control/ файлов для домена второго уровня

```
# hostname --fqdn
wh0rd.org
# cat me
wh0rd.org
# cat defaultdomain
wh0rd.org
# cat plusdomain
wh0rd.org
# cat locals
wh0rd.org
# cat rcpthosts
wh0rd.org
```

**Листинг 2.7:** Пример /var/qmail/control/ файлов для домена третьего уровня

```
# hostname --fqdn
mail.wh0rd.org
# cat me
mail.wh0rd.org
# cat defaultdomain
wh0rd.org
# cat plusdomain
wh0rd.org
# cat locals
mail.wh0rd.org
# cat rcpthosts
mail.wh0rd.org
```

### 3. *vrpmail*

**Листинг 3.1:** Emerge vrpmail

```
# emerge vrpmail
```

**Важно:** Эта часть построена вокруг vrpmail-5.4.6 или старше. Будет ли нижеизложенное работать с ранними версиями? Может быть. Должны ли вы обновиться? Да, если вы хотите быть уверены что это сработает.

**vrpmail** требует немного больше старания при установке, чем предыдущие пакеты. Мы должны убедиться что он запущен и работает (до mysql). Далее мы можем установить базу данных vrpmail. До того как приступить к следующему шагу, необходимо проверить что mysql корректно установлен. Учтите, пароль для базы

данных vpopmail будет 'vpoprw', конечно вы можете выбрать другой.

**Листинг 3.2:** Установка vpopmail в mysql

```
# rc-update add mysql default
```

Если вы установили mysql в первый раз, убедитесь что была запущена команда  
ebuild <mysql.ebuild> config до запуска mysql сервера.

```
# /etc/init.d/mysql start
```

```
# nano /etc/vpopmail.conf
```

(Изменим пароль с 'secret' на 'vpoprw')

```
# mysql -p << EOF
```

```
create database vpopmail;
```

```
use mysql;
```

```
grant select, insert, update, delete, create, drop on vpopmail.* to vpopmail@localhost identified by 'vpoprw';
```

```
flush privileges;
```

```
EOF
```

(Следующие шаги могут понадобиться, а могут и нет, но мы выполним их просто чтобы быть уверенными)

```
# chown root:vpopmail /etc/vpopmail.conf
```

```
# chmod 640 /etc/vpopmail.conf
```

```
# chown root:vpopmail /var/vpopmail/bin/vchkpw
```

```
# chmod 4711 /var/vpopmail/bin/vchkpw
```

**Предупреждение:** Если вы испытываете трудности с доступом в mysql/vpopmail, перезапустите mysql. Выполните /etc/init.d/mysql restart.

Теперь, vpopmail готов к использованию. В этой части, мы предоставим виртуальный хостинг для домена 'wh0rd.org'. Это означает что мы должны "сообщить" vpopmail об этом домене. Кстати, давайте добавим пользователя 'vapier'.

**Листинг 3.3:** Добавление домена

(это нужно сделать только в том случае если vadddomain выдает "command not found" (см. далее))

```
# source /etc/profile
```

(при отладке vpopmail, вам может понадобиться просмотр журналов)

```
# mysql -u vpopmail -p
```

```
mysql> select * from vpopmail.vlog;
```

```
# vadddomain wh0rd.org postpass
```

(теперь проверим, правильно ли установлен домен)

```
# printf "postmaster@wh0rd.org\0postpass\0blah\0" | vchkpw `which id` 3<&0
```

```
uid=89(vpopmail) gid=89(vpopmail) groups=0(root)
```

(если вы не видите что-либо подобное, значит права доступа где-то выставлены некорректно)

```
# vadduser vapier@wh0rd.org vappw
```

В каждом домене созданном vpopmail существует аккаунт 'postmaster'. Мы задаем пароль postmaster'а в 'postpass'. Чтобы vpopmail был действительно полезен нам, мы должны получать почту через courier и посылать с помощью qmail и SMTP.

## 4. Courier POP/IMAP

**Листинг 4.1:** Emerge courier-imap

```
# emerge net-mail/courier-imap
```

**Важно:** Сразу после vpopmail вы должны установить courier-imap Модуль authvchkpw будет встроен.(?)

**Важно:** Эта часть построена вокруг net-mail/courier-imap-3.0.7 или старше. Будет ли нижеизложенное работать с ранними версиями? Может быть. Должны ли вы



обновиться? Да, если вы хотите быть уверены что это сработает.

И еще несколько пост-инсталляционных шагов:). Они необходимы только в том случае если вы будете использовать SSL зашифрованные коммуникации (Рекомендуется!). Иначе вы можете пропустить последние 2 шага, в следующих листингах, удаляя '-ssl' из имени инициализационного скрипта.

**Листинг 4.2:** Конфигурирование POP3/SSL

```
# nano /etc/courier/authlib/authdaemonrc
(Оставьте в переменной authmodulelist только "authvchkpw")
# cd /etc/courier-imap
# nano pop3d.cnf
(Отредактируйте секцию [ req_dn ] )
# mkpop3dcert
# rc-update add courier-pop3d-ssl default
# /etc/init.d/courier-pop3d-ssl start
```

**Листинг 4.3:** Конфигурирование IMAP/SSL

```
# cd /etc/courier-imap
# nano imapd.cnf
(Отредактируйте секцию [ req_dn ] )
# mkimapdcert
# rc-update add courier-imapd-ssl default
# /etc/init.d/courier-imapd-ssl start
```

Ваш почтовый клиент теперь может соединиться с сервером (с запущенным на нем courier) и получать почту предназначенную для виртуального хоста. В моем случае, я могу соединиться с именем 'vapier@wh0rd.org' и паролем 'vappw'.

## 5. *qmail (разговаривая с миром)*

Давайте заставим SMTP работать, не причинив ущерба людям, открыв очередную спам-дыру.

**Листинг 5.1:** Запуск qmail SMTP service

```
# cd /var/qmail/control/
# nano conf-smtpd
(Раскомментируйте SMTP-AUTH переменную и установите QMAIL_SMTP_CHECKPASSWORD
 в /var/vpopmail/bin/vchkpw)
# nano servercert.cnf
(Отредактируйте секцию [ req_dn ] )
# mkservercert
# cd /service
# ln -s /var/qmail/supervise/qmail-smtpd qmail-smtpd
# /etc/init.d/svscan restart
```

Предположим вы не меняли файлов qmail, тогда он теперь принимает почту для виртуального домена wh0rd.org и для пользователей локальной машины. Более того, qmail доверяет любому кто посылает письма через 127.0.0.1 и для любого кто может пройти аутентификацию через vpopmail. При настройке своего почтового клиента для отправки почты, убедитесь что вы включили опцию вроде 'Server

requires authentication'. В моем случае, я поставил имя пользователя в 'vapier@wh0rd.org' и пароль - 'vappw'. И последняя деталь - убедитесь что настроили использование SSL/TLS для SMTP в своем клиенте. Qmail не позволит вам пройти аутентификацию если сессия не зашифрована.

## 6. Веб клиент Horde / IMP

Несмотря на то что существует множество веб-клиентов электронной почты (и вы свободны выбрать любой из них), я предпочитаю IMP Webmail Client который является частью "окружения Horde". Основная причина заключается в том что Horde предоставляет достаточно прозрачный web доступ к почте, и вы можете очень просто добавить другие компоненты, такие как Адресные Книги, Календари, Задачи, и др. И если он до сих пор вас не устраивает, тогда вам следует прочитать Horde.

On to the good stuff! Теперь установим IMP.

**Листинг 6.1:** Emerge IMP  
**# emerge horde-imp**

**Важно:** Эта часть построена вокруг horde-2.2.x and horde-imp-3.2.x. CVS версии (и следующие релизы) скорее всего, будут очень различаться в настройках. Так же имейте ввиду php-5.x, скорее всего, нарушит работу horde-2.x.

Horde не просто настроить если вы точно не знаете что делать. К счастью для вас, я точно знаю что делать, значит и вы будете. :)

**Листинг 6.2:** Конфигурирование Horde  
**# cd /var/www/localhost/htdocs/horde/config/**  
**# for f in \*.dist ; do mv \${f} \${f}.dist ; done**  
**# nano horde.php**  
(Ниже 'Horde Authentication':)  
\$conf['auth']['driver'] = 'imap';  
\$conf['auth']['params']['dsn'] = '{localhost:993/imap/ssl/novalidate-cert}';

(Ниже 'Horde Logging':)  
\$conf['log']['name'] = '/var/log/apache2/horde.log';

(Ниже 'Problem Reporting':)  
\$conf['problems']['enabled'] = true;  
\$conf['problems']['email'] = 'webmaster@wh0rd.org';

**# nano registry.php**  
(Ниже 'Handlers':)  
\$this->registry['auth']['login'] = 'imp';  
\$this->registry['auth']['logout'] = 'imp';

(Ниже 'Application registry':) Установите элемент 'status', раздел applications['imp'], из 'inactive' в 'active'

**# touch /var/log/apache2/horde.log**  
**# chown apache:apache /var/log/apache2/horde.log**

**Примечание:** Вы можете поменять аккаунт 'webmaster' на что-нибудь другое. Если вы хотите использовать этот e-mail адрес, тогда вы должны добавить такого пользователя в vpopmail. (или смотрите ниже о qmailadmin).

Теперь действительно быстро установим IMP.

**Листинг 6.3:** Конфигурирование IMP

```
# cd /var/www/localhost/htdocs/horde/imp/config/  
# for f in *.dist ; do mv ${f} ${f}.dist ; done  
# nano servers.php
```

```
(Отредактируйте массив $servers['imap']  
$servers['imap'] = array(  
    'name' => 'wh0rd.org',  
    'server' => 'localhost',  
    'protocol' => 'imap/ssl/novalidate-cert',  
    'port' => 993,  
    'folders' => "",  
    'namespace' => 'INBOX.',  
    'maildomain' => 'wh0rd.org',  
    'smtphost' => 'localhost',  
    'realm' => "",  
    'preferred' => ""  
);
```

Наконец, поднимем apache, чтобы начать использовать webmail.

**Листинг 6.4:** Запуск apache

```
# nano /etc/conf.d/apache2  
(Раскомментируйте APACHE2_OPTS="-D SSL -D PHP4")  
# rc-update add apache2 default  
# /etc/init.d/apache2 start
```

**Предупреждение:** Вы должны подумать о том, чтобы заставить работать Horde через https. Это не простая задача, но приглядитесь к ней ради безопасности.

Чтобы проверить установку IMP, запустите браузер и зайдите <http://localhost/horde/> (измените localhost на ваш сервер) Вы должны увидеть страницу приглашения Horde где вы можете войти. Опять же, в моей установке я вхожу с помощью 'varier@wh0rd.org' и 'vappw' соответственно как логин и пароль.

На данный момент, Horde и IMP установлены. Однако, вы можете пройти заново через все дерево конфигурационных директорий и настроить все что вашей душе угодно.

## 7. Дополнительные пакеты

### qmailadmin

Первый пакет на который я обращаю ваше внимание это qmailadmin. Это веб интерфейс для управления виртуальными доменами. Просто emerge net-mail/qmailadmin а затем укажите вашему браузеру на <http://localhost/cgi-bin/qmailadmin> чтобы начать использовать его. qmailadmin делает жизнь намного проще.

### qmHandle

Если вы вляпались в проблемы с очередями qmail и провели много времени над отладкой, то, возможно захотите взглянуть на qmHandle. Это простая перл

программа которая позволяет вам видеть и редактировать очередь сообщений qmail. Опять таки все что нужно сделать это emerge net-mail/qmhandle.

## Дополнения к horde

Я очень рекомендую посмотреть на множество приложений Horde. Turba, Kronolith, и Nag эти приложения очень хорошо дополняют IMP для разных случаев. Их настройка схожа с настройкой IMP, так что у вас не будет проблем с ними. Просто не забудьте отредактировать registry.php в директории конфигурации horde, и новые приложения появятся внизу странички Horde.

## ucspi-tcp

qmail использует ucspi-tcp для фильтрации входящих соединений. Если вы хотите настроить правила фильтрации, тогда смотрите файлы в /etc/tcprules.d/ (в старых версиях qmail эти файлы находятся в /etc). Здесь вы найдете по два файла на каждый сервис, конфигурационный файл (tcp.qmail-smtp) и его откомпилированную версию, ту которую и использует ucspi-tcp (tcp.qmail-smtp.cdb). Каждый раз после обновления конфигурационного файла, надо пересобрать бинарную его версию. Запустите `tcprules tcp.qmail-smtp.cdb tcp.qmail-smtp.tmp < tcp.qmail-smtp`. Каждый раз при соединении с qmail, откомпилированные правила перечитываются с диска, то есть, нет нужды перезапускать сервис.

## qmail-scanner

Если вы хотите обеспечить фильтрацию контента на своем mail сервере (спам, вирусы), тогда вам необходимо использовать другую программу для организации очередей. Одна из хороших программ это qmail-scanner. Наберите emerge qmail-scanner и отредактируйте /etc/tcprules.d/tcp.qmail-smtp.

Важно: Процесс сборки qmail-scanner очень уродлив. Чтобы поддерживать пакеты, они уже должны быть установлены в системе. Это означает что вы должны установить SpamAssassin и/или Clam AntiVirus до qmail-scanner'a. Смотрите подходящие секции ниже.

**Листинг 7.1:** Смена программы организации очередей.

```
# cd /etc/tcprules.d/
# nano tcp.qmail-smtp
(добавьте QMAILQUEUE="/var/qmail/bin/qmail-scanner-queue" в правило catchall)
# tcprules tcp.qmail-smtp.cdb tcp.qmail-smtp.tmp < tcp.qmail-smtp
```

Смотрите секции об установке спам и вирус фильтров. Вы можете отредактировать несколько опций в /var/qmail/bin/qmail-scanner-queue.pl.

## SpamAssassin

Один из лучших спам фильтров это SpamAssassin. Чтобы установить emerge mail-filter/spamassassin. Пакет поставляется двояко, как версия для командной строки и как клиент/серверная. Для серверов через которые проходит небольшое количество почты, использование версии для командной строки себя оправдывает. Но любой у кого большая загрузка серверов должен использовать клиент/серверную версию.

**Листинг 7.2:** Конфигурирование SpamAssassin

```
# nano /etc/mail/spamassassin/local.cf
```

(чтобы обеспечить самый минимум, добавьте следующие опции:)

```
required_hits 6  
skip_rbl_checks 1
```

```
# rc-update add spamd default
```

```
# /etc/init.d/spamd start
```

```
# nano /var/qmail/bin/qmail-scanner-queue.pl
```

(Убедитесь что \$spamc\_binary variable выставлена в '/usr/bin/spamc'.)

(Если она выставлена в "", тогда смотрите далее.)

**Важно:** Если у вас не был установлен SpamAssassin до установки qmail-scanner'a, Вы должны будете переустановить qmail-scanner. Процесс его сборки очень ужасен, включены будут только те функции, пакеты для которых уже были установлены.

На этом этапе входящая почта будет проходить через qmail-scanner который в свою очередь пропустит ее через SpamAssassin.

## Clam AntiVirus

Как и SpamAssassin, Clam AntiVirus поставляется в двойном исполнении. Я расскажу как быстро поставить клиент/серверную версию. Во-первых, emerge app-antivirus/clamav.

**Листинг 7.3:** Конфигурирование Clam AntiVirus

```
# nano /etc/conf.d/clamd
```

(Установите START\_CLAMD=yes)

```
# nano /etc/clamav.conf
```

(Настройте все как хотите)

```
# rc-update add clamd default
```

```
# /etc/init.d/clamd start
```

```
# nano /var/qmail/bin/qmail-scanner-queue.pl
```

(Убедитесь что \$clamscan\_binary переменная установлена в '/usr/bin/clamscan'. Если нет, смотрите ниже.)

```
# nano /var/qmail/control/conf-common
```

(Если ClamAV сообщает о проблемах с памятью попробуйте увеличить программный лимит (softlimit))

**Важно:** Если у вас не был установлен Clam AntiVirus до установки qmail-scanner'a, Вы должны будете переустановить qmail-scanner. Процесс его сборки очень ужасен, включены будут только те функции, пакеты для которых уже были установлены.

На этом этапе входящая почта будет проходить через qmail-scanner который в свою очередь пропустит ее через Clam AntiVirus.

## 8. Заключительные примечания

У меня нет особых примечаний, но если у вас проблемы с этим руководством, пожалуйста, свяжитесь со мной или разместите сообщение в системе распределения запросов Gentoo. Если у вас есть интересная информация которая вы считаете, может расширить это руководство, то шлите ее мне для включения. Я люблю qmail и с удовольствием добавлю вещи, которые могут расширить опыт работы пользователя с mta

# Руководство по электронной почте с использованием Mutt

Ссылка на оригинал: <http://www.gentoo.org/doc/ru/guide-to-mutt.xml>

С версии: 1.0

## 1. Введение в E-Mail

Если вы не любитель e-mail клиентов с замысловатым графическим интерфейсом, или если вы просто хотите поэкспериментировать с другими клиентами, чтобы выбрать подходящий для себя, самым простым способом в вашем случае будет использование связки консольных приложений:

**fetchmail->procmail->mutt->smtp**

Программы эти не просто мощны и гибко настраиваемы, они также легковесны и эффективны. Потому, однажды настроив и запустив всю эту e-mail систему, вы будете просто ошеломлены тем, что сможете делать с её помощью.

Поскольку это краткое руководство, мы не будем рассматривать службы отправки почты (Mail Transfer Agent), такие как sendmail, postfix или exim. Мы также не будем использовать 25 порт для почтовых служб.

Мы можем себе это позволить, так как fetchmail способен передавать принятую почту непосредственно службе доставки почты (MDA) вместо того чтобы перенаправлять её на 25 порт. И мы можем не использовать службу отправки почты (MTA) для простой передачи почты.

Для запуска своей e-mail системы вам понадобится установить следующее программное обеспечение.

**Листинг 1.1:** Установка нужных программ  
**# emerge fetchmail procmail mutt nbsmtp**

Теперь от полностью рабочей e-mail системы нас отделяют четыре маленьких шажка в направлении конфигурации файлов.

**Важно:** После каждого шага необходимо тестировать настройку на корректность. Это конечно же подразумевают, что всё необходимое программное обеспечение у вас уже установлено.

## 2. Fetchmail

Fetchmail забирает почту с удаленного сервера на вашу локальную машину. Для этого вам необходимо настроить файл .fetchmailrc в вашей домашней директории, подобно примеру:

**Листинг 2.1:** Пример .fetchmailrc

```
poll mail.myisp.net protocol pop3 user "myname" password "mypassword"
```

Сразу после создания файла, необходимо выставить права доступа к нему (он должен быть читаем только его владельцем). Сделать это можно следующей командой:

**Листинг 2.2:** Изменение прав доступа

```
# chmod 710 .fetchmailrc
```

Чтобы увидеть процесс в действии, используйте ключ -v. Чтоб получить все сообщения, используйте -a. И ещё вы должны использовать ключ -m для того, чтобы передать принятую почту procmail.

**Предупреждение:** Будет также хорошей идеей использовать ключ -k, в этом случае если что-то пойдет не так, ваша почта не будет удалена с сервера и вы сможете её повторно забрать.

Пришло время проверить fetchmail в действии!

**Листинг 2.3:** Тест Fetchmail #1

```
# fetchmail -akv -m "/usr/bin/procmail -d %T"
```

Как только вы добьётесь работающей системы, вы можете добавить процесс в cron или какой-нибудь монитор навроде gkrellm. Fetchmail также может быть запущен как демон с указанием секундного интервала.

### 3. Procmail

Procmail - это программа фильтрующая почту получаемую от fetchmail. А далее, также как и MDA, она поставляет отфильтрованную почту в ваши почтовые ящики, откуда её уже можно прочесть программой mutt (это почтовый клиент, который мы будем использовать).

Для использования procmail, также необходимо создать файл .procmailrc в своём домашнем каталоге. Для наших целей "быстрой настройки" мы будем использовать простой .procmailrc, который фильтрует почту от трех списков рассылки gentoo в три почтовых ящика: gentoo-dev, gentoo-user и gentoo-announce.

Примечание: Правила фильтра называются условиями. Я также добавил условия для того, чтобы отфильтровать некоторый спам.

**Листинг 3.1:** Пример .procmailrc

```
MAILDIR=$HOME/MuttMail      ##проверьте правильность пути
LOGFILE=$HOME/.procmaillog
LOGABSTRACT=no
#VERBOSE=on... используется только для отладки
VERBOSE=off
FORMAIL=/usr/bin/formail
NL=""
```

```

"
##условные строки начинаются с :0
##не записывайте комментарии в строки условия
##отредактируйте ненужные условия!
##строки условий начинаются с *, а регулярные выражения ваши лучшие друзья
##условия добавленные после * попадают прямо в egrep
##строка следующая за условиями, в следующем регистре является именем почтового ящика

#отлавливание копий, используя formail
:0 Whc: .msgid.lock
| $FORMAIL -D 16384 .msgid.cache

:0 а
$MAILDIR/duplicates

#люди которые всегда пишут с одного почтового адреса
:0
* ^From:.*(craig\@hotmail|renee\@local.com)
$MAILDIR/friends

#выборка некоторого спама
:0
* ^Subject:.*(credit|cash|money|debt|sex|sale|loan)
$MAILDIR/spam

#никаких html писем
:0
* ^Content-Type:.*html
$MAILDIR/junk

#складировать письма из списков рассылки в мои почтовые ящики
:0
* ^List-Id:.*gentoo-user
gentoo-user

:0
* ^List-Id:.*gentoo-dev
gentoo-dev

:0
* ^List-Id:.*gentoo-announce
gentoo-announce

#получать любую другую почту с gentoo
:0
* ^From:.*gentoo.org
gentoo

:0
* ^From:.*@freshmeat\.net
freshmeat

#####
# последние условие: складировать остальную #
# почту в почтовый ящик по умолчанию #
#####
:0
* .
default

# конец файла

```

**Примечание:** В данном случае вам только потребуется создать почтовую директорию в \$HOME/MuttMail, после чего Procmail создаст все необходимые файлы



почтового ящика в этом каталоге, используя названия из строк условий. Для дополнительной информации посетите <http://www.procmail.org/>

Для проверки нашего .procmailrc, повторно запустите fetchmail (который мы уже настроили). Помните также, что опция -k оставляет почту на удаленном сервере, потому её можно использовать для нашего теста.

**Листинг 3.2:** Тест Procmail #1

```
# fetchmail -akv -m "/usr/bin/procmail -d %T"
```

Ну всё, теперь когда fetchmail и procmail работают, мы можем зайдя в \$HOME/MuttMail прочитать нашу почту программой less или вашим любимым файловым менеджером.

## 4. Почтовый клиент Mutt

Mutt используется для чтения и написания писем. Это очень мощное, сложно-настраиваемое, легковесное и эффективное приложение.

Mutt поддерживает чтение и запись в различных форматах почтового ящика: mbox, MMDF, MH и Maildir. Тип почтового ящика распознается автоматически. В нашем случае мы используем формат mbox, где все сообщения почтового ящика сохраняются в отдельных файлах.

Mutt также имеет способность работать с папками, расположенными на удаленном IMAP сервере. См. Поддержку IMAP в разделе 4.11 руководства по Mutt и сайт Mutt <http://www.mutt.org/>.

При установке mutt создаётся основной файл конфигурации /etc/mutt/Muttrc. Вам также необходимо создать файл .muttrc в своём домашнем каталоге.

**Листинг 4.1:** Пример .muttrc

(Конечно, неплохо бы прочитать документацию Mutt из /usr/share/doc/mutt\*)

(Любые настройки здесь отменяют параметры общей конфигурации из /etc/mutt/Muttrc)

```
# cp /etc/mutt/Muttrc ~/.muttrc
```

```
# nano -w .muttrc
```

```
set pager_context=1
```

```
set pager_index_lines=6          #показывать оглавления в окне программы
```

```
set menu_scroll
```

```
set pgp_verify_sig=no           #не показывать pgp на странице
```

```
set status_on_top               #разместить статусную строку вверху
```

```
set sort=threads                #сортировать сообщения по заголовкам
```

```
set status_format=" %r %b %f %n   Del %d   Msgs %m %l %> (%P)"
```

```
set pager_format="%-10.10i %[%a %b %d %R]"
```

```
set date_format="%H:%M %a %d %b   "
```

```
set index_format="%4C %Z %[%b%d] %-15.15F %s"
```

```
set folder_format="%2C %t %8s %d %N %f"
```

```
#set sendmail="/usr/bin/nbsmtp -d isp.net -h smtp.isp.net -f yourname@isp.net"
```

```
#set from="default-mailaddress"   #задаёт ваш адрес в строке "from"
```

```
#set realname="myname"
```

```

set record="$HOME/MuttMail/sent"      #сохранять отправленную почту здесь
set delete=yes                       #удалить без подтверждения
set include=yes                      #выделять сообщение в ответе
set fast_reply=yes                   #не подтверждать ответ
set beep=no                          #не пищать
set markers=no                       #не помечать + сложенные строки
set confirmappend=no                 #не подтверждать сохранение в =keep
set to_chars=" +TCF"                 #нет L для mail_list

set folder = $HOME/MuttMail
mailboxes =gentoo-user
mailboxes =gentoo-dev
mailboxes =gentoo-announce
mailboxes =gentoo
mailboxes =freshmeat
mailboxes =duplicates
mailboxes =default
mailboxes =sent
mailboxes =friends
mailboxes =junk
mailboxes =spam
mailboxes =keep

save-hook .* =keep                   #mbox по умолчанию сохраняет (s) почту в =keep
subscribe gentoo-user gentoo-dev     #подписанные списки

bind pager h display-toggle-weed     #переключать заголовки кнопкой h

# симулировать старое url меню
macro index \cb |urlview\n 'call urlview to extract URLs out of a message'
macro pager \cb |urlview\n 'call urlview to extract URLs out of a message'

#запуск fetchmail нажатием кнопки G
macro index G "!fetchmail -a -m 'procmail -d %T\r'"
macro pager G "!fetchmail -a -m 'procmail -d %T\r'"

#редактирование .muttrc... не требует перезапуска

macro generic ,sm ":source $HOME/.muttrc\r"
macro generic \cj "!rxvt -bg wheat -e joe $HOME/.muttrc\r"

#по умолчанию список заголовков в полях удаляется перед показом почты
#игнорирует всё, кроме того, что вам нужно
ignore *
unignore Date To From: Subject X-Mailer Organization User-Agent
hdr_order Date From To Subject X-Mailer User-Agent Organization

##ваш Mutt должен поддерживать несколько цветов
##для обозначения четырех уровней выделенного текста
##данные настройки отменяют параметры общей конфигурации в /etc/mutt/Muttrc

#color quoted green default
#color quoted1 magenta blue
#color quoted2 yellow default
#color quoted3 red default
#color signature cyan cyan

#эта цветовая схема взята из /etc/mutt/Muttrc.color
#закомментируйте её, если вам нужна цветовая схема по умолчанию из /etc/mutt/Muttrc
# Je vois la vie en rose :-)
color   hdrdefault      brightcyan      blue
color   header          brightwhite     blue "^from:"
color   header          brightwhite     blue "^subject:"

```

```

color quoted      brightgreen blue
color signature   brightwhite blue

color indicator   blue      green

color error       red        black
mono error       bold
color status      black cyan
mono status      bold
color tree        yellow     blue

color tilde       brightmagenta blue
color body        brightwhite   blue "[-a-z_0-9.]+@[-a-z_0-9.]+"
mono body        bold          "[-a-z_0-9.]+@[-a-z_0-9.]+"
color body        brightyellow  black  "^Good signature"
mono body        bold          "^Good signature"
color body        brightwhite   red    "^Bad signature from.*"
mono body        bold          "^Bad signature from.*"
color normal      white         blue
color message     green        black
color attachment  brightgreen  blue

```

# конец файла... но вы можете его дописывать и дописывать... :)

Это только маленький пример файла .muttrc. На самом же деле гораздо больше опций поддаются конфигурации, например, те же настройки gpg. Для примеров и помощи посмотрите <http://mutt.netliberte.org/>.

Теперь вы можете протестировать наш .muttrc

**Листинг 4.2:** Тест .muttrc

```
# mutt -y
```

После чего должно появиться окно Mutt с почтовыми ящиками, которые мы создали, когда настраивали fetchmail.

Нажмите ? для получения помощи в навигации по почтовым ящикам в Mutt.

## 5. SMTP

Последний шаг, это настройка nbsmtp ('No-Brainer SMTP'), используемого для отправки почты на ваш сервер SMTP. Данная настройка самая простая и требует лишь добавления нескольких строк в конфигурационный файл .muttrc

domain: домен, сообщаемый nbsmtp. Будет почти всегда похож на окончание вашего адреса электронной почты.

from@addr: Это тот адрес, который будет сообщаться nbsmtp в строке "from". Обратите внимание, что данный адрес может отличаться от того, что записан в поле "From:" вашего почтового клиента (MUA).

host: Сервер smtp, куда, собственно, и будет отправляться почта.

**Листинг 5.1:** Добавление поддержки smtp

```
# nano -w .muttrc
```

```
set sendmail="/usr/bin/nbsmtp -d isp.net -h smtp.isp.net -f urname@isp.net"
```

Теперь всё готово для создания письма. В окне Mutt нажмите `m`, для того чтобы написать текстовое сообщение на ваш же почтовый ящик. Mutt использует значение `EDITOR` или `VISUAL`, указываемое с помощью опций `editor=` в `.muttrc`. После того, как сообщение будет написано, нажмите `u` для его отправки. Если всё прошло удачно, мы увидим сообщение 'sending mail', следующее за 'New mail in =sent'.

Помните, в `.muttrc` у нас задано сохранять всю отправленную почту при помощи строки `:set record="$HOME/MuttMail/sent"`

Теперь, чтобы завершить испытание, снова запустите `fetchmail` для получения всей почты и вашего тестового письма, которое вы себе отправили. Как только это тестовое письмо будет найдено, нажмите `h` для того, чтобы посмотреть все его заголовки и полный путь прохождения почты (mail transfer path).

Примечание: Есть ещё одна программа, которая, возможно, вам пригодится, под названием `urlview`. Она извлекает ссылки из текстовых сообщений и перенаправляет их в ваш браузер.

**Листинг 5.2:** Установка urlview

```
# emerge urlview
```

После чего создайте `~/.urlview` копируя конфигурационный файл из `/usr/share/doc/urlview*/`, и подправьте необходимые настройки под свой браузер.

Чтож, теперь у нас есть мощная почтовая система. Читайте различную документацию и руководства, а также ищите примеры конфигурационных файлов в 'google' по ключевым словам `muttrc` и `prosmailrc`.

# Руководство по предварительному связыванию в Gentoo Linux

Ссылка на оригинал: <http://www.gentoo.org/doc/ru/prelink-howto.xml>

С версии: 1.0

## 1. Введение

**Что такое предварительное связывание (Prelink) и как оно может помочь мне?**

Большинство приложений используют разделяемые библиотеки. Эти разделяемые библиотеки должны быть загружены в память во время выполнения программы, при этом необходимо определить различные символьные ссылки. Для большинства небольших программ динамическое связывание происходит очень быстро. Но для программ, написанных на C++, имеющих много библиотечных зависимостей, динамическое связывание может занять много времени.

На большинстве систем, библиотеки обычно остаются неизменными и когда программа запускается, операции необходимые для связывания, каждый раз одинаковы. Предварительное связывание использует данную особенность, выполняя связывание и сохраняя библиотеки в выполняемом файле, фактически уже связанными. Для связывания, вам потребуется ld-linux.so в glibc; для распознавания предварительного связывания версия glibc должна быть  $\geq 2.3.1-r2$

Предварительное связывание может уменьшить время загрузки приложений. Например, типичные программы KDE могут загружаться на 50% быстрее. Необходимым условием является перезапуск процесса предварительного связывания каждый раз при обновлении библиотек для программ, которые на них ссылаются.

### Резюме

Предварительное связывание производится программой, имеющей название, как не странно, prelink. Она изменяет выполняемый файл таким образом, что он запускается быстрее.

Если зависимые от приложения библиотеки были изменены после того, как были предварительно связаны с приложением, то они потребуют повторного предварительного связывания, иначе вы потеряете преимущества в скорости загрузки.

Изменения выполняемого файла полностью обратимы, поскольку программа prelink имеет функцию undo.

Новые версии portage могут обрабатывать, при помощи программы prelink, изменяющиеся контрольные суммы MD5 и поля mtime выполняемых файлов.

У вас должен быть glibc-2.3.1-r2 или более новый, и бинарные файлы должны быть скомпилированы с binutils-2.13.90.0.xx или выше.

## **2. Инициализация предварительного связывания**

### **Установка программ**

Примечание: Я подразумеваю, что у вас есть Gentoo-1.4, который был собран при помощи gcc-3.2 или более поздней версии и binutils-2.13.90.0.xx. Это требуется для того, что бы выполняемые файлы могли быть предварительно связаны.

Предупреждение: Должен быть установлен glibc 2.3.1 или более новый, иначе программа prelink может испортить все ваши выполняемые файлы!

Для начала обновите свое дерево портежей, потому что многие из необходимых приложений часто обновляются и постоянно выходят исправления ошибок.

**Листинг 2.1:** Обновление вашего дерева портежей

```
# emerge sync
```

Далее, убедитесь, что у вас установлен portage-2.0.26 или более новый. Это требуется для того, что бы portage могло распознать предварительно связанные выполняемые файлы и при необходимости корректно их деинсталлировать. Т.к. предварительное связывание изменяет контрольную сумму MD5 бинарных файлов.

**Листинг 2.2:** Проверка версии дерева портежей

```
# emerge ">=portage-2.0.46"
```

Теперь вы можете вызвать emerge для программы предварительного связывания. Процесс emerge автоматически проверит, что ваша система может безопасно выполнять предварительное связывание.

**Листинг 2.3:** Устанавливаем предварительное связывание

```
# emerge prelink
```

Некоторые люди, в ходе выполнения тестов, получают ошибки выполняя emerge для программы prelink. Тесты размещены в пакете из соображений безопасности, предварительное связывание будет непредсказуемым, если их отменить. Эти ошибки обычно связаны с базовыми пакетами, такими как binutils, gcc и glibc. В таком случае попробуйте выполнить emerge этих пакетов повторно.

**Примечание:** Если ошибки продолжают появляться, попробуйте скомпилировать и протестировать программу prelink самостоятельно (./configure ; make ; make check ). В случае сбоя вы можете просмотреть \*.log файлы в директории с набором тестов. Они могут дать вам некоторые полезные подсказки.

Если вы имеете набор действий, которые демонстрируют ошибки в другой системе, пожалуйста отправьте его на Stefan Jones.

## Настройка

Portage автоматически генерирует файл /etc/prelink.conf, который говорит программе prelink какие файлы требуется предварительно связать.

К сожалению, вы не сможете предварительно связать файлы, которые были скомпилированы старыми версиями пакета binutils. Большинство таких приложений поставляется уже скомпилированными, не имея исходных кодов, и устанавливаются в каталоге /opt. Создайте следующий файл, для того, что бы запретить программе prelink использовать предварительное связывание для подобных файлов.

**Листинг 2.4:** /etc/env.d/99prelink

```
PRELINK_PATH_MASK="/opt"
```

**Примечание:** Вы можете добавить больше или меньше директорий, в список, разделенный двоеточиями.

## 3. Предварительное связывание

### *Применение предварительного связывания*

Я использую следующую команду для предварительного связывания всех бинарных файлов в директориях, заданных в файле /etc/prelink.conf.

**Листинг 3.1:** Предварительное связывание заданных файлов

```
# prelink -afmR
```

**Предупреждение:** Наблюдалось, что при наличии небольшого свободного пространства в момент выполнения предварительного связывания для всей системы, существует вероятность усечения выполняемых файлов. Результатом может стать испорченная система. Используйте команду file или readelf для проверки

целостности выполняемых файлов. Также вы можете заранее проверить наличие свободного места командо `df -h`

Описание опций:

- `-a` All, применить предварительное связывание ко всем выполняемым файлам.
- `-f` Вынуждает повторить предварительное связывание для файлов, которые уже подвергались ему. Это требуется т.к. программа `prelink` прекращает обработку существующих старых связанных файлов, зависимые библиотеки которых могли измениться.
- `-m` Сохраняет пространство виртуальной памяти. Это требуется, если вы имеете много библиотек, которые необходимо связать.
- `-R` Random, -- Выбирает случайный порядок адресов, это увеличивает безопасность против переполнений буферов.

**Примечание:** Более подробно об опциях можно узнать, набрав `man prelink`

## 4. Известные проблемы и методы их решения

**"Предварительное связывание не работает с разделяемой библиотекой, собранной без опции PIC"**

Причиной данной проблемы являются объектные файлы разделяемой библиотеки, скомпилированные без опции `gcc -fPIC`.

Далее следует список проблемных библиотек и список пакетов, для которых требуется повторно вызвать `emerge`, в случае возникновения данной проблемы.

**Листинг 4.1:** Исправления

(Для библиотеки `ORBit /usr/lib/liblOP.so.0.5.17`)

**`emerge ">=sys-apps/tcp-wrappers-7.6-r4" ORBit`**

(Для библиотеки `zlib /usr/lib/libz.so.1.1.4`)

**`emerge ">=sys-libs/zlib-1.1.4"`**

(Для `svglib, /usr/lib/libsvg.so.xx`)

**`emerge ">=media-libs/svglib-1.9.16"`**

(Для библиотеки `XFree OpenGL, libGLU.so.1`)

**`emerge ">=x11-base/xfree-4.2.1-r2"`**

(Для `libpcap.so.0.6`)

**`emerge ">=net-libs/libpcap-0.7.1-r2"`**

(Для библиотеки `lcms, /usr/lib/liblcms.so.1`)



**Примечание:** Многие библиотеки статично связаны с zlib и/или tcp-wrappers, так что сначала попробуйте выполнить emerge для них, после чего, повторно запустите emerge для проблемных библиотек.

Если вы имеете проблемы с предварительным связыванием QT/KDE, то сначала попытайтесь обновить x11-base/xfree до версии 4.2.1-r2 или более новой и x11-libs/qt до версии 3.1.0-r1 или более новой. Если QT так и не работает, тогда попробуйте скомпилировать его без поддержки xinerama, задав строку myconf="-no-xinerama \$ {myconf}" в файле ebuild для qt.

Далее приводится список библиотек, которые пока не работают или не могут работать вообще:

- Библиотеки в пакете wine, включая windex. Они в любом случае не могут ускорить выполняемые файлы MS Windows.
- Библиотека в media-video/mjpgtools, /usr/lib/liblavfile-1.6.so.0

Если у вас есть проблемы с библиотекой, которой нет в списке, пожалуйста сообщите об этом, предпочтительно, добавив -fPIC к соответствующим CFLAGS.

**Прерывание предварительного связывания файла, с выводом сообщения, вида: "1631 Aborted ...."**

Вам необходимо использовать опцию -f программы prelink; т.е. повторить предварительное связывание для всей системы с нуля. Попробуйте запустить prelink -af

**"Ошибка: <file>: error while loading shared libraries: unexpected reloc type..."**

Это было исправлено в пакете sys-libs/glibc-2.3.1-r2 2002/11/18, выполните emerge для glibc, если у вас более старая версия.

Так же запустите prelink -u -a -m ; prelink -a -m это должно помочь. Если все это не помогает, просто запустите prelink -u <file>

**У меня проблемы с библиотеками Nvidia openGL**

Ускоренные библиотеки openGL идущие с пакетом nvidia-glx скомпилированы нестандартным образом, и программа prelink выводит массу предупреждений. Здесь нет повода для беспокойства, и это не может быть исправлено кем-либо помимо компании nvidia. Вы всегда можете вернуться к xfree версии библиотеки libGL.so, если вам не требуется ускорение 3D. Хотя драйвер xfree nvidia работает нормально.

**После предварительного связывания моей системы, некоторые статичные выполняемые файлы более не работают**

Там где участвует glibc, не бывает такой вещи как 100% статичный выполняемый файл. Если вы статично скомпилировали выполняемый файл с glibc, он может продолжать оставаться зависимым от других системных файлов. Далее приводится объяснение от Dick Howell.

"Я предполагал, что все должно находиться в скачанном файле и не должно быть никаких зависимостей с локальными библиотеками на целевой системе. К сожалению, в Linux, и я думаю везде, где используется GLIBC, это до сих пор не совсем верно. Существует библиотека "libnss" (name service switch - переключатель сервиса имен, некоторые люди думают, что это network security system - сетевая система безопасности) которая предоставляет функции для доступа к различным базам данных для аутентификации, сетевой информации и для других вещей. Эта библиотека предполагает сделать приложения независимыми от отдельно сконфигурированных сетевых окружений машины. Хорошая идея, но изменения GLIBC может вести к проблемам ее загрузки. И вы не можете статически связать "libnss" так как она сконфигурирована для каждой машины индивидуально. Я думаю, что, главным образом, это происходит из-за статического связывания других GLIBC библиотек, особенно "libpthread", "libm" и "libc", откуда приходят несовместимые вызовы к функциям "libnss"

**Предварительное связывание обрывается с ошибкой

```
prelink: dso.c:306: fdopen_dso: Assertion `j == k' failed.
```

"**

Это известная проблема, любезно продиагностирована здесь. Программа prelink не может справиться с выполняемыми файлами сжатыми методом UPX. Для версии prelink-20021213 нет другого способа решить эту проблему, кроме как спрятать выполняемые файлы во время предварительного связывания. Смотрите раздел конфигурации выше, где обсуждается вопрос о том, как это осуществить.

## **5. Заключение**

Предварительное связывание может решительно ускорить время запуска для некоторых больших приложений. Его поддержка встроена в дерево портежей. Предварительное связывание безопасно, т.к. вы всегда можете отменить его действие для любого выполняемого файла, если вы столкнетесь с какими-либо проблемами. Помните что когда вы обновляете glibc или другие библиотеки, с которыми вы осуществили предварительное связывание, вам надо будет перезапустить prelink. В общем, удачи!

# Руководство по файловой системе для устройств

Ссылка на оригинал: <http://www.gentoo.org/doc/ru/devfs-guide.xml>

С версии: 1.0

## 1. Что такое devfs?

### Старые (добрые?) времена

Традиционная реализация Linux предоставляет пользователю абстрактный каталог устройств, называемый `/dev`. Внутри этого каталога находятся device nodes, специальные файлы, символизирующие устройства внутри системы. Например, `/dev/hda` символизирует первое IDE устройство в вашей системе. Файлы устройств позволяют создавать программы, которые взаимодействуют с устройствами так, как если бы устройства были обычными файлами, вместо использования специальных API.

Файлы устройств подразделяются на две группы, называемые character devices (символьные устройства) и block devices (блочные устройства). Первая группа содержит устройства, для которых отсутствует буферизация чтения/записи. Вторая группа, естественно, содержит устройства, для которых чтение/запись буферизируется. Из обоих типов устройств может быть прочитан за раз один символ или блок. Поэтому такой способ присваивания имён может сбивать с толку и на самом деле неправилен.

Если посмотреть на конкретное устройство, можно обнаружить что-то, похожее на это:

**Листинг 1.1:** Исследование информации о файле устройства

```
# ls -l /dev/hda
```

```
brw-rw---- 1 root disk 3, 0 Jul 5 2000 /dev/hda
```

В предыдущем примере мы увидели, что `/dev/hda` — это блочное устройство. Однако важнее то, что ему присвоено два специальных номера 3, 0. Эта пара называется major-minor. Она используется ядром, чтобы соотнести файл устройства и реальное устройство. major (старший) относится к типу устройства, minor (младший) к конкретному устройству. Выглядит запутано, не правда ли?

Ещё два примера — `/dev/hda4` и `/dev/tty5`. Первое устройство соответствует четвёртому разделу на первом IDE-устройстве. Его пара major-minor 3, 4. Другими словами, minor соответствует разделу, тогда как major соответствует устройству. Во втором примере пара major-minor — 4, 5. В этом случае первое число соответствует драйверу терминала, тогда как второе соответствует номеру терминала (в данном

случае пятый терминал).

## Проблемы

Если вы заглядывали в папку `/dev`, вы обнаружили, что там перечислены не только все ваши устройства, но и все возможные устройства, которые только могут быть. Другими словами, у вас есть файлы устройств для устройств, которых у вас нет. Управление такой кучей устройств по крайней мере можно назвать громоздким. Представьте, что надо поменять права на все устройства, реально существующие в вашей системе, оставив остальные без изменений.

Затем вы добавили новое оборудование в вашу систему, для него может не оказаться уже существующего файла. Продвинутые пользователи знают, что эта задача может быть выполнена с помощью команды `./MAKEDEV` внутри директории `/dev`, но разве вы сразу знаете, что за устройство вам придётся создать?

Когда у вас есть программы, взаимодействующие с оборудованием при помощи файлов устройств, вы не можете смонтировать корневой раздел только для чтения, в то время как в дальнейшем нет необходимости в том, чтобы он был смонтирован на чтение и запись. И вы не можете иметь `/dev` на отдельном разделе, так как `mount` необходим, `/dev` чтобы монтировать разделы.

## Решения

Как вы могли себе представить, kernel hackers нашли достаточно решений для вышеперечисленных проблем. Однако многие из этих решений имеют собственные проблемы описанные в <http://www.atnf.csiro.au/people/rgooch/linux/docs/devfs.html#faq-why>. Мы не будем обсуждать эти варианты, а сконцентрируемся на одном способе, который был реализован в официальной версии исходников ядра:

### **devfs как абсолютный победитель?**

**devfs** решает все перечисленные проблемы. Она просто предоставляет пользователю доступ к существующим устройствам, добавляет новые device nodes (файлы устройств), когда найдены новые устройства, и делает возможным монтировать корневую файловую систему в режиме `read only` (только чтение). А также решает многие проблемы, которые мы раньше не обсуждали, потому что они не так интересны для пользователей...

Как пример, с **devfs** вам не надо беспокоиться о паре `major/minor`. Она продолжает поддерживаться (для обратной совместимости), но в ней нет необходимости. Это позволяет Linux поддерживать ещё больше устройств, так как больше нет ограничений (числа всегда имеют границы :)

Однако у devfs есть свои проблемы, не столь очевидные для пользователей, но достаточно серьезные, чтобы разработчики ядра поместили ее как *obsolete* (устаревшее), порекомендовав использовать udev, которая также поддерживается Gentoo.

Чтобы узнать, почему devfs считается устаревшей, читайте udev FAQ и udev versus devfs document.

## 2. Навигация через дерево устройств

### Директории

Одна из первых особенностей которые вы можете заметить это то что devfs использует директории для объединения устройств вместе. Это повышает читабельность, так как теперь все связанные между собой устройства находятся внутри одной общей директории.

Например, все устройства, относящиеся к IDE, находятся в директории `/dev/ide/`, а все относящиеся к SCSI в директории `/dev/scsi/`. SCSI и IDE диски во многом похожи, у них одинаковая структура поддиректорий.

IDE и SCSI диски управляются при помощи адаптера (встроенного или отдельной платой), называемого *host*. Каждый адаптер может иметь несколько каналов. Канал называется *bus*. На каждом канале может быть несколько IDs (идентификаторов). ID служит для идентификации диска. Этот ID называется *target*. Многие SCSI устройства могут иметь множество LUN (Logical Unit Numbers (Номер Логического Устройства)), Например устройства которые управляют несколькими носителями одновременно (*hi-end tapedrives*). У вас скорее всего будет только один *lun*, `lun0/`.

Итак, несмотря на то, что раньше использовался `/dev/hda4`, теперь появился `/dev/ide/host0/bus0/target0/lun0/part4`. Это намного проще... нет, не спорьте со мной... это проще... как бы то ни было! :)

Примечание: Вы также можете использовать более похожие на Unix названия для жёстких дисков, такие как `c0b0t0u0p2`. Они могут быть найдены в `/dev/ide/hd`, `/dev/scsi/hd` и т.д.

Чтобы дать вам лучше понять идею с директориями, вот листинг директорий которые есть у меня:

#### Листинг 2.1: Директории в /dev

```
cdroms/  cpu/    discs/   floppy/
ide/      input/  loop/    misc/
netlink/  printers/ pts/      pty/
scsi/     sg/      shm/      sound/
sr/       usb/     vc/       vcc/
```

## Обратная совместимость при помощи devfsd

Использование этой новой структуры выглядит здорово, но многие утилиты и программы используют предыдущую, старую структуру. Для уверенности, что система не будет нарушена, был создан devfsd. Этот демон создаёт символичные ссылки на новые файлы устройств, но со старыми именами (compatibility symlinks).

**Листинг 2.2:** Созданные символичные ссылки

```
$ ls -l /dev/hda4
```

```
lr-xr-xr-x  1 root  root      33 Aug 25 12:08 /dev/hda4 -> ide/host0/bus0/target0/lun0/part4
```

При помощи devfsd, вы можете устанавливать права доступа, создавать новые файлы устройств и т.д. Всё это описывается в следующей главе.

## 3. Администрирование дерева устройств

### Перезагрузка devfsd

Если вы изменили файл /etc/devfsd.conf, и хотите чтобы изменения вступили в силу, вам не обязательно перезагружаться. В зависимости от того, что вы хотите, вы можете использовать любой из следующих сигналов:

SIGHUP заставит devfsd перечитать конфигурационный файл, перезагрузить разделяемые объекты (shared objects) и сгенерировать событие REGISTER для каждого листа в дереве устройств.

SIGUSR1 сделает то же самое, но не будет событий REGISTER.

Чтобы послать сигнал, просто используйте kill или killall:

**Листинг 3.1:** Посылка сигнала SIGHUP демону devfsd

```
# kill -s SIGHUP `pidof devfsd`
```

или

```
# killall -s SIGHUP devfsd
```

### Удаление compatibility symlinks

**Предупреждение:** В настоящее время Gentoo не может существовать без этих ссылок.

Если вы хотите удалить из вашей системы ссылки которые засоряют /dev (в Gentoo они используются по умолчанию), отредактируйте /etc/devfsd.conf и удалите следующие две строки:

**Листинг 3.2:** /etc/devfsd.conf для обратной совместимости

# Закомментируйте эти две строки для удаления симлинков

```
REGISTER    .* MKOLDCOMPAT
UNREGISTER  .* RMOLDCOMPAT
```

Вам придётся перезагрузиться, чтобы изменения вступили в силу.

## Удаление возможности автосоздания файлов устройств

Когда вы загружаете модуль, devfs автоматически создаёт файлы устройств. Если вы не хотите, чтобы он так делал, удалите эту строку из /etc/devfsd.conf:

**Листинг 3.3:** /etc/devfsd.conf, autoload functionality

```
LOOKUP     .* MODLOAD
```

## 4. Вопросы, относящиеся к правам доступа

### Установка/изменение прав доступа при помощи PAM

Хотя вы можете установить права доступа в /etc/devfsd.conf, мы советуем использовать PAM (Pluggable Authentication Modules ()). Так как PAM имеет решающий голос при установке прав доступа, и может проигнорировать изменения, которые вы сделали в /etc/devfsd.conf.

PAM использует /etc/security/console.perms для установки прав доступа. Файл состоит из двух частей: в первой описываются группы, а во второй права.

Давайте сначала взглянем на часть с группами. Как пример мы рассмотрим sound-group:

**Листинг 4.1:** Sound group в /etc/security/console.perms

```
<sound>=/dev/dsp* /dev/audio* /dev/midi* \
/dev/mixer* /dev/sequencer* \
/dev/sound/* /dev/snd/* /dev/beep \
/dev/admm* \
```

```
/dev/adsp* /dev/aload* /dev/amidi* /dev/dmfm* \  
/dev/dmmidi* /dev/sndstat
```

Синтаксис достаточно прост: вы начинаете с имени группы, и заканчиваете списком устройств, принадлежащих этой группе.

Теперь для того, чтобы с группами можно было что-нибудь сделать, рассмотрим следующую часть, описывающую, как управлять правами.

**Листинг 4.2:** Права доступа для sound group в /etc/security/console.perms

```
<console> 0600 <sound> 0600 root.audio
```

- Первое поле — это проверка терминала. На большинстве систем это console-group. РАМ будет проверять это поле при каждом входе в систему. Если вход произошёл на устройстве, содержащемся в console-group, РАМ проверит и возможно сменит права на некоторые файлы устройств.
- Второе поле содержит права, которые установятся на файл устройства после удачного входа в систему. Когда человек вошел в систему, а файлы устройств принадлежат пользователю и группе по умолчанию, РАМ сменит владельца на вошедшего пользователя и установит на них права из второго поля. В данном случае используется 0600 (пользователь имеет право на чтение/запись, все остальные нет).
- В третьем поле содержатся группы устройств, чьи права будут изменены. В данном случае, вторая группа (все устройства, относящиеся к звуку) будут изменены.
- Четвёртое поле определяет права, которые будут установлены на файлы устройств после возврата в состояние по умолчанию. Другими словами, если человек, который владеет правами на все файлы устройств, выйдет из системы, РАМ установит права обратно в состояние по умолчанию, описанному в этом четвёртом поле.
- Пятое поле определяет собственника (с группой если вам надо) к которому будут установлены атрибуты устройства после возврата в состояние по умолчанию. Другими словами, если человек, владеющий правами на все файлы устройств, выйдет из системы, РАМ установит собственника обратно в состояние по умолчанию, описанному в пятом поле.

## Установка/изменение прав доступа при помощи devfsd

Если вы действительно хотите установить права, используя /etc/devfsd.conf, тогда используйте синтаксис приведённый в этом примере:

**Листинг 4.3:** Права в /etc/devfsd.conf

```
REGISTER ^cdroms/. * PERMISSIONS root.cdrom 0660
```



Второе поле — это группа устройств, начиная с /dev. Это регулярное выражение, означающее, что вы можете выбрать несколько файлов устройств с одним правилом.

Четвёртое поле — это владелец файла устройства. В отличие от PAM, он не изменяется (если он не упоминается в console.perms, так как PAM главнее).

Пятое поле содержит права на файлы устройств.

## Ручная установка прав и их сохранение devfsd

Это обычная ситуация для Gentoo: если вы делаете chown (CHange OWNEr (Смена владельца)) и chmod (CHange MODe (Смена вида)) некоторым файлам устройств, то devfsd сохраняют информацию, когда вы выключаете систему. Это происходит из-за того, что файл /etc/devfsd.conf содержит следующие строки:

**Листинг 4.4:** /etc/devfsd.conf для сохранения прав доступа

```
REGISTER    ^pt[sy]/.*  IGNORE
CHANGE      ^pt[sy]/.*  IGNORE
CREATE      ^pt[sy]/.*  IGNORE
DELETE      ^pt[sy]    IGNORE
REGISTER    ^log      IGNORE
CHANGE      ^log      IGNORE
CREATE      ^log      IGNORE
DELETE      ^log      IGNORE
REGISTER    .*        COPY   /lib/dev-state/$devname $devpath
CHANGE      .*        COPY   $devpath /lib/dev-state/$devname
CREATE      .*        COPY   $devpath /lib/dev-state/$devname
DELETE      .*        CFUNCTION GLOBAL unlink
/lib/dev-state/$devname
RESTORE      /lib/dev-state
```

Другими словами, изменённые файлы устройств копируются в /lib/dev-state, когда выключается система, и копируются в /dev, когда система грузится.

Другая возможность - монтировать /lib/dev-state в /dev во время загрузки. Чтобы это сделать, вы должны быть уверены, что devfs не монтируется автоматически (это значит, что вы должны перекомпилировать ядро), и что /dev/console существует. Затем, где-то в начале bootscripts (загрузочных скриптов) вашей системы, вы должны разместить:

**Листинг 4.5:** Монтирование /lib/dev-state в /dev

```
mount --bind /dev /lib/dev-state
mount -t devfs none /dev
devfsd /dev
```

# Русское WIKI

## ***Настройка дисковой подсистемы***

### **Всеобщий ликбез**

Под IDE понимаются устройства, подключаемые к IDE-интерфейсу. Обычно это жесткие диски и дисководы CD-ROM. Эти устройства должны быть документированы, как:

- IDE
- ATA
- ATAPI
- Enhanced IDE (EIDE)
- Fast ATA или Fast ATA-2

IDE устройства могут передавать данные по шине в двух режимах - PIO и DMA. Учтите, что это именно передача данных по шине, с пластины данные читаются медленнее, а вот из аппаратного кеша на полной скорости.

PIO - Программный ввод/вывод, метод передачи данных между двумя устройствами, использующий процессор как часть маршрута данных (процессор выполняет команду чтения порта, считывает байт или слово данных в свой регистр, после чего переписывает его в память, затем повторяет эту процедуру до тех пор, пока вся необходимая информация не будет считана из устройства в память).

Бывает PIO Mode 0 1 2 3 4. Чем больше номер режима, тем быстрее. IDE ZIP100 приводы от Iomega например умеют только PIO mode 0. Старые CD-ROM обычно умеют PIO mode 4, если не могут DMA.

PIO Mode 0 = 3.3 Mb/s

PIO Mode 1 = 5.2 Mb/s

PIO Mode 2 = 8.3 Mb/s

PIO Mode 4 = 11.1 Mb/s

PIO Mode 5 = 16.7 Mb/s

DMA - Direct Memory Access - прямой доступ к памяти - собирательное название протоколов, позволяющих периферийному устройству передавать информацию непосредственно в системную память без участия центрального процессора, жесткие диски используют эту возможность в сочетании с возможностью

перехватывать управление шиной и самостоятельно управлять передачей информации (bus mastering), что уменьшает нагрузку на процессор и повышает скорость передачи данных.

DMA встречается двух типов: UDMA и MDMA.

UDMA - ultra DMA - наиболее предпочитаемый тип, основной на сегодня. Бывает UDMA 0 1 2 3 4 5 6. Реально встречаются:

UDMA 2 = 33 mb/s

UDMA 4 = 66 mb/s

UDMA 5 = 100 mb/s

UDMA 6 = 133 mb/s

Последний встречается на не Intel матерях и не все винты его умеют. Кстати SATA винчестеры используют UDMA = 150 Mb/s.

Для UDMA 66 - 100 - 133 необходим 80-жильный шлейф, кроме того старые матери умеют его не на всех каналах, смотрите в инструкцию. Форсирование этих режимов с 40-жильным шлейфом может убить Ваши данные.

MDMA - multiword dma, более древний режим, предпочтителен для старых винчестеров и CD-ROM.

MDMA0 = 4.2 mb/s

MDMA1 = 13.3 mb/s

MDMA2 = 16.7 mb/s

На большинстве современных систем ядро автоматически определяет и настраивает IDE подсистему на максимальную производительность, если правильно его сконфигурировать. Но настроить что-нибудь всё равно можно.

## ОЧЕНЬ ВАЖНО

Всё нижеописанное может убить Ваши данные, сломать Вам винчестер, спалить Вашу машину, удивить Вашу кошку и т.п. отмазы :)

Всё нижеописанное тестировалось, работало и работает на пяти машинах с ядром 2.6.9-gentoo-r4 и hdparm-5.7-r1, ACCEPT\_KEYWORDS="~x86".

У Вас что-нибудь может не работать, работать не так.

## ВАЖНО

Всё нижеописанное не относится к SATA дискам работающим через libata

интерфейс (то есть видимым как sdX а не hdX). Поскольку sata интерфейс гораздо ближе к scsi чем к pata, разработчики вполне обоснованно используют scsi подсистему для работы с sata. Поэтому на текущий момент настройка sata винчестеров средствами hdparm невозможна, так как в libata не реализована специфичная для подобных програм функциональность.

В Сети есть патчик на ядро, добавляющий нужный функционал, но он ОЧЕНЬ сыр и с вероятностью в 100% убьёт вам раздел. Даже ссылку давать не буду, если Вы камикадзе, сами найдёте.

Для начала

Убедитесь, что Ваши IDE диски используют DMA интерфейс, и ядро правильно настроено.

Для этого сделайте

`dmesg | less`

или загляните в логи.

Нас будут интересовать сообщения о настройке ide контроллера и дисков. Мой кусок kern.log выглядит так (с моими комментариями):

Nov 14 17:45:54 tsotimus kernel: ide: Assuming 33MHz system bus speed for PIO modes; override with idebus=xx

```

                ~~~~~
~~~~~
```

Можно при загрузке сказать ядру idebus=66, но это работает только для

не использующих DMA винчестеров и сидиромов.

Подробности в

`/usr/src/linux/Documentation/ide.txt`

Nov 14 17:45:54 tsotimus kernel: ICH2: IDE controller at PCI slot 0000:00:1f.1

Nov 14 17:45:54 tsotimus kernel: ICH2: chipset revision 2

```

                ~~~~~
```

Определили первый контроллер.

Nov 14 17:45:54 tsotimus kernel: ICH2: not 100%% native mode: will probe irqs later

```

                ~~~~~
```

Это из-за того что интеловский контроллер использует разные прерывания для

каждого из каналов. Это нормально.

Nov 14 17:45:54 tsotimus kernel: ide0: BM-DMA at 0xf000-0xf007, BIOS settings: hda:DMA, hdb:pio

контроллер работает в busmaster режиме, dma для первого

диска

включен в BIOS.

Nov 14 17:45:54 tsotimus kernel: ide1: BM-DMA at 0xf008-0xf00f, BIOS settings: hdc:prio, hdd:DMA

Nov 14 17:45:54 tsotimus kernel: Probing IDE interface ide0...

Nov 14 17:45:54 tsoptimus kernel: hda: ST340016A, ATA DISK drive

Nov 14 17:45:54 tsoptimus kernel: ide0 at 0x1f0-0x1f7,0x3f6 on irq 14

Nov 14 17:45:54 tsoptimus kernel: Probing IDE interface ide1...

Nov 14 17:45:54 tsoptimus kernel: hdd: DV-516E, ATAPI CD/DVD-ROM drive

Nov 14 17:45:54 tsoptimus kernel: ide1 at 0x170-0x177,0x376 on irq 15

Nov 14 17:45:54 tsotimus kernel: PDC20265: IDE controller at PCI slot 0000:02:0a.0

[illegible]

## Определили интегрированный promise контроллер

Nov 14 17:45:54 tsotimus kernel: ACPI: PCI interrupt 0000:02:0a.0[A] -> GSI 17 (level, low) -> IRQ 17

Nov 14 17:45:54 tsoptimus kernel: PDC20265: chipset revision 2

Nov 14 17:45:54 tsotimus kernel: PDC20265: 100%% native mode on irq 17

Nov 14 17:45:54 tsoptimus kernel: PDC20265: (U)DMA Burst Bit ENABLED Primary MASTER Mode Secondary MASTER Mode.

[illegible]

△△△△△△△△△△

BIOS этого уродца не включает DMA, поэтому его включило ядро

:)

Nov 14 17:45:54 tsoptimus kernel: ide2: BM-DMA at 0xac00-0xac07, BIOS settings: hde:pio, hdf:pio

^ ^ ^ ^ ^ ^ ^

Что мы собсно и наблюдаем.

Nov 14 17:45:54 tsotimus kernel: ide3: BM-DMA at 0xac08-0xac0f, BIOS settings: hdq:pio, hdh:pio

Nov 14 17:45:54 tsoptimus kernel: Probing IDE interface ide2...

Nov 14 17:45:54 tsoptimus kernel: hde: FUJITSU MPG3204AT E, ATA DISK drive

Nov 14 17:45:54 tsoptimus kernel: ide2 at 0x9c00-0x9c07,0xa002 on irq 17

Nov 14 17:45:54 tsoptimus kernel: Probing IDE interface ide3...

Nov 14 17:45:54 tsotimus kernel: hda: max request size: 128KiB

Nov 14 17:45:54 tsoptimus kernel: hda: 78165360 sectors (40020 MB) w/2048KiB Cache, CHS=65535/16/63, UDMA(100)

△△△△△△△△

Говорит само за себя

Λ Λ

До этого было supported, разберусь...

AA

```
CONFIG IDE TASKFILE IO=y
```

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Не разбирался :)

#

# IDE chipset support/bugfixes

#

# CONFIG\_IDE\_GENERIC is not set

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Не нужен если Ваши IDE контроллеры известны ядру. У 99% людей они известны. Посему выключен.

# CONFIG\_BLK\_DEV\_CMD640 is not set

CONFIG\_BLK\_DEV\_IDEPCI=y

CONFIG\_IDEPCI\_SHARE\_IRQ=y

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Разрешает устройству совместно использовать одно прерывание с другими устройствами.

В случае проблем можно выключить.

# CONFIG\_BLK\_DEV\_OFFBOARD is not set

# CONFIG\_BLK\_DEV\_GENERIC is not set

# CONFIG\_BLK\_DEV\_OPTI621 is not set

# CONFIG\_BLK\_DEV\_RZ1000 is not set

CONFIG\_BLK\_DEV\_IDEDMA\_PCI=y

# CONFIG\_BLK\_DEV\_IDEDMA\_FORCED is not set

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

ОПАСНО!!!! Если ваш винт или контроллер совсем никак не умеет DMA или blacklisted, то есть риск потерять раздел.

CONFIG\_IDEDMA\_PCI\_AUTO=y

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

А вот это значительно нежнее :) Если с винтом и контроллером всё в порядке, ядро само включит Вам DMA.

# CONFIG\_IDEDMA\_ONLYDISK is not set

Ежели ваш CDROM глючит с DMA или работает медленнее чем в PIO режиме, ядро включит DMA только для винчестеров.

# CONFIG\_BLK\_DEV\_AEC62XX is not set





```
CONFIG_IDEDMA_AUTO=y  
# CONFIG_BLK_DEV_HD is not set
```

Ускоряемся :)

Сделайте, если ещё не сделали,

```
emerge hdparm
```

На одной консоли сделайте

```
man hdparm
```

А на другой мы начнём играть :) )

Делайте

```
hdparm -tT /dev/hdX
```

Где X - буква вашего винчестера

Мы запустили тест скорости чтения из кеша и с пластины. Для усреднения результатов запустите его несколько раз.

Должны увидеть следующее:

```
/dev/hda:
```

```
Timing cached reads: 616 MB in 2.00 seconds = 307.74 MB/sec
```

```
Timing buffered disk reads: 74 MB in 3.04 seconds = 42.33 MB/sec
```

Первый результат с учётом программного кеширования, он у всех большой :)

Второй - собственно чтение с диска с учётом аппаратного кеширования.

Второй результат мы собственно и будем улучшать :) Цифры в районе 35-55 mb/s хороший результат, выше 50 без raid контроллера вы на параллельном интерфейсе вряд ли получите (хотя... кто знает...). Но в любом случае кроме скорости можно

Разбираемся с железом на железнном уровне

Имеет смысл заглянуть в BIOS и в корпус.

**ВНИМАНИЕ!!!** Если вы страдаете хронической неуверенностью, криворукостью, боитесь потерять гарантию, пропустите этот раздел!

Иногда сборщики компьютеров страдают излишней жадностью или криворукостью, поэтому всегда имеет смысл проверить, всё ли внутри корпуса как надо.

Каждый канал IDE/ATA интерфейса поддерживает подключение двух устройств - master и slave. Конфигурация обычно задается переключком на устройстве. Кроме этих двух позиций там обычно присутствует и третья - cable select. Для работы устройств в положении переключки cable select требуется специальный Y-образный шлейф, центральный разъем которого подключается к материнской плате. Крайние разъемы такого кабеля неравноправны - устройство, подключенное к одному разъему, автоматически становится master, к другому - slave.

Проверьте, чтобы шлейфы были 80-жильные, на 40-жильных вы не сильно ускоритесь :).

Каждый канал в каждый момент времени может обрабатывать только один запрос к одному устройству. Следующий запрос, пусть даже к другому устройству, будет ожидать завершения текущего. Разные каналы при этом могут работать независимо. Поэтому не стоит подключать два активно используемых устройства (например, два жестких диска), к одному каналу.

В идеале каждое IDE-устройство стоит подключать к отдельному каналу (в этом заключается основное преимущество SATA).

Практически все современные чипсеты поддерживают возможность использования различных режимов передачи данных для устройств, подключенных к одному каналу. Однако злоупотреблять этим все-таки не стоит. Два устройства, существенно различающихся по скорости, лучше все-таки разнести по разным каналам.

Не рекомендуется подключать к одному каналу жесткий диск и ATAPI-устройство (например, CD-ROM). Как было сказано выше, протокол ATAPI использует другую систему команд, и, кроме того, даже самые быстрые ATAPI-устройства намного медленнее жесткого диска, что может замедлить работу последнего.

Идеальная конфигурация.

Каждый винт и CD-ROM на отдельном шлейфе.

Приемлемая конфигурация.

На первом канале основной винт, на втором дополнительный master и CD-ROM slave

Тоже неплохо.

На первом канале два винчестера одного поколения, CD-ROM на втором.

Плохо!!!

На одном канале современный винт и древняя древность на полтора гига :)

Смотрим в BIOS. Включаем UDMA где только можно, включаем Bus master для контроллеров IDE, включаем IDE Block mode.

Разбираемся с железом на софтверном уровне

А что умеет наш винчестер? И что у него включено сейчас?

Делаем

```
hdparm -ilv /dev/hdX | less
```

Получаем длииинный листинг с описанием возможностей нашего винчестера. Пример моего с комментариями приведён ниже.

Эта секция коротко описывает, что у нас включено прямо сейчас.

/dev/hda:

multcount = 16 (on)

IO\_support = 1 (32-bit)

unmaskirq = 1 (on)

using\_dma = 1 (on)

keepsettings = 0 (off)

readonly = 0 (off)

readahead = 256 (on)

geometry = 65535/16/63, sectors = 40020664320, start = 0

Эта секция показывает необработанные для читабельности данные, прочитанные напрямую с винчестера.

Model=ST340016A, FwRev=3.19, SerialNo=3HS9R2GG

Config={ HardSect NotMFM HdSw>15uSec Fixed DTR>10Mbs RotSpdTol>.5% }

RawCHS=16383/16/63, TrkSize=0, SectSize=0, ECCbytes=4

BuffType=unknown, BuffSize=2048kB, MaxMultSect=16, MultSect=16

CurCHS=4047/16/255, CurSects=16511760, LBA=yes, LBASects=78165360

IORDY=on/off, tPIO={min:240,w/IORDY:120}, tDMA={min:120,rec:120}





Собственно настраиваем

Итак, параметры мы передаём как

hdparm -параметр /имя/диска

Если hdparm ругается, значит или режим не поддерживается, или параметр не верный. Если режим не поддерживается, команда dmesg покажет нам последней строчкой ругань драйвера.

Интересные параметры

-а Количество секторов упреждающего чтения для файловой системы, то есть софтверное упреждающее чтение. Обычно значение по дефолту приемлемое, а наличие аппаратного упреждающего чтения смазывает эффект. Чем меньше, тем лучше в случае чтения кучи мелких файлов разбросанных хаотично по диску. Чем больше число, тем лучше для копирования фильмов и mp3.

-А Включение аппаратного упреждающего чтения винчестером. Обычно всегда включено по дефолту. Изменять следует, если в вышеописанном информационном выводе не стоит звёздочка перед Look-ahead в разделе Enabled Supported. 1 - включено, 0 - выключено.

-В Имеет смысл для ноутбуков, управляет настройкой энергосбережения. Чем значение ниже, тем больше у винта желание остановить блин и заснуть :), от чего растёт время отклика. Значение 255 вырубает управление питанием, соответственно уменьшая время отклика. Не все винчестеры имеют собственное управление питанием. Проверьте параметр AdvancedPM= в коротком листинге с сырыми данными.

-с Значение 1 включает 32 битную передачу по внутренней шине, по шлейфу возможно только 16 бит. Значение 3 включает то-же самое с контролем чётности. Это более надёжно, но чуть медленнее чем 1. По умолчанию всегда выключено, имеет смысл включить. разницы между 1 и 3 не заметил ни по скорости, ни по надёжности.

-d 1 - включено, 0 - нет. Признак использования DMA. Если у Вас стоит звёздочка напротив одного из dma режимов в вышеприведённых листингах, значит Вам этот параметр не нужен. Если не стоит, можно попробовать включить. ВНИМАНИЕ!!! Машина может зависнуть, раздел может не вернуться!!! Если это случилось, что-то крепко не в порядке с железом или ядром!!!

-Е Скорость sdrom. Если сильно воет... Просто укажите желаемую скорость. Для

винтов бесполезен :)

-k и -K Сохранение настроек и опций винчестером. Это не означает что они автоматически сожранятся после перезагрузки. Это означает, что они сохранятся, если драйвер сделает reset контроллеру, или после спячки, но не гибернации (aka suspend to disk). 1 - сохранять. 0 - нет.

-m Количество секторов для аппаратного упреждающего чтения. Не может быть больше чем умеет винт. Смотреть в диагностический листинг, параметр MaxMultSect=. По дефолту обычно всегда максимум.

-M Управление акустикой. 254 - шумно и быстро. 0 - медленно и тихо. Многие драйвы умеют только 128 и 254. Многие вообще нисколько не умеют :)

-P Ещё одна ручка для регулирования аппаратного упреждающего чтения. Не работает ни на одном виденном мною винте. Может у Вас заработает?

-u Включить размаскирование прерываний. Если значение 1, то контроллер разрешает другие прерывания во время операции ввода-вывода. Это снижает нагрузку на систему и повышает отклик. Нужно включать. По умолчанию выключено. На очень древних контроллерах система может повиснуть.

-W 1 - включает кеширование записи. Включено по дефолту у всех виденных мною винтов. Но пригодится может.

-X Самый мощный параметр. Позволяет вам принудительно выставить режим работы DMA. В последних версиях hdparm задаётся буквенно, например -X udma5. Внимание!!! Не включайте режимы, не поддерживаемые Вашим контроллером, или с плохим (40-жильным) шлейфом!!! Обычно ядро само выбирает максимальный режим DMA, и если оно не смогло, или выбрало как Вам кажется меньший режим чем можно, ПОДУМАЙТЕ!!! Вдруг тому есть ОБЪЕКТИВНЫЕ причины?.

Пробуйте параметры по одному, запускайте hdparm -tT, смотрите в dmesg.

Сидиромы понимают не все параметры из указанных выше. У разных винчестеров понимание тоже может отличаться :)

После того как наиграетесь, будем закреплять.

Закрепляем

В /etc/conf.d/hdparm заносим понравившиеся параметры. У меня это выглядит так:

```
hda_args="-u1c3M254Kk"
```

```
hde_args="-u1c3M254Kk"
```

```
#hdd - cdrom
```

```
hdd_args="-u1c3Kk"
```

Делаем: rc-update add hdparm default

Затем: /etc/init.d/hdparm start

## Выбор планировщика ввода-вывода

В ядрах 2.6 появилась возможность выбирать между четырьмя планировщиками ввода-вывода. У каждого планировщика есть свои достоинства и недостатки. По умолчанию в ядро всунуты все четыре, и выбран anticipatory io scheduler. Кратенько опишу их.

no-op - очень мелкий и лёгкий планировщик. Мало чего умеет. Для винтов не пригоден. В основном используется если вместо винта - флеш.

anticipatory - выбирается по умолчанию. Середина-наполовину для всего на свете. И там хорош, и тут хорош.

deadline - более лёгкий и простой чем anticipatory, лучше себя ведёт при "взрывных" нагрузках. При равномерной нагрузке имеет особенность задумываться и притормаживать.

CFQ - размазывает ввод-вывод равномерно между всеми процессами. Ввод-вывод медленный, но плавный и равномерный независимо от загрузки. Это мой выбор на сегодня.

## Как попробовать?

Передайте ядру в строке загрузки GRUB или другого загрузчика параметр elevator=[cfq|as|deadline|noop], поработайте, сравните. Ненужные планировщики ввода/вывода можно убрать из конфигурации ядра перед его компиляцией. Подробнее можно узнать в статье "Компиляция ядра Linux". Они в

general setup ->

Configure standart kernel features...

Какнибудь потом расскажу про тюнинг vm и свопа ...



## **Тонкая настройка IDE дисков с помощью *hdparm***

### Введение

К сожалению настройки по-умолчанию для IDE жёстких дисков сделаны с уклоном в надёжность работы, в том числе на некачественном оборудовании. На большинстве же современных материнских плат и жёстких дисков можно заметно увеличить производительность IDE подсистемы, причём это не потребует много времени.

### Характеристики IDE устройства

Для получения характеристик IDE устройства выполните команду:

```
``hdparm -i /dev/hda``
```

Вместо /dev/hda можно указать любое другое IDE устройство

Пример результата для HDD

Пример результата для CD-RW

### Расшифровка

Max Mult Sect?: Это поле содержит максимальное число секторов, которое ваш жесткий диск может прочесть за «один присест».

Mult Sect?: Текущая настройка количества секторов, считываемых за один раз.

PIO modes и DMA modes: Здесь перечисляются режимы, которые поддерживает жесткий диск. Режим, помеченный символом «звездочка» (\*), является текущим.

Advanced PM?: Признак 'yes' указывает на то, что жесткий диск поддерживает APM (Advanced Power Management – Расширенное Управление Питанием).

Запустив другую команду, вы получите дополнительную информацию:

```
darkstar:$ hdparm /dev/hda
```

Результат работы команды:

/dev/hda:

multcount = 0 (on)

I / O? support = 0 (16-bit)

unmaskirq = 0 (off)

using\_dma = 0 (off)

keepsettings = 0 (off)

nowerr = 0 (off)

readonly = 0 (off)

readahead = 8 (on)

geometry = 2482/255/63, sectors = 39876480, start = 0

Коротко...

multcount – число секторов, считываемых одновременно;

I / O? support – режим обмена с жестким диском (16/32/32sync);

using\_dma – указывает, используется ли в настоящий момент режим DMA или нет;

keepsettings – указывает, сохраняются ли настройки после программного сброса (изменять эту настройку не рекомендуется, если вы не уверены в том, что делаете);

readonly – обычно установлен в 1 только для CD-ROM, этот флаг сообщает системе – является ли устройство read-only или нет;

readahead – количество секторов для опережающего чтения;

Настройка

**ПРЕДУПРЕЖДЕНИЕ!** Неправильная настройка может привести к потере данных на жёстком диске, а возможно и самого дика.

I / O? Support

-c0 — установка 16-и битного режима (по-умолчанию)\

-c1 — установка 32-х битного режима (обычно то, что вам нужно)\

-с3 — установка 32-х битного синхронного режима (мне не известно, когда именно это над

## Multicount

Количество секторов передаваемых сразу (одним блоком). Степень двойки, максимальное число — параметр Max Mult Sect? в выводе hdparm -i. Чаще всего вам именно его и нужно указывать.

## Включение DMA

Большинство современных чипсетов и HDD поддерживают DMA под линуксом. Для некоторых самых новых чипсетов в данный момент поддержки DMA может не быть, однако это не так часто бывает. Обычно в течении нескольких недель после выхода нового чипсета появляется его поддержка в последнем ядре, так что если DMA у вас не включается, то следует, скорее всего, обновить ядро.

-d0 — запрещение DMA\

-d1 — включение DMA

## DMA mode и PIO mode

Самая опасная настройка — её неправильное использование чаще всего приводит к повреждению данных или оборудования.

-X<число> — установка режима PIO или DMA

## Режимы multiword DMA:

32 (базовое число) + номер режима:

-X32 — mdma0

-X33 — mdma1

-X34 — mdma2

Для PIO базовое число 8, для Ultra DMA? 64.

-X64 — udma0

-X65 — udma1

-X66 — udma2

-X67 — udma3

-X68 — udma4

-X69 — udma5

Readahead (опережающее чтение)

На сколько чекторов вперёд читать при запросе. Ускоряет производительность при чтении файлов большого размера, однако замедляет работу с файлами небольшого размера.

Лучше всего установить то же значение, что и для multicount.

-aN — опережающее чтение N секторов

Тестирование

Для тестирования наберите:

```
hdparm -t <имя устройства>
```

Желательно это делать, когда заметной дисковой активности нет.

Потом попробуйте какой-нибудь вариант конфигурации, например:

```
hdparm -u1c1d1m8a8 <имя устройства>
```

После чего опять повторите:

```
hdparm -t <имя устройства>
```

И сравните результаты.

По результату этого теста не стоит подбирать опции -a и -m, для этого лучшим тестом будут реальные приложения, исполняемые на вашей машине. Если это домашняя машина, а не боевой сервер, на котором важна свертонкая подстройка (но на таких серверах редко оказываются IDE-диски, о которых здесь идёт речь), то тогда параметр multicount (-m) ставите на максимум, а параметр read-ahead (-a)

равным multicount.

## **Работа с CD/DVD**

Основные программы

Для начала, убедитесь что у вас установлены программы cdrecord и mkisofs, которые входят в пакет cdrtools. Если нет -- установите его:

```
emerge cdrtools
```

Конфигурация ядра и загрузчика

Убедитесь, что у вас ядро настроено соответствующим образом.

Ядра серии 2.4.x

Для ядер серии 2.4.x в ядро необходимо включить поддержку SCSI эмуляции: Linux Kernel Configuration: Конфигурация ядра 2.4.x

ATA/IDE/MFM/RLL Support -->

IDE, ATA, and ATAPI Block Devices -->

<M> SCSI Emulation Support

SCSI Support -->

<M> SCSI Support

<M> SCSI CDROM Support

<M> SCSI Generic Support

Теперь, еще необходимо ядру при загрузке передать параметр hdc=ide-scsi (в данном случае предполагается, что ваш CD привод - это hdc).

Т.е. если вы используете grub, то его конфиг должен выглядеть примерно так:  
Файл: /boot/grub/grub.conf

```
title Gentoo Linux
```

```
root (hd#,#)
```

```
kernel (hd#,#)/YOUR_KERNEL_NAME_HERE root=/dev/hdx# hdc=ide-scsi  
ANY_OTHER_KERNEL_OPTIONS_NEEDED
```

Если вы используете lilo, то соответствующая часть конфига: Файл: /etc/lilo.conf

# Linux bootable partition config

```
image = /boot/YOUR_KERNEL_NAME_HERE
append = "hdc=ide-scsi"
root = /dev/hdx#
label = Gentoo
read-only
```

Ядра серии 2.6.x

В ядрах серии 2.6.x вам уже нет необходимости включать SCSI (и, следовательно, не надо переконфигурировать загрузчик). Вместо этого, надо указать поддержку ATAPI CDROM: Linux Kernel Configuration: Конфигурация ядра 2.6.x

Device Drivers -->

ATA/ATAPI/MFM/RLL support -->

<\*> Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support

<\*> Include IDE/ATAPI CDROM support

Подготовка ISO образа

Для начала, создайте папку: `mkdir /tmp/iso`

а затем скопируйте все файлы, которые вы хотите записать, в созданную папку. с помощью `*.doc /tmp/iso`

Затем создайте image.iso `mkisofs -o /tmp/image.iso /tmp/iso`

Дополнительные опции при создании ISO образа

-r : добавляет (частично) поддержку так называемых расширений Rock Ridge, т.е. сохраняет специфичные для unix атрибуты файлов. Например, добавляет поддержку длинных имён (до 255 символов, но не для MS Windows), символических ссылок. В отличие от опции -R, которая полностью соответствует Rock Ridge, права доступа на файлы не сохраняются, они становятся доступны для чтения любому пользователю.

-jcharset=koi8-r : подразумевает под собой 2 опции: 1) -J включает поддержку Joliet, которая используется для распознавания длинных имён файлов под MS Windows; 2) -input-charset устанавливает перекодировку символов для корректного отображения русских имён файлов. ("koi8-r" здесь дано для примера - замените на кодировку, используемую в вашей системе)

-joliet-long : Allow Joliet filenames to be up to 103 Unicode characters. This breaks the Joliet specification - but appears to work. Use with caution.

-f : следование символическим ссылкам. Если вместо копирования файлов в /tmp/iso, вы просто создаёте в этой директории симлинки на них, то вы должны включить эту опцию.

-C : эта опция необходима, если вы хотите дописать мультисессионный диск (см. раздел [Как дописывать диски])

## Определение ATAPI-приводов

Выполните `cdrecord --dev=ATAPI --scanbus`

Вы должны увидеть нечто похожее на следующее:

Cdrecord-Clone 2.01a25 (i686-pc-linux-gnu) Copyright (C) 1995-2004 J?rg Schilling

scsidesv: 'ATAPI'

devname: 'ATAPI'

scsibus: -2 target: -2 lun: -2

Warning: Using ATA Packet interface.

Warning: The related libscg interface code is in pre alpha.

Warning: There may be fatal problems.

Using libscg version 'schily-0.8'.

scsibus0:

0,0,0 0) 'SONY ' 'CD-RW CRX175A1 ' '5YS2' Removable CD-ROM

0,1,0 1) \*

0,2,0 2) \*

0,3,0 3) \*

0,4,0 4) \*

0,5,0 5) \*

0,6,0 6) \*

0,7,0 7) \*

Здесь показано, что на шине 0,0,0 есть CD-RW привод. Предупреждение: В новых версиях cdrtools (2.01.01) и ядра 2.6.x надо писать `cdrecord --dev=ATA --scanbus`

## Запись ISO образа

`cdrecord --dev=ATA:0,0,0 /tmp/image.iso`

(Замените шину 0,0,0 на необходимую) Примечание: Вместо dev=ATA:0,0,0 можно указывать просто имя устройства, например, dev=/dev/cdrw

#### Дополнительные возможности

Используйте опцию --speed=# для указания скорости записи

```
cdrecord --dev=ATAPI:0,0,0 --speed=4 /tmp/image.iso
```

Используйте --blank=fast для быстрой очистки CD-RW диска.

```
cdrecord --dev=ATAPI:0,0,0 --blank=fast
```

Большинство современных записывающих устройств поддерживает технологию BURN-free. К сожалению она не включена по умолчанию при запуске cdrecord. Для включения, добавьте опцию --driveropts=burnfree. Вы можете посмотреть, какие ещё опции поддерживает ваш CD-ROM посредством команды `cdrecord dev=/dev/cdrw driveropts=help -checkdrive`

Если в файле /etc/default/cdrecord установить переменную CDR\_DEVICE=ATAPI:0,0,0 (можно и CDR\_DEVICE=/dev/hdc), то не придется каждый раз указывать устройство (ключ --dev=ATAPI:0,0,0) и тогда для очистки CD-RW диска понадобится набрать только следующее:

```
cdrecord --blank=fast
```

Если нужно разрешить пользователю запись на CD, то включите его в группу cdrom.

#### Копирование дисков

Если у вас всего один CD-ROM, то необходимо вначале создать файл-образ диска. Это можно сделать многими способами (с помощью dd, cat, cdrdao, readcd и т.д.). Лучше воспользоваться readcd, т.к. он уже входит в пакет cdrtools, и имеет множество интересных опций (смотри man readcd): `readcd dev=/dev/cdrw -clone f=/tmp/image.iso`

После этого запишем образ на болванку с помощью cdrecord: `cdrecord -v -eject dev=/dev/cdrw -raw96r -clone /tmp/image.iso`

**ВАЖНО:** Ключи -raw96r и -clone необходимы ТОЛЬКО если вы создавали файл-образ с помощью команды readcd -clone.

#### Как дописывать диски

Прежде всего, при создании и последующем дописывании мульти-сессийных дисков, необходимо всегда для cdrecord указывать опцию -multi, а для mkisofs



использовать ключ -r или -R (т.е. включать расширение Rock Ridge). Таким образом создание первой сессии будет выглядеть примерно так: Code: Запись первой сессии

```
mkisofs -r -f -jcharset=koi8-r -o /tmp/image.iso /tmp/iso/
```

```
cdrecord -v -multi driveropts=burnfree -eject dev=/dev/cdrw /tmp/image.iso
```

Для создания последующих сессий, нам нужна информация о номере начального сектора последней сессии и о номере сектора с которого будет начинаться новая сессия. Эта пара чисел выводится командой `cdrecord -msinfo dev=/dev/cdrw`. После этого данные числа указываются в команде `mkisofs` в качестве аргументов к опции -C. Также, чтобы были видны файлы из предыдущей сессии, необходим ключ -M, аргументом к ключу -M служит имя устройства вашего CD-ROM (т.е. в нашем примере - это `/dev/cdrw`). Команда `cdrecord` остаётся такой же. Резюмируя вышесказанное получаем для записи последующей сессии: Code: Запись последующих сессий

```
mkisofs -r -f -jcharset=koi8-r -C $(cdrecord dev=/dev/cdrw -msinfo) -M /dev/cdrw -o /tmp/image.iso /tmp/iso/
```

```
cdrecord -v -multi driveropts=burnfree -eject dev=/dev/cdrw /tmp/image.iso
```

#### Полезные ссылки

Классическое how-to по записи дисков на английском языке: <http://www.tldp.org/HOWTO/CD-Writing-HOWTO.html>

То же самое на русском: <http://www.opennet.ru/docs/HOWTO-RU/CD-Writing-HOWTO.html>

Создание аудио дисков (на английском языке): en:HOWTO Create an Audio CD

## **Создание DVD**

### Введение

#### О чем эта статья

В данной статье будет подробно рассмотрен вопрос формирования и записи видео дисков формата DVD в Gentoo Linux.

#### Необходимые программы

Для начала, убедитесь что у вас установлены следующие программы:

transcode

emerge -n transcode

cdrecord и mkisofs

emerge -n cdrtools

growisofs

emerge -n dvd+rw-tools

dvdauthor

emerge -n dvdauthor

также могут пригодиться:

kino

emerge -n kino

cinelerra

emerge -n cinelerra-cvs

dvdrrip

USE='cdr mplayer ogg subtitles xvid rar' emerge -n dvdrrip

replex

emerge -n replex

cdrecord-prodvd

emerge -n cdrecord-prodvd

Кодирование видео в формате DVD

Использование transcode для декодирования видео произвольного формата и кодирования его в формат MPEG2, совместимый с DVD

Хотелось бы заметить, что если вы не любители программ с кучей непонятных опций, то вам следует попробовать программу dvdrrip, которая является неплохим

фронтендом для transcode с понятным интерфейсом.

Transcode, пожалуй, лучший инструмент для нашей задачи. Вам, возможно, придётся заглянуть в man страницу по transcode, чтобы понять те команды, которые используются здесь. Также вам могут пригодиться и другие инструменты для декодирования и кодирования вашего формата, так что не поленитесь заглянуть и в их man странички. Здесь будет приведен пример того, как конвертировать стандартный divx файл в широкоэкранный формат (без черных полос сверху и снизу). Затем мы конвертируем все это в формат letterbox 1.5:1. Зачем, спросите вы. Затем, что изначально широкоэкранный формат имеет соотношение 1.73 (16:9) с разрешением 720x416. DVD в свою очередь поддерживает размеры изображения только 720x480, 704x480, 352x480 и 352x240 (возможно это не все). Поэтому самое простое и качественное решение здесь - это привести наше видео к формату 720x480 (соотношение 1.5), добавив по 32 черных полосы сверху и снизу к изображению.

```
transcode -i test-divx.avi -V -y mpeg -F d -Y -32,0,-32,0 -E 44100 -b 128 -o showgirls-dvd
```

Отрицательные значения для параметра -Y сообщают transcode о том, чтобы добавить нужное количество черных полосок сверху и снизу. Если использовать положительные значения, то полосы будут соответственно убираться сверху и снизу. Часто вам может попасться видео уже в формате 720x480 или еще каком-то "хорошем" формате, тогда не нужно изменять его размеры, то есть просто не используйте опцию -Y.

Here's what some of those command line arguments mean: Вот, что означают остальные опции:

-V использовать YV12/I420 в качестве встроенного видео кодека [выключено изначально]

Это, как правило, намного быстрее чем RGB процессинг, но некоторые импортированные модули могут не поддерживать этот формат.

Используйте эту опцию всегда, когда это возможно.

-y vmod[,amod]

видео[,аудио] модули для экспортирования [отсутствуют изначально].

Если опустить, то transcode не будет кодировать никаким модулем.

Полностью установленный transcode имеет следующие модули:

<вырезано>

mpeg - (видео) MPEG 1/2 | (аудио) MPEG 1 Layer II

asm должен быть доступен во время компиляции. Этот модуль не имеет зависимостей. Этот модуль хорошо поддерживается.

Интерфейс к библиотеке libmpeg (включена в transcode). Может кодировать в mpeg1, VCD, SVCD, MPEG2 и DVD видео форматы.

Кодированное видео является отдельным файлом, который следует объединить с соответствующим аудио файлом после кодирования.

-F codec\_string

параметры энкодера [зависят от модуля].

Параметр -F имеет разные значения для разных модулей.

<вырезано>

-y mpeg:

Синтаксис: -F "<base-profile>[,<resizer-mode>[,user-profile]]"

<base-profile> может быть одним из:

'1' = MPEG 1 (по умолчанию)

'b' = big MPEG 1 (экспериментальный)

'v' = VCD

's' = SVCD

'2' = MPEG2

'd' = DVD совместимый

<resizer-mode> может быть одним из:

0 = выключить resizer (по умолчанию)

1 = 352x288

2 = 480x480

3 = 480x576

4 = 352x240

<user-profile> Имя файла профиля. Вы можете

можно указать как абсолютный путь так и относительный

(относительно директории из которой запускаете transcode).

-Y top[,left[,bottom[,right]]]

определить (для энкодера) рамку по которой будет обрезано изображение.

Отрицательные значения будут добавлять рамку [отключено изначально].

-E r[,b[,c]]

частота сэмплирования выходного аудио [Гц], битов на сэмпл и каналы

[входящие данные]. Опция "-J resample" должна быть передана

не совместимым с ресэмплингом.

-B n[,m[,M]]

изменить размеры до высота-n\*M горизонтальных полосок

[,ширина-m\*M] вертикальных полосок [отключено изначально,32].

M должно иметь одно из значений 8, 16 или 32. Неважно какое M

вы используете. Также взгляните на флаг fast опции -Z если

не хотите считать n и m вручную.

-b b[,v[,q[,m]]]

битрейт аудио энкодера кБит/с [,vbr[,quality[,mode]]] [128,0,5,0]

Параметр mode определяет какие модули будет использовать lame при кодировании. Возможные значения для mode:

- 0 Joint Stereo (по умолчанию)
- 1 Full stereo
- 2 Mono

-o file

имя выходного файла, по умолчанию [/dev/null].

Если бы, к примеру, наше видео имело формат 728x424, то следовало бы воспользоваться командой типа следующей:

```
transcode -i showgirls-divx.avi -V -y mpeg -F d -B 1,1,8 -E 44100 -b 128 -o showgirls-dvd
```

Также обратитесь к опции -X для того, чтобы увеличить размеры видео.

Использование tcplex для объединения аудио и видео данных

Transcode должен был создать \*.m2v (mpeg-2 видео) и \*.mpa (аудио) файлы в текущей директории. Теперь мы их объединим так:

```
tcplex -o showgirls-dvd.vob -i showgirls-dvd.m2v -p showgirls-dvd.mpa -m d
```

Использование kino (альтернатива transcode+tcplex)

Kino используется для получения видео с dv камер. Эта программа имеет возможности простейшего видеомонтажа (вырезать сцену, вставить, поменять местами). Она имеет удобный интерфейс для захвата видео (хотя вы можете использовать dvgrab для этой цели). Она использует формат qt4linux mov по умолчанию для захвата.

Для более продвинутого монтажа cinelerra, пожалуй, пока единственный выбор под Linux. Это достаточно мощная, но достаточно долгая в освоении и пока что нестабильная (сохраняйтесь как можно чаще) программа. Вы можете просто импортировать тот же .mov полученный с помощью kino (dvgrab) в cinelerra (на самом деле, она падает при работе с большинством других форматов). Затем вы можете использовать cinelerra, чтобы привести оконченный фильм обратно к формату mov.

Как только вы наконец получите тот .mov файл, какой вы хотели, загрузите его опять в kino и перейдите в закладку export. Здесь вы уже можете экспортировать video в формат vcd, svcd или dvd. Формат DVD потребует достаточно много времени для конвертирования (желателен мощный процессор).

Использование replex (подготовка DVB записей с цифрового телевидения (Digital

Television))

"Эти инструкции были испробованы на записях формата DVB-C, сделанных в Финляндии"

Примечание: пожалуй, для России эта глава абсолютно бесполезна, но если есть большое желание, то я переведу и её. Мало ли, кто где живет ;)

Digital television streams are transmitted as MPEG2-TS in which the TS stands for Transport Stream. This means that extra information, such as multiple audio streams, could be transferred along with the actual video. Quite fortunately, the video stream is already suitable for DVD without re-encoding which is both time consuming and stupid as it lowers the quality. Enter replex. First use czap for tuning into some channel:

```
czap -r -n 3
```

This tunes the DVB card to channel number 3. Now, record some data by typing:

```
cat /dev/dvb/adapter0/dvr0 > stream.ts
```

This simply writes raw data from the first tuner card into a file. After awhile press CTRL-C to stop recording.

Now, you should have a blob of data which could be played with e.g. mplayer. Next, you should use replex to transform TS format suitable for DVD. Enter following command:

```
replex -k -i TS -t DVD -o stream.dvd stream.ts
```

This command ignores possible errors in original stream (-k), assumes that input file is in TS format (-i TS), outputs format suitable for DVD (-t DVD), writes to a file called stream.dvd (-o stream.dvd) and reads its input from the file entered as a last parameter (stream.ts).

After a moment a file called stream.dvd should appear in the same directory where above command was entered. Now just follow the instructions from the next chapter onwards to burn this baby on a DVD.

Создание структуры директорий (authoring) с использованием dvdauthor

Это самый сложный этап! Имея готовый MPEG2 файл, воспользуйтесь программой dvdauthor (или вашим любимым графическим фронтендом к ней (например, dvdstyler)), чтобы создать структуру директорий для записи DVD (VIDEO\_TS и AUDIO\_TS).

```
dvdauthor -o . showgirls-dvd.vob && dvdauthor -o . -T
```

## Создание DVD образа для записи

Для более подробной информации смотрите здесь: [\[\[1\]\]](#)

Если директории AUDIO\_TS и VIDEO\_TS созданы у вас в директории bar, которая в свою очередь находится в текущей директории, то введите команду:

```
mkisofs -dvd-video -o ./bar.img ./bar/
```

и вы получите образ bar.img в текущей директории, который теперь следует записать на диск вашей любимой программой для записи дисков.

Или можете записать директории AUDIO\_TS и VIDEO\_TS прямо из каталога bar/ набрав:

```
growisofs -Z /dev/dvd -dvd-video -V название_диска .
```

Где /dev/dvd - ваш DVD-writer.

## Запись

В Linux существует множество способов записи DVD. Из уже показанных можно использовать cdrecord или growisofs.

С помощью графического интерфейса

Формат ISO можно записать с помощью программы X-CDroast (она использует cdrecord-ProDVD, которая требует ключ, что доставляет определенное неудобство).

Также можно воспользоваться программой k3b, которая использует growisofs. Это, пожалуй, самый простой способ.

Из командной строки

Можно записать образ ещё и так:

```
cdrecord-ProDVD -dao -speed=4 -dev=/dev/dvd ./bar.img
```

Воспользуйтесь командой cdrecord -scanbus предварительно, чтобы определить какое устройство использовать для записи. Для устройств ATAPI используйте формат -dev=/dev/dvd или -dev=/dev/hdd (зависит от машины), для эмуляции SCSI

устройств используйте `-dev=0,0,0` или `-dev=0,0,1` (зависит от машины).

Если у вас нет `cdrecord-ProDVD` можете использовать `growisofs`.

```
growisofs -dvd-compat -Z /dev/dvd= ./bar.img
```

Подсказка: С помощью `growisofs` вы можете писать диски налету (без создания четырёхгигабайтного файла `.iso`):

```
growisofs -dvd-compat -Z /dev/dvd -dvd-video ./bar
```

Можно также воспользоваться скриптом, который расположен по адресу <http://james.nontrivial.org/projdvd.htm>

Внешние ссылки

## ***Система Portage***

### **Portage через NFS**

Проблемы

Слишком долгое время выполнения `'emerge --sync'` на всех серверах в LAN.

Согласно этикету `gentoo`, вы не можете синхронизироваться чаще одного раза в день

Решение

Общий NFS portage. Одна машина синхронизируется под управлением `cron`, остальные машины монтируют свою директорию `/usr/portage` с помощью NFS.

Общее представление

1. Обязательная поддержка `nfs` в ядре. В случае компиляции ее как модуля, добавить соответствующие строки в `modules.autoload`
2. Если потребуется, перезагрузить или загрузить модуль `nfs`
3. Выполнить `emerge nfs-utils`



4. Настроить NFS/fstab для автоматического монтирования общего репозитория portage.

5. Установить локальный PORTAGE\_TMPDIR на локальную файловую систему.

#### Выгоды

Только один сервер заботится о синхронизации portage (ночью по заданию cron'a).

Не нужен локальный rsync сервер.

Общий для всех каталог /usr/portage/distfiles означает, что вам не нужно скачивать одно и то же каждый раз, чтобы синхронизировать репозиторий portage на ваших компьютерах.

И вообще, NFS - это классная штука ;)

Предупреждение: NFS является очень скудным протоколом по части безопасности. Убедись, что используете его только в пределах вашей локальной сети, где вы полностью доверяете вашим пользователям. Мы вас предупредили!

#### Подробный процесс

#### Поддержка ядра

У вас ДОЛЖНА быть включена поддержка NFS в вашем ядре, либо вкомпиленная в ядро (в таком случае необходимо будет перезагрузиться): Linux Kernel Configuration: Встроенная поддержка

File systems --->

Network File Systems --->

<\*> NFS file system support

[\*] Provide NFSv3 client support

[ ] Provide NFSv4 client support (EXPERIMENTAL)

[ ] Allow direct I/O on NFS files (EXPERIMENTAL)

<\*> NFS server support

[\*] Provide NFSv3 server support

[ ] Provide NFSv4 server support (EXPERIMENTAL)

[ ] Provide NFS server over TCP support (EXPERIMENTAL)

либо в качестве модулей: Linux Kernel Configuration: Включение модулей

File systems --->

Network File Systems --->

<M> NFS file system support

[\*] Provide NFSv3 client support

[ ] Provide NFSv4 client support (EXPERIMENTAL)

[ ] Allow direct I/O on NFS files (EXPERIMENTAL)

<M> NFS server support

[\*] Provide NFSv3 server support

[ ] Provide NFSv4 server support (EXPERIMENTAL)

[ ] Provide NFS server over TCP support (EXPERIMENTAL)

Добавляем NFS server support, так как на данной машине будет размещаться общий каталог portage. Поддержку клиента NFS можно добавить по желанию.

Сохраните конфиг вашего ядра и пересоберите его. Code: Пересборка ядра

```
gentoo # mount /boot //если это еще не сделано
```

```
gentoo # cd /usr/src/linux
```

Для ядер 2.4:

```
gentoo # make dep && make bzImage modules modules_install install
```

Для ядер 2.6:

```
gentoo # make && make modules_install
```

Включение поддержки

Перезагрузитесь, если включили поддержку NFS в ядро, или, если вы собрали ее как модули, просто запустите: `modprobe nfs`

Установка и настройка

Установите пакет `nfs-utils`: `emerge nfs-utils` Настраиваем автоматическое монтирование общего portage `rc-update add portmap default`

`rc-update add nfs default` Отредактируйте `crontab` на сервере (метод зависит от того, какой cron вы используете). Для `vixie-cron` и `dcron` команда выглядит так: `crontab -e` Добавьте код:

```
0 0 * * * emerge --sync > /dev/null 2>&1 || true --nospinner && emerge world -vup
```

Отредактируйте файл `/etc/exports` на сервере, чтобы в нем были следующие строчки:

```
/usr/portage ip_range/subnet(sync,no_root_squash,rw)
```

Отредактируйте /etc/fstab на клиентах, добавив в него следующий код:

```
SERVER_IP:/usr/portage /usr/portage nfs bg,hard 0 0
```

Отредактируйте /etc/modules.autoload.d/kernel-version, добавив следующую строку:

nfsПримечание: Кеш портежей не расшарен через nfs. Если каждый раз вас докучает обновление кеша при команде emerge сразу после синхронизации, попробуйте добавить следующее в crontab клиентов. Это обновит кеш на компьютерах ваших клиентов спустя 10 минут после старта синхронизации на сервере. Установите время согласно скорости вашего подключения.

```
10 0 * * * emerge --metadata
```

PORTAGE\_TMPDIR

PORTAGE\_TMPDIR должен быть локальным каталогом (для скорости).

Переменная PORTAGE\_TMPDIR устанавливается в файле /etc/make.conf и по умолчанию указывает на /var/tmp, то есть на локальную файловую систему, но просто для того, чтобы убедиться в том, что это так, а не иначе, сделайте: emerge info Если всё прошло успешно (а всё так и должно быть), вы спокойно можете очистить содержимое каталога /usr/portage на КЛИЕНТСКОЙ машине (например, чтобы освободить дополнительное свободное место на диске)

Пояснение кода

Запись crontab'a

```
0 0 * * * emerge --sync > /dev/null 2>&1 || true --nospiner && emerge world -vup
```

говорит о том, что раз в сутки, в полночь будут выполняться синхронизация с перенаправлением вывода в мусор и без бегунка и emerge world -vup и отправка письма root'u о том, что должно быть обновлено - настройте корректно почту).

/etc/exports:

"/usr/portage" - каталог для экспорта

"ip\_range/subnet" - только компьютеры из этого диапазона могут подключать его по сети

"(sync,no\_root\_squash,rw)" - полезные параметры, всегда мною используемые

/etc/fstab:

"SERVER\_IP:/usr/portage" - что вы хотите подключить  
"/usr/portage" - куда вы хотите подключить  
"nfs" - ФС ресурса  
"bg,hard 0 0" - полезные параметры, всегда мною используемые

/etc/modules.autoload.d/kernel-version:

"nfs" сообщает какой модуль загружать при загрузке ядра версии kernel-version

Comments

Советы? Пожелания? Похвала? Дайте нам знать.

Если вы получаете ошибку "Error starting NFS daemon" попробуйте

```
# mount -t nfsd nfsd /proc/fs/nfsd
```

это работает для меня.

## Настройка Portage

Portage Setup Howto

Целевая аудитория

Это руководство создается для людей, которые хотят использовать все возможности системы Portage.

Зачем нужно это howto?

Portage — это система управления пакетами Gentoo Linux. Ее возможности довольно велики — сборка пакетов из исходников, включая конфигурирование пакета на основе пользовательских USE-флагов; начальная настройка приложений; установка предкомпилированных пакетов; удаление програм; автоматическое обновление, и т. д. Однако, как показывает опыт, несмотря на отличную документацию, большинство пользователей не знают всех возможностей этой замечательной системы, поэтому очень часто Gentoo Linux подвергается незаслуженной критике. Данный документ — это попытка восполнить этот пробел.

Профили Portage (Portage profiles)

Gentoo — многоплатформенная система, при этом количество поддерживаемых архитектур постоянно увеличивается. Поэтому возникает необходимость в так называемых профилях сборки. Профиль — это набор настроек по умолчанию для определенной архитектуры, или определенного класса задач (например, для систем с повышенной безопасностью существуют профили `hardened` и `selinux`). Наличие профилей, которые легко создавать, позволяет называть Gentoo не дистрибутивом, а метадистрибутивом, то есть дистрибутивом для создания других дистрибутивов ;). Но не об этом сейчас разговор.

Итак, для того, чтобы иметь работающую Gentoo-систему, необходимо выбрать себе нужный профиль. Профили по умолчанию находятся в каталоге `/usr/portage/profiles`. По названию профиля легко определить тот, который нужен вам. Например, `'default-x86-2004.2'` — это стандартный профиль для архитектуры x86 версии 2004.2. Если в каталоге с профилем есть файл `deprecated`, то это означает, что данный профиль больше не поддерживается и выбирать его не следует.

После того как вы определились с выбранным профилем необходимо создать симлинк `/etc/make.profile` на каталог с выбранным профилем. Например, `ln -s /usr/portage/profiles/default-x86-2004.2 /etc/make.profile` Когда выйдет новая версия Gentoo, вам не надо бежать в магазин, покупать новые диски. Достаточно просто поменять симлинк `/etc/make.profile` на симлинк, указывающий на каталог с новым профилем. Примечание: Сейчас идет переход на новый вид профилей — Cascading

Profiles. Такие профили имеют вид:

`default-linux/x86/2004.2/`

То есть настройки, общие для нескольких профилей не надо записывать по нескольку раз. Пока что это экспериментальная фишка, но если вы хотите помочь ее протестировать, то используйте на здоровье ;).

Примечание: Вообще-то переход на `cascading profiles` уже произошел, посему надо обновить то, что написано выше в этой секции "Portage profiles"

## Настройки профилей

В профилях содержится следующая информация:

use-флаги использующиеся по умолчанию

системные пакеты, то есть такие пакеты, которые устанавливаются при

`emerge system`

Маскированные пакеты.

Настройки сборки пакетов по умолчанию. (`CFLAGS`, `CHOST` и другие)

Какие пакеты надо устанавливать при наличии альтернативных

(например, xorg или xfree)

Конфигурацию профилей напрямую менять не следует, так как ваши настройки будут удалены после следующей синхронизации с rsync-сервером. О том как правильно настраивать portage будет рассказано в следующем разделе.

/etc/portage и /etc/make.conf

Для того, чтобы изменить конфигурацию выбранного профиля, нужно использовать конфигурационные файлы в /etc/portage. Если у вас нет этого каталога, то его необходимо создать. Вот список конфигурационных файлов, которые читает Portage во время своей работы:

/etc/portage/package.mask

Синтаксис: в каждой строке содержится DEPEND ATOM, то есть один из знаков >, <, >=, <=, =, категорию, название и версию пакета. Логические знаки обязательны если указана версия пакета.

Например: >=net-www/mozilla-1.7

Неправильные примеры:

net-www/mozilla-1.7 (нет логической операции)

>=mozilla-1.7 (нет категории)

Предназначение: Маскирует пакеты. Может использоваться для того, чтоб Portage не обновлял нужный вам пакет. Например, если вы хотите использовать apache-1.3 вместо apache-2.0

/etc/portage/package.unmask

Синтаксис: такой же, как и у package.mask Предназначение: Демаскирует пакеты, которые маскированы в профиле. Используется для установки нестабильных пакетов. Рекомендуются использовать только тем, кто хочет протестировать какой-то нестабильный пакет.

/etc/portage/profile/package.provided

Синтаксис: В каждой строке содержится полное имя пакета, то есть category/name-version Например: dev-lang/ghc-6.2.1-r1 Предназначение: Замена 'emerge --inject'. При наличии пакета в package.provided Portage будет считать, что этот пакет уже установлен в системе. Примечание: Иногда этого бывает недостаточно и пакет приходится вносить в

/etc/make.profile/package.provided (который удаляется после каждого emerge sync)

/etc/portage/mirrors

Синтаксис: Такой же как и у файла /usr/portage/profiles/thirdpartymirrors. Предназначение: Список зеркал, которые будут использоваться в первую очередь. Можно также указать зеркала sourceforge или gnu, которые будут использоваться в первую очередь. Например:

sourceforge <http://keihanna.dl.sourceforge.net/sourceforge>

Так же можно использовать тип зеркала 'local'. Такое зеркало будет проверяться, даже если в ebuild'е пакета есть RESTRICT="NOMIRROR". Например:

local <ftp://gentoo.linux.kiev.ua/pub/Linux/Gentoo/distfiles/>

/etc/portage/package.use

Синтаксис: DEPEND ATOM USE-флаги Например: x11-libs/gtk+ doc

Предназначение: Установка индивидуальных USE-флагов для отдельных пакетов

/etc/portage/package.keywords

Синтаксис: DEPEND ATOM KEYWORD KEYWORD — это идентификатор, который показывает, насколько стабилен пакет на данной архитектуре. Всего есть 4 вида KEYWORDS:

arch — пакет стабилен на архитектуре arch

~arch — пакет стабилен на архитектуре arch, но еще не прошло время тестирования

-arch — у пакета наблюдаются проблемы на архитектуре arch.

-\* — пакет нестабилен на всех архитектурах

Например: >=app-editors/emacs-cvs-21.3 ~x86

Предназначение: Установка индивидуальных KEYWORDS для отдельных пакетов. Используйте если хотите, чтоб в вашей стабильной системе были отдельные нестабильные пакеты или наоборот :)

`/etc/portage/categories`

Синтаксис: В каждой строке содержится название категории. Предназначение: Используется для создание новых категорий для Portage. Например:

`app-vasia`

Теперь вы можете создать ebuild для пакетов категории `app-vasia`. `emacs $ {PORTDIR_OVERLAY}/app-vasia/pupkin/pupkin-0.0.1.ebuild`

`/etc/make.conf`

Файл `/etc/make.conf` служит для настройки самого процесса сборки пакетов. Никаких сложных опций там нет, читайте внимательно комментарии в `/etc/make.conf.example`.

`$PORTDIR_OVERLAY`

Представьте себе такую ситуацию: вы хотите установить пакет, ebuild'a которого нет в официальном дереве, но кто-то уже его написал. Или вы собираетесь самостоятельно научиться писать ebuild'ы, и вы хотите, чтоб Portage мог устанавливать нужные вам программы. Если вы запишете свой ebuild в `/usr/portage`, то он сотрется при следующей синхронизации с rsync-сервером. Для того, чтобы third-party ebuild'ы сохранялись, необходимо создать каталог для их хранения (например, `/usr/local/portage`) и записать его имя в переменную `$PORTDIR_OVERLAY` в файле `/etc/make.conf`. Теперь вы можете размещать свои ebuild'ы в этом каталоге, пример был показан в предыдущем разделе.

## Portage Overlay

Введение

Это руководство расскажет о том как установить ebuild который не включен в официальное дерево портежей.

Установка Portage Overlay

Первым делом мы должны указать `PORTAGE_OVERLAY` директорию, для этого необходимо отредактировать файл `/etc/make.conf` Файл: `/etc/make.conf`



`PORTDIR_OVERLAY="/usr/local/portage"`

Таким образом мы позволяем устанавливать дополнительные ebuild'ы при этом не нарушая процесс синхронизации основного дерева портежей.

### Месторасположение ebuild

Проверьте наличие директории `/usr/local/portage`, и если ее нет создайте ее `install -d /usr/local/portage`

Когда помещаете новый ebuild в `/usr/local/portage`, вы должны использовать такую же схему как в `/usr/portage` (`category/program/program.ebuild`)

### Создание digest-файлов

digest-файлы -- это файлы, которые содержат md5 суммы файлов, необходимых для установки пакетов. После того, как Portage скачает необходимые файлы, будет сделана проверка на соответствие файлов их md5-суммам (это дает возможность убедиться в целостности файлов). Для того, чтобы Portage мог устанавливать пакеты, необходим digest-файл. Digest-файл можно создать с помощью команды `ebuild /usr/local/portage/category/program/program.ebuild digest` Вручную редактировать созданные этой командой файлы не следует.

При этом надо учитывать требования к имени ebuild-а. При переименовании ebuild-а в произвольное имя, команда

`ebuild /usr/local/portage/category/program/program.ebuild digest` выдаст ошибку:

`!!! /usr/local/portage does not seem to have a valid PORTDIR structure.`

Чтобы этого не произошло необходимо переименовывать ebuild в то же самое имя, но с приставкой номера релиза.

Пример:

В портежах имеется ebuild:

`/usr/portage/media-sound/ncmipc/ncmipc-0.11.1-r1.ebuild`

И архив исходных кодов:

`/usr/portage/distfiles/ncmpe-0.11.1.tar.gz` для этого `ebuild-a`.

Вы хотите наложить свой патч и добавить `ebuild` вашего пропатченного пакета в систему. Для этого вы должны скопировать исходный `ebuild` в ваш новый, но добавить к нему приставку `'-rN'`, где `N` - номер релиза. Если такая приставка уже есть - изменить её номер:

```
cp /usr/portage/media-sound/ncmpe/ncmpe-0.11.1-r1.ebuild /usr/local/portage/media-sound/ncmpe/ncmpe-0.11.1-r2.ebuild
```

После этого пропатчите исходные коды пакета, на который ссылается исходный `ebuild` `/usr/portage/distfiles/ncmpe-0.11.1.tar.gz` и сохраните его в `/usr/portage/distfiles/ncmpe-0.11.1-r2.tar.gz`

Далее, нам нужно будет поправить наш новый `ebuild`:

```
vi /usr/local/portage/media-sound/ncmpe/ncmpe-0.11.1-r2.ebuild И изменить строку:
SRC_URI="http://mercury.chem.pitt.edu/~shank/{P}.tar.gz mirror://sourceforge/musicpd/{P}.tar.gz"
```

На строку: `SRC_URI="http://mercury.chem.pitt.edu/~shank/{P}-r2.tar.gz mirror://sourceforge/musicpd/{P}-r2.tar.gz"`

Т.е. добавив наш префикс `'-r2'`, чтобы система знала, какой именно файл с исходными кодами ей необходимо прописать. В данном случае скачать с сайта этот пакет конечно же не удастся, но нам этого и не требуется - нам необходимо установить нами поправленный пакет.

После этого даём команду `ebuild /usr/local/portage/media-sound/ncmpe/ncmpe-0.11.1-r2.ebuild digest`, после чего `emerge` будет знать о вашем пакете.

Установка нового `ebuild`

Когда вы будете устанавливать пакет используя новый `ebuild`, скорее всего он будет `masked`, поэтому надо добавить имя пакета в `/etc/portage/package.keywords`

Создайте директорию `/etc/portage/`, если она еще не создана, и затем выполните команду:

```
echo "<category>/<package> ~x86" >>/etc/portage/package.keywords
```

Теперь этот пакет можно установить как и любой другой: `emerge -p package`

`emerge package`

Дополнительно

Unofficial ebuilds содержит список с коротким описанием сайтов где можно найти неофициальные пакеты

Многие, написанные юзерами ebuild'ы, которые еще не попали в официальное дерево, находятся в Bugzill'e Gentoo. Найти нужные ebuild'ы можно с помощью поиска

## Экономия трафика с помощью deltup

Что это такое?

Как известно, программы в Gentoo Linux устанавливаются из исходников. Это чистый текст. Однако, такие программы как, например, gcc или openoffice весят непростительно много. А у большинства из нас интернет не позволяет качать файлы больших объемов, а пользователи модемов вообще толком ничего обновить не могут. Для людей, которые экономят свои деньги, и была написана программа deltup. Вместо `wget` мы заставляем `emerge` использовать `getdelta`. Эта программа скачивает со специального сайта не весь архив программы, а только разницу между требуемой версией и тем, что у вас есть. Если у вас нет предыдущей версии программы, то придется качать весь архив.

Установка и настройка

Установка deltup

Теперь установим deltup: `emerge deltup` и `getdelta`: `emerge getdelta`

note. Если при выполнении `emerge deltup` выйдет ошибка `"!!!All ebuilds that could satisfy "deltup" have been masked."` перед командой добавьте `ACCEPT_KEYWORDS="~x86"`

Настройка emerge

Теперь нужно предупредить `emerge` о том что мы хотим использовать `getdelta` вместо `wget`. Для этого добавим в `/etc/make.conf` параметр `FETCHCOMMAND`: Файл: `File /etc/make.conf`

..

FETCHCOMMAND="/usr/bin/getdelta.sh \\${URI}"

..

## Конфигурация getdelta

У getdelta есть свой файл конфигурации: /etc/deltup/getdelta.rc. Рассмотрим некоторые полезные параметры:

DELTUP\_SERVER - указывает на используемый deltup сервер. Рекомендуется оставить без изменений, так как deltup-сервера сейчас собраны в сеть обращение к которой идёт через один сервер linux01.gwdg.de.

QUEUERETRY - количество секунд, которые будет ждать getdelta, пока сервер не сделает dtu-файл.

MAXIMUM\_ACCEPTABLE\_QUEUEPOS - максимальный номер в очереди ожидания. В связи с ростом загруженности deltup-сервера имеет смысл установить число побольше.

REMOVE\_OLD - удалять старые версии файлов. Полезно, если вы не хотите складировать дистфайлы.

DO\_NOT\_REMOVE - путь к файлу, содержащему имена файлов, которые не следует удалять при включенной опции REMOVE\_OLD

## Дополнительно

Могут возникнуть проблемы если вы скачали часть файла нужного для установки. Дело в том, что для продолжения докачки emerge использует не FETCHCOMMAND, а RESUMECOMMAND. Однако не следует менять RESUMECOMMAND на getdelta, так как getdelta не поддерживает докачку.

## Обновление портежей без доступа в интернет непосредственно из системы

Данное руководство может помочь вам если с доступом в интернет из системы Gentoo проблемы. Однако у вас есть свежий portage-xxxxxxx.tar.bz2. Если portage-xxxxxxx.tar.bz2.md5sum не старше 40 дней(возможно это ограничение можно убрать, но автор этих строк не знает как) тогда проще сделать так :

Создаем папку /var/tmp/emerge-websync/ командой:

```
# mkdir /var/tmp/emerge-websync/
```

Копируем файлы portage-xxxxxxx.tar.bz2 и portage-xxxxxxx.tar.bz2.md5, которые

находятся к примеру, в корневом каталоге на CD-ROM диске (/mnt/cdrom/portage-xxxxxxx.tar.bz2 и /mnt/cdrom/portage-xxxxxxx.tar.bz2.md5sum), командами:

```
# cp /mnt/cdrom/portage-xxxxxxx.tar.bz2 /var/tmp/merge-webrsync/ #
cp /mnt/cdrom/portage-xxxxxxx.tar.bz2.md5sum /var/tmp/merge-webrsync/ и
обновляем портежи при помощи emerge -webrsync: # emerge-webrsync
```

Теперь мы получили систему портежей по состоянию на xxxxxxxx.

Если же portage-xxxxxxx.tar.bz2 старше 40 дней тогда:

Первым делом переименовываем каталог /usr/portage например в /usr/portage2:

```
# mv /usr/portage /usr/portage2
```

затем создаем переименованный /usr/portage:

```
# mkdir /usr/portage
```

Перемещаем из /usr/portage2/distfiles исходники (если вам они нужны :) ) обратно в /usr/portage/distfiles:

```
# cp /usr/portage2/distfiles /usr/portage/distfiles
```

Теперь можно распаковать наш новый portage-xxxxxxx.tar.bz2, который находится, к примеру, в корневом каталоге на CD-ROM диске (/mnt/cdrom/portage-xxxxxxx.tar.bz2), командой:

```
# tar -xvpjf /mnt/cdrom/portage-xxxxxxx.tar.bz2 -C /usr/portage
```

После достаточно продолжительной распаковки нужно обновить кэш системы портежей:

```
# emerge metadata
```

Теперь мы получили систему портежей по состоянию на xxxxxxxx, а в каталоге /usr/portage2 - её бэкап.

Дополнительно можно прочитать HOWTO Обновление пакетов без доступа в интернет непосредственно из системы.

## **Обновление пакетов без доступа в интернет непосредственно из системы**

В моем случае это понадобилось в следующей (я полагаю довольно распространенной) ситуации: дома - нет интернета (дорогой, медленный - нужное подчеркнуть :) ), а на работе - хороший и бесплатный. Итак для начала необходимо получить список нужных пакетов. В этом примере мы хотим установить/обновить

glibc baselayout texinfo gettext zlib binutils gcc ncurses. Список мы сохраняем на устаревший магнитный носитель, но счастливые обладатели более надежных носителей естественно могут пользоваться ими.

Получение списка для скачивания:

(Не забываете '2' перед '>') Code:

```
# emerge -fp glibc baselayout gettext zlib binutils gcc ncurses 2> stage1.list
# mount -t vfat /dev/fd0 /mnt/floppy
# cp /mnt/gentoo/stage1.list /mnt/floppy
# umount /mnt/floppy
```

Теперь вставляем дискету в компьютер с хорошим каналом. (В моем случае это рабочий компьютер.) Если заглянуть в файл stage1.list, можно увидеть, что в нем перечисляются несколько ссылок на каждую загрузку. К сожалению, это не совсем то, что нам надо. Сначала список надо почистить от лишнего:

Очистка от лишних ссылок:

Этот скрипт привязан к формату вывода emerge, который может измениться без предварительного предупреждения - используйте с осторожностью! # cut -f 1 -d ' ' stage1.list > stage1.download

Теперь используем wget для загрузки списка пакетов:

```
# wget -N -i stage1.download
```

Получив все файлы, переносим их в наш компьютер в /mnt/gentoo/usr/portage/distfiles. И для проверки запустим

```
# emerge -pv glibc baselayout gettext zlib binutils gcc ncurses
```

Если в строке Total Download: указана цифра 0 - значит все было сделано верно. Можно теперь запускать установку:

```
# emerge glibc baselayout gettext zlib binutils gcc ncurses
```

У всего вышеописанного есть один недостаток. В файле stage1.download есть все

необходимые для установки файлы. Даже те, которые у вас могут быть в distfiles. А как вырезать ссылки из stage1.download на имеющиеся в distfiles файлы я не знаю. Если кто знает - напишите.

Взято с Gentoo.org: <http://www.gentoo.org/doc/ru/altinstall.xml>

Дополнительно можно прочитать HOWTO Обновление портежей без доступа в интернет непосредственно из системы

## Полное обновление системы

Приведение в порядок /var/lib/portage/world

В world должен быть список программ, которые нужно доустановить к тем, которые уже входят в "system" (т.е. в текущий профайл).

в world не должно быть никаких библиотек, и т.д., которые не нужны сами по себе, а нужны только для удовлетворения чьих-то зависимостей (чтобы не продолжать устанавливать/обновлять их, если они уже станут не нужны по какой-то причине)

программ, которые уже входят в "system", не должно быть в world

в world нельзя указывать определенную версию софта, это лучше делать в /etc/portage/package.mask

скрипт regenworld может помочь восстановить world путем анализа /var/log/emerge.log и генерации на его базе файла world (он перезапишет текущий world!)

скрипт dep -p -w поможет найти избыточные записи в world(которые всё-равно нужны другим записям в world или входят в system).

Перед серьёзными обновлениями желательно просмотреть /etc/portage/\*, т.к. там могут быть уже не актуальные записи мешающие текущему обновлению.

## Обновление profile

Не каждый Gentoo release включает в себя новый profile (например, 2004.1 был без profile).

Даже если новый profile есть, то переходить на него не обязательно (если это будет обязательно, то старый профайл будет deprecated и emerge об этом должен будет громко кричать).

Инструкции по обновлению profile будут выкладываться здесь: <http://www.gentoo.org/doc/en/gentoo-upgrading.xml> и как правило сводиться к изменению симлинка /etc/make.profile

## USE-флаги

Запустить emerge -uDpv --newuse world и проверить что USE-флаги для всех пакетов выставлены корректно, и при необходимости скорректировать

USE-флаги выставляются в /etc/make.conf и /etc/portage/package.use

Запуск обновления системы (если не нужно обновлять toolchain)

Если

```
emerge -uDav --newuse world
```

показывает что будет обновляться пакет входящий в toolchain (linux-headers, glibc, binutils или gcc), то крайне рекомендуется полностью перекомпилировать всю систему - см. следующий пункт - а иначе можно вместо следующего пункта просто запустить:

```
emerge -uDav --newuse world
```

Некоторые причины не использовать emerge -U вместо -u

Причина 1: Проблемы со SLOT

Это, к примеру, происходит потому, что некоторые люди хотели gimp-2 вместо gimp-1.2. Представьте ситуацию, где gimp-1.2 помечен stable и находится в SLOT 1, gimp-2 помечен unstable и находится в SLOT 2. Теперь при выполнении ACCEPT\_KEYWORDS=~x86 emerge gimp получите gimp-2.

Позже, когда вы посчитаете, что наступило время обновить свою систему чем-либо похожим на "emerge -U world", эта команда установит gimp-1.2, потому, что gimp находится в world-файле, и флаг "-U" не обрабатывает SLOT должным образом.

Причина 2: Проблемы, в случае удаления ebuild-ов с Portage-дерева.

Допустим, в Portage находятся 2 версии пакетов foo, foo-1.4 (помеченный как stable) и foo-1.6 (помеченный как unstable). Вы хотите вариант unstable и делаете emerge, как с вышеуказанным gimp. Позже обновляете world как было сказано выше, но в промежутке этого времени вышло критическое обновление для foo-1.6 - foo-1.6.1. Теперь появляется несколько возможностей обработки.

foo-1.6 был удален из Portage. Будет установлен foo-1.4, несмотря на "снижение" версии вместо флага "-U"

Ситуация будет еще хуже, если foo-1.6 не был удалён из Portage по какой-либо причине: foo-1.6 (тот, что с критической уязвимостью) будет оставаться на вашей системе до тех пор, пока не будет помечено stable что-либо выше чем foo-1.6.

Обновление одного из пакетов входящих в toolchain

Если обновляется хотя-бы один из linux-headers, glibc, binutils или gcc, то рекомендуется пересобрать их дважды, после чего весь system, после чего весь world. Примечание: Цель двойной компиляции toolchain - получить гарантированно стабильный и корректный toolchain не зависящий от предыдущего.



Перекомпилировать system/world после этого жёсткой необходимости нет, по крайней мере если остальной софт продолжает работать (возможно даже используя библиотеки из старого toolchain - см. предыдущие пункты об апгрейде).

Цель перекомпиляции system/world - чтобы весь софт получил потенциальное преимущество от установки нового toolchain. system перекомпилируется перед world из тех-же соображений, т.к. при компиляции программ из world используются утилиты из system.

Если увеличивается первая или вторая цифра версии gcc, то перед второй сборкой нужно переключиться на новую версию через gcc-config - иначе новый gcc просто установится параллельно со старым в "новый слот", но по умолчанию использоваться будет старый.

При сборке system после двойной перекомпиляции toolchain нет необходимости опять компилировать toolchain как часть system. Аналогично при сборке world после system нет необходимости опять компилировать пакеты из system как часть world. Это можно попробовать обойти либо вручную, либо используя скрипты [1], либо через бинарные пакеты и `emerge -k` (я предпочитаю последний вариант).

Итак, рекомендованный набор команд:

```
# для того, чтобы безопасно использовать `emerge -k` нужно очистить
```

```
# каталог с текущими бинарными пакетами
```

```
# (напр., переместить его в /tmp/portage-packages)
```

```
pkgdir=$(portageq pkgdir)
```

```
mv $pkgdir /tmp/portage-packages1
```

```
install -d -o portage -g portage $pkgdir
```

```
# первая сборка toolchain
```

```
emerge linux-headers glibc binutils gcc-config gcc
```

```
# выбрать новый gcc если он установился в новый слот
```

```
gcc-config имя_или_номер_нового_gcc
```

```
# см. `gcc-config -l`
```

```
source /etc/profile
```

```
# компиляция toolchain с созданием бинарных пакетов
```

```
emerge -b glibc binutils gcc portage
```

```
# не компилировать glibc, binutils и gcc
```

```
emerge -bke system
```

```
# не компилировать предыдущие пакеты (включая system)
```

```
emerge -bke world
```

Примечание: Чисто теоретически существует пакет binutils-config, который когда-нибудь может потребоваться использовать аналогично gcc-config.

Обновление безопасности. Примечание: Даже после `emerge -uDav --newuse world` в системе могут оставаться

устаревшие пакеты с дырами в безопасности - в слотах!

```
glsa-check -l | grep '[N\]'  
emerge ... # если нужно
```

Удаление неиспользуемых пакетов.

После обновления системы в ней могут оказаться пакеты, которые никто не использует. Эти пакеты желательно удалить, т.к. они не будут в дальнейшем обновляться при `emerge -uDav --newuse world`.

```
emerge -a depclean # очень осторожно!!!
```

После обновления библиотек может потребоваться перекомпилировать программы, которые эти библиотеки используют:

Примечание: Для glsa-check, revdep-rebuild необходимо установить пакет gentoolkit

```
rm /root/.revdep-rebuild*.*_*  
revdep-rebuild -p  
revdep-rebuild
```

Обновление конфигов.

```
dispatch-conf
```

Если используется runit-init и обновлялся пакет baselayout, то нужно восстановить /sbin/init:

```
ls -l /sbin/*init*  
if (/sbin/init это бинарник, а не симлинк) {  
mv /sbin/init /sbin/init-sysv  
ln -s runit-init /sbin/init  
}
```

Отслеживание важных сообщений при установке пакетов.

В процессе emerge world выдаётся очень много сообщений, причём важные комментарии перемешаны с командами компиляции, и отследить их при сборке нескольких пакетов одновременно не возможно.

Но все эти сообщения можно получить из log-файлов после окончания установки emerge world. Для этого нужно использовать либо enotice, либо portlog-info.

## Сборка на другом компьютере

Существует много методов добиться этого, но мы используем самый удобный - тот, что работает с портежами.

Другие методы могут быть более подходящими для других ситуаций, когда главная проблема не только медленный процессор и долгое время сборки, например, если надо перенести Gentoo на систему, которая не может загружаться с CD или сети.

Скомпилировать Gentoo для старых систем x86 (например, Pentium Pro 200MHz с небольшим количеством RAM) можно на x86\_64-системе с работающей Gentoo. Обратный трюк, собрать 64-битную систему на 32-битном компьютере, невозможен.

Я собирал систему в chroot-окружении согласно Gentoo-handbook, используя мою 64bit-Gentoo вместо "живого" CD:

```
# mkdir /your/new/gentoo
# tar xjvpf stage3-x86-*.tar.bz2 -C /your/new/gentoo
```

Отредактируйте переменные CHOST и CFLAGS, соответствующие вашей целевой системе (не изменяйте CHOST, если используете стадию 2 или 3 - возьмите стадию для соответствующей платформы). У меня было:

```
# CHOST="i686-pc-linux-gnu"
# CFLAGS="-march=pentiumpro -Os -momit-frame-pointer -pipe"
# CXXFLAGS="$CFLAGS"
```

Далее переходим в chroot-окружение и выполняем все шаги, описанные в Руководстве Gentoo по установке.

Чтобы собрать ядро для i386 на x86\_64-машине добавьте ARCH=i386 при конфигурации и компиляции ядра.

```
# make menuconfig ARCH=i386
# make clean dep modules modules_install bzImage ARCH=i386
# cp arch/i386/boot/bzImage /boot/vmlinuz
```

Размечать диск и создавать файловые системы на старом компьютере вам придется с помощью "живого" CD или другой системы Linux.

Переносим готовую систему на старый компьютер:

```
# cd /your/new/gentoo
# echo "/sys/*" >> tar_exclusions
# echo "/var/run/*" >> tar_exclusions
```

```
# echo "/tmp/*" >> tar_exclusions
# echo "/proc/*" >> tar_exclusions
#
# tar -C /your/new/gentoo -X tar_exclusions --preserve -cf ../gentoo.tar .
## можете добавить -v для подробного вывода или -j / -z для компрессии
# scp [-P ssh_port] ../gentoo.tar user@dest_pc:
```

На старой машине подготавливаем диски и распаковываем систему:

```
# tar -C /your/new/system --preserve -xf ~user/gentoo.tar
```

Теперь вы можете войти в chroot-окружение. Сконфигурируйте системный загрузчик и перезагрузитесь. Теперь вы находитесь в новой, с иголки, Gentoo.

Система теперь должна работать, но я заметил проблему - я не могу ничего скомпилировать. Похоже, что не работает ни C-препроцессор, ни gcc (обычно configure говорит, что /lib/cpp fails sanity check). Не знаю почему, но копирование /usr/include из chroot-каталога с исходной машины на старый компьютер эту проблему решило.

Удачи!

Оригинал на [http://gentoo-wiki.com/HOWTO\\_Compile\\_on\\_another\\_computer](http://gentoo-wiki.com/HOWTO_Compile_on_another_computer)

Перевод Poor Fred

## **Прочее**

### **Разделы MS Windows по-русски**

Введение

Достаточно часто возникает необходимость подмонтировать раздел с установленным M\$ Windows.

Ядро

Для того, чтобы система могла работать с разделами FAT или NTFS, необходимо включить их поддержку в ядре. Linux Kernel Configuration: Filesystems

File systems ->

DOS/FAT/NT Filesystems ->

- (M) DOS FAT fs support
- (M) MSDOS fs support
- (M) VFAT (Windows-95) fs support
- (M) NTFS file system support

Также надо включить поддержку следующих кодировок: Linux Kernel Configuration: Native Language Support

File systems ->

Native Language Support ->

<M> Windows CP1251 (Bulgarian, Belarusian)

<M> Codepage 866 (Cyrillic/Russian)

## Монтирование

Теперь можно примонтировать любой windows-раздел программой mount. `mount -t file_system /dev/device /mnt/dir` В нашем случае в качестве file\_system необходимо указывать vfat или ntfs, в зависимости от файловой системы раздела с windows.

## Монтирование FAT

Для монтирования разделов FAT нужно использовать следующую команду: `mount -t vfat -o codepage=866,icharset=koi8-r,quiet,umask=000 /dev/hdd2 /mnt/win1` Для украинских символов: `mount -t vfat -o codepage=866,icharset=koi8-u,quiet,umask=000 /dev/hdd2 /mnt/win1` Мы использовали дополнительные опции icharset, codepage, quiet и umask. Рассмотрим их внимательнее:

codepage - это кодовая страница используемая на монтируемой ФС. Так как мы из России, то нужно писать codepage=866. Эта опция нужна для перекодировки имен файлов.

icharset - указывает какую кодировку использовать для ввода/вывода. Нужно указать свою системную локаль. Локаль можно узнать командой locale. (Здесь под вводом/выводом понимается ввод с и вывод на терминал. Именно поэтому нужно указывать кодировку локали, т.е. кодировку которую используют программы запущенные в терминале.)

quiet - при копировании на файловую систему FAT программа попытается записать права доступа, владельца и т.д., но на FAT их сохранить нельзя и будет выведено сообщение об ошибке ( хотя файл скопируется ). Чтобы ошибки не появлялись мы просим смонтировать в "молчаливом" режиме

umask - в некоторых случаях у простого пользователя может не хватить прав на чтение или запись файлов из-за неправильно установленных прав на файлы и директории. Поскольку у файловой системы FAT нет прав доступа, они выставляются операционной системой ( в данном случае Linux ) на лету в соответствии с указанным значением umask. Про umask смотрите в umask(2)

Параметры `codepage` и `iocharset` можно задать прямо в ядре. Тогда не нужно будет писать их каждый раз: Linux Kernel Configuration: FAT

File systems ->

DOS/FAT/NT Filesystems ->

(utf8) Default iocharset for FAT

(866) Default codepage for FAT

То есть вместо `mount -t vfat -o codepage=866,iocharset=utf8,quiet,umask=000 /dev/hdd2 /mnt/win1` можно писать `mount -t vfat -o quiet,umask=000 /dev/hdd2 /mnt/win1`

## Монтирование NTFS

Для монтирования NTFS разделов опции `mount` немного изменяться. Делаем это так: `mount -t ntfs -o nls=koi8-r,umask=0,ro /dev/hdd2 /mnt/win1` Для украинских символов: `mount -t ntfs -o nls=koi8-u,umask=0,ro /dev/hdd2 /mnt/win1` Вот что они значат:

`nls` - указывает на вашу системную локаль. Отсутствие опций `codepage` и `iocharset` связано с тем, что NTFS разделы используют юникод для хранения имен файла.

`umask` - указывает на то что читать файлы с этого раздела могут все. А вот писать не разрешает опция `ro`. Мы рекомендуем ее поставить, так как функция записи на разделы NTFS еще не отлажена.

В случае если вы в качестве системной кодировки используете `utf-8`, то опции `mount` должны быть следующими: `mount -t ntfs -o utf8,umask=0,ro /dev/hdd2 /mnt/win1`

`utf8` - использовать UTF-8 для перекодировки имён файлов.

## Автомонтирование

Если у вас все еще есть разделы ОС Windows на харде, то можно добавить монтирование этих разделов при загрузке. Для этого необходимо отредактировать файл `/etc/fstab`. Файл: `/etc/fstab`

...

```
/dev/hda9 /mnt/win1 vfat auto,codepage=866,iocharset=koi8-r,quiet,umask=000 0 0
```

```
#/dev/hda9 /mnt/win1 vfat auto,codepage=866,iocharset=koi8-u,quiet,umask=000 0 0
```

```
/dev/hda3 /mnt/win2 ntfs auto,nls=koi8-r,umask=0,user 0 0
```

еще к опциям можно добавить `showexes`, чтобы все файлы не выглядели

исполняемыми, в случае с FAT32

## Udev

Настройка системы

Установка необходимых программ

```
emerge udev hotplug coldplug
```

Конфигурация ядра

Для активизации udev нам необходимо ядро 2.6. Лично у меня 2.6.12-mm1.

General setup --->

☒ Support for hot-pluggable devices

File systems --->

Pseudo filesystems --->

☐ /dev file system support (OBSOLETE)

☐ Automatically mount at boot (NEW)

☒ Virtual memory file system support (former shm fs)Примечание: devfs Может быть включен, но автоматическое монтирование обязано быть выключено. А с учетом того, что из 2.6.13 его вообще собираются убрать... ну смотрите сами.

Примечание: Во всяком случае, из 2.6.14 убрали

Ядро собираем и устанавливаем.

Конфигурация системы

Не буду рассматривать как udev работает из таррбола, рассматриваю вариант чистого udev.

/etc/conf.d/rc:

...

```
RC_DEVICE_TARBALL="no"
```

...

```
RC_DEVFSD_STARTUP="yes"
```

...Примечание: Последнюю опцию можно поставить в "но", когда мы убедимся в нормальной работоспособности udev

```
rc-update add coldplug boot rc-update add hotplug default
```

Конфигурация процесса загрузки

GrUB

К строке kernel от старого ядра необходимо дописать:

```
kernel /boot/kernel-2.6.9-r4 root=/dev/hda3 gentoo=noudev
```

А в строке kernel для нового ядра прописать:

```
kernel /boot/kernel-2.6.10-r6 root=/dev/hda3 gentoo=nodev
```

Troubleshooting

Попытка номер раз

В принципе можно перегружаться, но нас ждет одно неприятное сообщение. Вот примерно такое:

```
WARNING: Unable to open an initial console
```

Для того чтобы все было хорошо, нам необходимо загрузиться с другого носителя (например любой LiveCD) подмонтировать наш корневой раздел, и в каталоге /dev (каталог по идее должен оказаться абсолютно пустым) проделать следующее: Code:

```
# mknod -m 660 console c 5 1
```

```
# mknod -m 660 null c 1 3
```

РОМАН007: Зачем же так жестоко - заставлять грузиться с компакт? Согласно udev guide с [www.gentoo.org](http://www.gentoo.org/doc/en/udev-guide.xml) (<http://www.gentoo.org/doc/en/udev-guide.xml>), доступ к девственному /dev можно получить альтернативно: Code: Code Listing 3.1: Listing device nodes available at boot

```
# mkdir test
```

```
# mount --bind / test
```

```
# cd test/dev
```



# ls

Попытка номер два

С консолью вроде все в порядке, а вот иксы не хотят... грузимся в шелл и в файле конфигурации Xorg прописываем в разделе мыши:

Option "Device" "/dev/input/mice"

PS

Ну теперь вроде все так как было раньше. и старая конфигурация тоже работает. Если старая конфигурация не нужна, то убираем все из загрузчика, ставим в /etc/conf.d/rc - "no" и живем спокойно под udev.

PPS

Кстати тарболл может потребоваться пользователям нестандартных устройств, а так же некоторых сторонних модулей ядра

## **Использование CFLAGS для оптимизации собранных программ**

Для чего все это надо?

Стремление выжать из своего компьютера максимум производительности есть в каждом, ну почти в каждом ;-). Особенно его много в русских линуксоидах Gentoo'шниках ;) Мы попытаемся путем изменения флагов оптимизации ускорить работу приложений нашей системы. ВНИМАНИЕ Некоторые флаги могут сделать приложения нестабильными, так что нужно быть аккуратным.

Оптимизация

Переменная окружения CFLAGS

Для указания параметров оптимизации компилятору GCC, используется переменная окружения CFLAGS. Эта переменная определена в /etc/make.conf, её можно изменить двумя способами:

Отредактировать эту переменную в /etc/make.conf;

Экспортировать ее в окружение (emerge будет использовать эти параметры, но

каждый раз выполнять export неудобно):

export CFLAGS='параметры оптимизации'

## Уровни оптимизации

Для gcc версий 3.x и выше существует только 5 уровней оптимизации: -O0 (без оптимизации), -O1, -O2 и -O3 (O3 - самый высокий уровень), а так же -Os.

Примечание: Если вы используете несколько -O опций, то только последняя объявленная будет оказывать влияние на процесс компиляции.

### -O0

Отключает оптимизацию. Только переменные, объявленные register, сохраняются в регистрах.

### -O(-O1)

Включает оптимизацию. Пытается уменьшить размер кода и ускорить работу программы. Соответственно увеличивается время компиляции. При указании -O активируются следующие флаги: -fthread-jumps, -fdefer-pop.

На машинах, у которых есть слоты задержки, включается опция -fdelayed-branch.

На тех машинах, которые способны поддерживать отладку даже без указателя на стек функции, также включается опция -fomit-frame-pointer.

На других машинах могут быть включены и другие флаги.

### -O2

Оптимизирует еще больше. GCC выполняет почти все поддерживаемые оптимизации, которые не включают уменьшение времени исполнения за счет увеличения длины кода. Компилятор не выполняет раскрутку циклов или подстановку функций, когда вы указываете -O2. По сравнению с -O, эта опция увеличивает как время компиляции, так и эффективность сгенерированного кода.

-O2 включает все флаги оптимизации наследованные от -O. Также включает следующие флаги оптимизации:

- fforce-mem -foptimize-sibling-calls
- fstrength-reduce -fcse-follow-jumps -fcse-skip-blocks
- frerun-cse-after-loop -frerun-loop-opt -fgcse -fgcse-lm
- fgcse-sm -fgcse-las -fdelete-null-pointer-checks -fexpensive-optimizations
- fregmove -fschedule-insns -fschedule-insns2 -fsched-interblock
- fsched-spec -fcaller-saves -fpeephole2 -freorder-blocks
- fre-order-functions -fstrict-aliasing -funit-at-a-time -falign-functions
- falign-jumps -falign-loops -falign-labels -fcrossjumping

-O3

Оптимизирует еще немного. Включает все оптимизации -O2 и также включает флаг -finline-functions и -fweb.

-Os

Включает оптимизацию по размеру. -Os флаг активирует все флаги оптимизации из -O2, в основном те, которые не увеличивают размер выходного файла. В дальнейшем выполняются оптимизации по уменьшению размера кода.

-Os выключает следующие флаги оптимизации: -falign-functions, -falign-jumps, -falign-loop, -falign-labels, -freorder-blocks, -fprefetch-loop-arrays.

Примечание: Более полное описание флагов -Ox, -fflag смотрите в man gcc

## Оптимизация под тип процессора

Не все процессоры одинаковы, поэтому давайте укажем компилятору на наш тип процессора. Для этого есть опции -mtune и -march. Отличие в том, что с опцией -mtune компилятор сделает код, который будет совместим с более младшими моделями процессора, в то время как с -march этого не происходит. Вот список возможных значений для данных опций:

i386

i486

i586

i686

pentium

pentium-mmx

pentiumpro

pentium2

pentium3

pentium4

pentium-m

prescott

nocona

k6

k8

k6-2 (не рекомендуется ставить, из-за багов в компиляторе, заменять на i686)

k6-3

athlon

athlon-tbird

athlon-4

athlon-xp  
athlon-mp  
athlon64  
opteron  
winchip-c6  
winchip2  
c3.

Внимание! pentium-m - это аналог для pentium3. Если процессор в вашем ноутбуке Mobile Intel Pentium 4 - M, то нужно ставить опцию pentium4 или pentium4m (они равнозначны)

Выбор оптимальных параметров

Для этого есть очень интересная утилита. emerge acovea

Правда существующие профили рассчитаны только на pentium 3/4, и на gcc 3.3/3.4, Но в принципе добавить свою конфигурацию тоже не составляет труда. Также рекомендуется добавить в конфигурацию опции -ftracer и -mfpmath=sse. В некоторых случаях они дают значительный прирост производительности сгенерированного кода.

После чего вызываем утилиту runacovea -config gcc33\_pentium3.acovea -bench evobench.c Ждем несколько часов и получаем оптимальные флаги компиляции.

Возможны различные тесты, которые хранятся в каталоге /usr/share/acovea/benchmarks, И различные конфигурации платформы /usr/share/acovea/config, к которым при желании можно добавить свою.

## Редактор nano

nano - Nano's ANOther editor

Nano - достаточно простой и удобный текстовый редактор, клон редактора pico

Установка

emerge nano

КонфигурацияФайл: /etc/nanorc

Основной файл конфигурации

Запуск

Запуск редактора

`nano -w somefile`

**Параметр -w здесь означает запрет переноса строк.**

## Udev и автомонтирование носителей

Введение

В данной статье речь пойдет о том, как настроить udev для автоматического монтирования и отмонтирования usb-носителей. Ведь согласитесь, неудобно каждый раз монтировать флэшки из консоли. А когда вам нужно быстро что-то скинуть с одной флэшки на другую? Здесь я привожу одно из возможных решений.

Требования

У меня установлена следующая версия udev:

`sys-fs/udev-069`

Я умышленно не ставлю более новые версии, т.к. с ними возникают проблемы при загрузке firmware на моем компьютере. Возможно в более новых версиях что-то работает по-другому, но суть одна и та же.

Настраиваем udev

Создаем правила для flash-носителей

Правила udev, в соответствии с которыми устройствам назначаются имена, а также выполняются специфические действия, расположены в `/etc/udev/rules.d/*`. В этой директории уже есть файлы, содержащие правила, у меня их было 2:

`# ls /etc/udev/rules.d`

`05-udev-early.rules`

`50-udev.rules`

Такие названия им были даны неспроста. Дело в том, что udev обрабатывает файлы в этой директории в алфавитном порядке по возрастанию. Вообще, при написании правил udev следует помнить 2 главные вещи:

Udev считывает файлы из /etc/udev/rules.d в алфавитном порядке.

Найдя первое подходящее правило для устройства, udev прекращает дальнейший поиск.

Давайте добавим свой файл с правилами так, чтобы он обрабатывался раньше, чем файл с правилами по умолчанию (50-udev.rules). Назовем его 10-udev-my.rules. Содержимое нового файла будет следующим: Файл: /etc/udev/rules.d/10-udev-my.rules

# First rule

```
SUBSYSTEM=="block", KERNEL=="sd*", ACTION=="add", NAME="%k", GROUP="disk",  
RUN+="/etc/udev/scripts/udev-flash-mount add %k"
```

#Second rule

```
SUBSYSTEM=="block",          KERNEL=="sd*",          ACTION=="remove",          RUN  
+="/etc/udev/scripts/udev-flash-mount remove %k"
```

Теперь разберемся в том, что написали.

Первое правило (добавление устройства)

**SUBSYSTEM=="block"** При добавлении нового устройства в систему возникает множество событий разных классов. block - это класс для блочных устройств, который нам больше всего подходит для составления правила. Для просмотра значений всех переменных, возникающих событий используйте `udevmonitor --env T.O.` здесь мы реагируем только на событие, в окружении которого есть переменная SUBSYSTEM со значением block .

**KERNEL=="sd\*"** Эта часть правила сравнивает имя устройства, которое было дано по умолчанию ядром, с шаблоном. В данном случае нам нужны такие имена как: sda, sda1, sdb, sdb1 и т.д.

**ACTION=="add"** Реагируем на событие добавления устройства. Например, usb флэшка была вставлена в разъем.

**NAME="%k"** Обратите внимание на то, что здесь стоит один знак равенства, а не два, как в предыдущих частях правила. Это говорит о присваивании, а не сравнении. Здесь мы присваиваем вставленному устройству имя, которое соответствует тому, что было дано по умолчанию ядром. %k это одна из переменных окружения. Подробнее о них можно прочитать в `man udev`.

**GROUP="disk"** Причисляем устройство к группе дисковых.

**RUN+="/etc/udev/scripts/udev-flash-mount add %k"** Это самая интересная часть скрипта. Здесь мы добавляем ("+=") одно из действий, которое произойдет после обработки правила. В данном случае мы запускаем скрипт (он будет рассмотрен далее) /etc/udev/scripts/udev-flash-mount и передаем ему 2 параметра. Первый указывает на то что нужно выполнить действия по добавлению и инициализации нового устройства, а второй содержит имя устройства, с которым нужно выполнять

действия.

Второе правило (удаление устройства)

ACTION=="remove" Реагируем на удаление устройства.

RUN+="/etc/udev/scripts/udev-flash-mount remove %k" Здесь мы запускаем скрипт, использовавшийся в первом правиле, но уже для выполнения действий после удаления устройства из системы. Этот скрипт будет выполнен как только ядро заметит, что устройство было удалено из системы. Это не самое лучшее решение, т.к. данные могут быть утеряны, но об этом в следующем разделе.

Пишем скрипт

Теперь напишем скрипт, который будет выполняться после добавления устройства и после его удаления. Допустим он будет размещен в /usr/bin/. Итак, от пользователя root:

```
# su
```

Создадим файл:

```
# touch /etc/udev/scripts/udev-flash-mount
```

Сделаем его исполняемым:

```
# chmod u+x /etc/udev/scripts/udev-flash-mount
```

Напишем в нем следующее: Файл: /etc/udev/scripts/udev-flash-mount

```
#!/bin/bash
```

```
LOG="/var/log/udev"
```

```
sleep 3
```

```
DEV=`echo $2 | sed -n '/^sd[a-z][1-9]\?/p`
```

```
if [ "$1" = "add" ]; then
```

```
    if [ "$DEV" != "" ]; then
```

```
        echo "--- `date` ---" >> $LOG
```

```
        echo "Mounting /dev/$DEV" >> $LOG
```

```
        mkdir /mnt/$DEV >> $LOG 2>&1
```

```
        chmod a+rw /mnt/$DEV >> $LOG 2>&1
```

```
        mount /dev/$DEV /mnt/$DEV -o sync,umask=0000,ioccharset=cp1251 >> $LOG 2>&1
```

```
    fi
```

```
elif [ "$1" = "remove" ]; then
```

```

if [ "$DEV" != "" ]; then
    echo "--- `date` ---" >> $LOG
    echo "Unmounting /dev/$DEV" >> $LOG
    umount /dev/$DEV >> $LOG 2>&1
    rm -rf /mnt/$DEV >> $LOG 2>&1
fi
fi

```

Задержка в 3 или более секунд необходима для того, чтобы ядро успело выполнить свои действия по инициализации устройства. Если не лень, то поэкспериментируйте с задержкой. Если скрипту был передан параметр add, то он создает папку с именем устройства, затем пытается примонтировать устройство в эту папку. Весь вывод, в том числе и ошибки, выводятся в файл лога. Соответственно, при вызове скрипта с параметром remove, он пытается отмонтировать устройство и удалить ранее созданную папку. Предупреждение: Опция sync в параметрах mount является необходимой, т.к. форсирует мгновенную запись на устройство. Если ее не будет, то когда вы удалите устройство из системы, данные не будут записаны на него. Поэтому, чтобы избежать проблем такого рода, дождитесь пока индикатор на носителе перестанет мигать и после этого извлеките устройство. НО для файловой системы fat с версии ядра linux 2.6.13 sync ставить нельзя т.к. функционирование данной опции сильно изменилось и максимальная скорость записи на usb2.0 это 200 килобайт в секунду а для usb1.0 это где то 10 - 20 килобайт в секунду. Кроме того, опция sync в параметрах mount убьет Вашу флешку с гарантией за 6 месяцев использования. См. <http://bugs.debian.org/309625>

Предупреждение: Внимание! Если какая-либо запущенная программа просматривает содержимое папки, в которую смонтировано устройство, то скрипт не сможет отмонтировать его и удалить папку. Поэтому сначала убедитесь в том, что ни одна из программ не использует эту папку.

## Заключение

Последний скрипт можно немного модернизировать. Предположим, если вы монтируете свою камеру, чтобы просмотреть фотографии, то можно передать скрипту еще один параметр из udev правила, например, с производителем устройства. Либо можно проанализировать содержимое каталога и на основе этого выполнить какие-либо действия. Все в ваших руках, дерзайте

## Оптимизация glibc

ведение



Glibc - свободная библиотека, предоставляющая системные вызовы и другие основные операторы для основных систем GNU/Linux. Если C является наиболее распространённым языком, используемым для программирования в Linux, такая же судьба постигла и glibc - она является продуманной частью ядра системы. Glibc может быть оптимизирована на вашей системе для ускорения операций различными способами. Оптимизация, естественно, зависит от вашей системы и её использования. Это также зависит от установки флагов USE, доступных при выполнении emerge. До применения рекомендаций рекомендуется узнать о назначении и использовании флагов USE. Есть неплохая инструкция [здесь](#).

## Оптимизация компиляции Glibc

### Эффект CFLAGS

Компиляция Glibc с возможностями GCC используется в основном коде glibc. Glibc идентифицирует GCC как компилятор и использует установленный флаги CFLAGS. Подробнее [здесь](#). Используя правильные флаги CFLAGS для вашей системы, вы можете хорошо оптимизировать glibc.

### Улучшение оптимизации

Если вы использовали `-fomit-frame-pointer`, вы можете также поместить `glibc-omitfp` во флаги USE. Повторно пересоберите glibc. Code: `Remerging glibc`

```
emerge --newuse -v world
```

При этом соберутся Glibc и её зависимости. При компиляции glibc этот флаг активизирует флаг `-enable-omitfp`, который, в свою очередь, указывает glibc на использование `--fomit-frame-pointer`. В результате более безопасно устанавливается указатель кадра. Этот флаг устанавливает максимальную оптимизацию glibc и предоставляет два типа библиотек - 'оптимизированная' и 'стандартная'. То есть, по умолчанию будут использоваться оптимизированные библиотеки, но при необходимости можно будет использовать 'стандартную' версию. Это увеличит размер glibc и приведет к обычному поведению при установленном `--fomit-frame-pointer`, т.е. сделает невозможным отладку (debug) программ, так что не используйте этот флаг если вы планируете заниматься разработкой программ. Теоретически этот флаг может вызвать некоторые ошибки компилятора, хотя на практике он достаточно безопасен, тем не менее вы были предупреждены.

### Модели потоков

glibc поддерживает 2 различных модели потоков - старую `linuxthreads` и новую `nptl`. По умолчанию, если не был указан флаг `nptlonly`, кроме `nptl` собирается и версия с `linuxthreads`. В случае, если у вас нет устаревшего программного обеспечения, вы можете указать флаг `nptl`. Если у вас современная система без сторонних бинарных пакетов, будет лучше указать `nptlonly`, чтобы избежать сборки glibc дважды (с

поддежкой linuxthreads и без неё).

## Локализация

В основном это основано на еженедельном новостном бюллетене Gentoo от 8 ноября 2004 года, который можно найти здесь (en:TIP Specifying only needed locales). Вы можете выбрать, какие локали (включая раскладки и установки клавиатуры, времени и т.д.) будут собраны при установке. Если вы не ограничите локали, используемые в вашей системе, будут собраны все имеющиеся локали, начиная с aa\_DJ (локаль Афар для Джибути) через en\_GB (английская локаль для Великобритании) и заканчивая zu\_ZA.utf8 (локаль Зулу для Южной Африки). При ограничении сборки локалей вы можете сохранить до 90% места, необходимого для Glibc в вашей системе, сохранить время, необходимое на сборку ненужных вам локалей и, как следствие, уменьшить общее время компиляции. Если вы действительно не нуждаетесь в них всех (а трудно представить, кому это могло бы понадобиться), вы можете ограничить их необходимым минимумом.

Используйте флаг USE userlocales для сборки только тех локалей, которые указаны в /etc/locales.build.

Отредактируйте /etc/make.conf в вашем любимом редакторе и поместите userlocales где-нибудь между "" в строке USE="".

Другой способ - изменить /etc/portage/package.use используя данную команду: Code: Activating the userlocales USE flag for glibc

```
echo "sys-libs/glibc userlocales" >> /etc/portage/package.use
```

После этого вы можете указать локали, которые хотите использовать: Файл: nano -w /etc/locales.build

#Читайте комментарии в начале файла для большей информации!

```
en_US/ISO-8859-1
```

```
en_US.UTF-8/UTF-8
```

```
#en_GB/ISO-8859-1
```

```
#en_GB.UTF-8/UTF-8
```

```
de_DE/ISO-8859-1
```

```
de_DE@euro/ISO-8859-15
```

```
ru_RU.UTF-8/UTF-8
```

В glibc-2.3.6-r4 или glibc-2.4-r2, был удалён флаг userlocales. Вы должны править

файл /etc/locale.gen и удалить /etc/locales.build.

Вы можете воспользоваться для этого следующими командами: Code: Convert locales.build to locale.gen

```
cd /etc
```

```
grep '^[^#].*' locales.build | sed 's:/: :> locale.gen
```

```
rm locales.build
```

```
nano -w locale.gen
```

Файл: nano -w /etc/locale.gen

#Читайте комментарии в начале файла для большей информации!

```
en_US.UTF-8 UTF-8
```

```
en_US ISO-8859-1
```

```
ru_RU.UTF-8 UTF-8
```

```
de_DE ISO-8859-1
```

```
de_DE@euro ISO-8859-15
```

Вы можете найти правильные значения локалей в файле /usr/share/i18n/SUPPORTED.

Помогите! Я не знаю, какими должны быть настройки моей локали!

Не волнуйтесь, это просто. Записи в этом файле должны быть в формате <locale>/<charmap>. <locale> это локаль из каталога /usr/share/i18n/locales, а <charmap> -- это имя одного из файлов каталога /usr/share/i18n/charmaps/. Нужно только понимать два момента: первое, локаль с новой валютой, такой как евро, требует изменения @euro, описанного в примере выше. Второе, локали UTF-8 требуют от вас добавления .UTF-8 в конец определения локали (нет, я не знаю, зачем, но иначе оно не работает).

Если вы В САМОМ ДЕЛЕ хотите минимизировать ваши локали, вам всего лишь нужно перевести вашу систему на Юникод. Вообще-то пока есть причины этого не делать, но это уже становится стандартом de facto, так как несёт множество преимуществ перед старыми форматами ISO, ASCII и другими. Больше об этом можно прочесть [здесь](#) и [здесь](#), а также непосредственно в соответствующем руководстве с сайта Gentoo.

Вам просто нужно определить следующее: Файл: nano -w /etc/locales.build

#Как раньше!

en\_US.UTF-8/UTF-8

ru\_RU.UTF-8/UTF-8

Обратите внимание, я оставил строку US UTF-8. Её лучше не убирать, так как некоторые программы не соберутся без неё.

Сделайте это до установки Gentoo

Вы можете выполнить эти настройки до начальной сборки системы с нуля. Просто следуйте инструкциям из Gentoo Handbook. Это позволит меньше волноваться о возможных недоделках программ после сборки и наслаждаться быстрой сборкой.

Сделайте это после установки Gentoo

Если glibc уже установлена, вам потребуется пересобрать glibc: Code: Пересборка glibc

emerge glibc

Вы также можете пересобрать world, если пакеты были скомпилированы с поддержкой множества локалей, но полученное ускорение работы пакетов и сэкономленное дисковое пространство не стоят затрат времени. Придётся затратить очень много времени на пересборку всей системы.

Возможно, вам стоит обратить внимание на пакет localepurge, который может очистить систему от любых установленных man страниц или info-файлов на языках не используемых в вашей системе. Вначале прочитайте man к localepurge, что бы определить локали, которые необходимо пропустить в /etc/locale.nopurge.

Для большей информации об управлении локалями читайте Gentoo Linux Localization Guide.

Или на русском языке Руководство по русской локализации Gentoo Linux.

Эта оптимизация полностью безопасна (Кроме случаев, если вы не можете читать на языке, который установили!).

Безопасность

Как безопасность относится к оптимизации? Итак, это оптимизация безопасности,

поскольку ранее было сказано, как оптимизация зависит от вашей системы и как всё это работает. Если это сервер, вы, к примеру, можете пожертвовать оптимизацией в угоду безопасности, или выполнить обе оптимизации.

## Стабилизация Glibc

Это предполагает (как вы, возможно, догадывались...) использование флага `USE hardened`. Это сделает вашу систему более устойчивой к нападениям разного вида.

## Усиление стека

Используйте флаг `erandom` если вам необходимо использование модуля случайных чисел, установленного в вашем ядре.

# Подключение USB-flash

## Подключение USB-flash к компьютеру

### Введение

В Linux я пользователь новый и с этим вопросом никогда не сталкивался, и более того - ядро по жизни собирал без поддержки USB устройств, т.к. ничего такого у меня просто нет :), но вот, как говорится - пришлось и срочно. И как оказалось - это довольно несложно.

Итак, по прочтении парочки-тройки вопросов связанных с USB, и небольшого количества теории - я без проблем подмонтировал USB флешку. Примечание: Что такое флешка и USB описывать не буду, т.к. если читатель этого не знает - то имхо ему следует сначала прочесть несколько других статей.

Итак, начнем

### Конфигурация ядра

USB флешки видны в Linux как `scsi` устройства, ибо поддержка `usb` устройств в ядре Linux реализуется именно посредством эмуляции `scsi`, т.е. типа `/dev/sda1(2,3)` и т.д., смотря сколько партиций.

Следовательно, чтобы была возможность подключения флешки, сначала необходимо добавить поддержку `scsi` в ядро Linux Kernel Configuration: Включаем поддержку `SCSI`

Device Drivers --->

SCSI device support

```
<*>legacy /proc/scsi/ support
--- SCSI support type (disk, tape, CD-ROM)
<*> SCSI disk support
<*> SCSI generic support
--- Some SCSI devices (e.g. CD jukebox) support multiple LUNs
<*> Probe all LUNs on each SCSI device
```

Далее добавляем непосредственно поддержку USB (я не говорю о клавиатурах или других устройствах USB, а говорю только о флешке) Linux Kernel Configuration: Поддержка USB

Device Drivers --->

USB support

```
<*> Support for Host-side USB
<*> USB device filesystem
<*> EHCI HCD (USB 2.0) support
< > OHCI HCD support
<*> UHCI HCD (most Intel and VIA) support
<*> USB Mass Storage support
```

Примечание: В разделе USB Mass Storage support есть куча других устройств - я там не отмечал ничего, у меня заработало и так. Подозреваю, что для специфических устройств (всмысле каких то навороченных флешок) нужно будет что нибудь еще там отметить.

Кроме того, для этих всяких сложных флешок скорее всего придется добавить MTD в ядро Linux Kernel Configuration: MTD

Device Drivers ---->

Memory Technology Devices (MTD) ---->

ну и там смотрим, что нужно. Примечание: Для большинства современных флэшек, USB носителей и цифровых фотоаппаратов поддержка MTD не требуется.

Монтирование

После загрузки с новым ядром, нажимаем Alt+F12 и вставляем флешку - если с ядром все ок- то сразу появятся сообщения, что мол есть устройство USB , потом пишется, что есть /dev/uba ну и т.п.

Если этого не происходит, делаем

```
dmesg | grep usb
```

Если и здесь ничего нет, то советую перегрузиться и посмотреть включенны ли USB контроллеры в BIOS :) как это не смешно звучит, но я сам себе это отрубил, когда ядро собирал, а потом включить забыл, и минут 15 гадал - чего же оно не работает :)

Если `dmesg | grep usb` выдает результат, а когда вставляем флешку - никакой реакции, то нужно проверить еще раз - все ли отмечено в ядре.

Вобщем - надеемся , что все заработало и теперь нужно смонтировать флешку.

Добавляем в `/etc/fstab` строку Файл: `/etc/fstab`

```
/dev/uba1 /mnt/usbdisk auto user,noauto,rw 0 0
```

и можно монтировать! но есть одно но, в опциях я не указал кодировки, т.к. кодировка по умолчанию указана в разделе Native Languages конфигурации ядра.

Не будет лишним напомнить отмонтировать флешку перед тем как вытаскивать ее!

Если такой вариант вас не устраивает, то вы можете установить пакет `submount` и монтировать флэшку вот так: Файл: `/etc/fstab`

```
/dev/uba1 /mnt/usb subfs fs=vfat,auto,umask=0,quiet,sync 0 0
```

Параметр `sync` необходим чтобы отключить отложенную запись и избежать потерю данных при копировании файлов на флэшку.

Вроде бы все. Удачи всем!

И естественно- за любые комментарии, дополнения, уточнения и исправления ошибок – буду только благодарен :)

надеюсь, кому-то помог tradakad

исправил касательно монтирования при помощи `submount` (dernik)

Монтирование под kde

Установите пакеты: `hal`, `dbus`, and `hotplug`

Добавьте флаг `hal` в `/etc/make.conf`

```
emerge -avt kdbase-kioslaves
```

Убедитесь, что `hal` используется всеми пакетами:

```
emerge -DNu world
```

```
rc-update add dbus default
```

```
/etc/init.d/dbus start
```

```
rc-update add hald default
```

```
/etc/init.d/hald start
```

так же можно монтировать устройства без изменения `/etc/fstab` с помощью

emerge -av pmount

Добавьте себя в групп plugdev

gpasswd -a USER plugdev

Включите "Storage media" ("Устройства хранения данных" в русском варианте) в systray (на панель). Наслаждайтесь :)

Автоматическое монтирование

Если лень замарачиваться каждый раз при втыкании флэшки или сидюка, а хочется, чтоб как в одной другой ОС, добавьте параметр managed в /etc/fstab к требуемому устройству, например Файл: /etc/fstab

```
/dev/cdrw /mnt/cdrom iso9660 user,noauto,ro,managed 0 0
```

```
/dev/sda /mnt/flash vfat exec,user,noauto,sync,managed 0 0
```

Типа, все, теперь при появлении флэшки в системе будет появляться иконка на рабочем столе.

ps. распишите эту статью подробнее. Добавьте gpasswd в статью: [http://gentoo-wiki.com/HOWTO\\_D-BUS,\\_HAL,\\_KDE\\_media/](http://gentoo-wiki.com/HOWTO_D-BUS,_HAL,_KDE_media/)

## Чтение-запись NTFS с использованием драйвера ntfs-3g

Введение

"Драйвер ntfs-3g является открытым, лицензированным по GPL, драйвером NTFS для Linux, созданным в рамках проекта Linux-NTFS. Он предоставляет полный доступ к разделам NTFS (чтение-запись), кроме работы с зашифрованными файлами и записи сжатых файлов. Так же не поддерживается смена владельца файла и его прав доступа. Технически он является сильно усовершенствованным драйвером ntfsmount. Улучшена функциональность, качество и добавлены дополнительные возможности." (Szakacsits Szabolcs, автор ntfs-3g)

Автор создал драйвер, который гораздо лучше пригоден для использования, чем другие драйвера NTFS для Linux, и при этом не менее быстрый, чем драйвера родных файловых систем Linux - временами в два раза быстрее чем родной модуль файловой системы EXT3 в ядре.



Для получения более подробной информации посетите: [1] Предупреждение: Драйвер имеет статус БЕТА, имеются некоторые не решённые вопросы, возможна потеря данных и/или зависания при попытке примонтировать раздел. Он НЕ работает на архитектурах, отличных от x86 и amd64.

СДЕЛАЙТЕ РЕЗЕРВНУЮ КОПИЮ ДАННЫХ ПЕРЕД ЛЮБЫМИ ПОПЫТКАМИ ИСПОЛЬЗОВАНИЯ !!! В ЛЮБОМ СЛУЧАЕ - ВЫ ПРЕДУПРЕЖДЕНЫ !!!

## Установка

ebuild драйвера зависит от sys-fs/fuse (portage установит его автоматически).

Поскольку это программное обеспечение находится на стадии бета-тестирования, вы должны размаскировать его:

```
# echo "sys-fs/ntfs3g ~x86" >> /etc/portage/package.keywords
```

И установить его:

```
# emerge sys-fs/ntfs3g
```

Модуль ядра fuse должен быть загружен перед использованием драйвера

```
# modules-update
```

```
# modprobe fuse
```

## Примеры использования

Монтирование раздела:

```
# ntfs-3g /dev/hda1 /mnt/windows
```

Полный доступ для всех пользователей, без ограничений прав доступа, и поддержкой заданной локали (В примере используется ru\_RU.utf8, однако у вас она может отличаться):

```
# ntfs-3g /dev/hda1 /mnt/windows -o silent,umask=0,locale=ru_RU.utf8
```

Размонтирование

```
# fusermount -u /mnt/windows
```

Прочтите страницу руководства по ntfs-3g для получения более подробной

информации.

```
$ man ntfs-3g
```

Монтирование при загрузке:

Для автоматической загрузки модуля fuse при старте системы выполните:

```
# echo "fuse" >> /etc/modules.autoload.d/kernel-2.6
```

Отредактируйте /etc/fstab:

```
# echo "/dev/hda1 /mnt/windows ntfs-3g silent,umask=0,locale=ru_RU.utf8 0 0"
>> /etc/fstab
```

Вопрос обновления ядра

Если после обновления ядра драйвер перестал работать, сделайте следующее:

```
# emerge sys-fs/fuse
```

```
# modules-update
```

```
# modprobe fuse
```

Отказ от гарантий

Автор этого документа не отвечает за какие либо последствия, связанные с использованием этого драйвера. Всё что вы делаете - вы делаете на свой страх и риск.

## ***Сетевые сервисы***

### **Установка Apache2**

Установка

Apache2 сейчас работает на большинстве системах без проблемм, если у вас возникли проблемы прочтите раздел Common Problems это может помочь.

Начнем установку apache2: добавьте "apache2" к вашим USE флагу в файле /etc/make.conf и выполните

```
emerge apache
```

Запуск Apache

Запустите Apache2:

```
/etc/init.d/apache2 start
```

Если вы хотите запускать apache2 при запуске системы выполните команду:

```
rc-update add apache2 default
```

Посмотрите init scripts section для дополнительной информации.

Теперь у вас есть работающий веб сервер Apache2. Перейдите в вашем браузере на страницу <http://localhost/> и вы должны увидеть страницу приветствия.

Apache's simplest functionality is just serving plain old files. The basic pattern is: replace <http://hostname/> with the directory (document root) that contains the website. По умолчанию Apache2 сконфигурирован таким образом, что путь к страницам сайта `/var/www/localhost/htdocs/`. Для примера веб-страница:

<http://hostname/index.html> находится `/var/www/localhost/htdocs/index.html`

<http://hostname/foo/bar.txt> находится `/var/www/localhost/htdocs/foo/bar.txt`

Вы найдете в `/var/www/localhost/htdocs/index.html` HTML код страницы приветствия которую вы видите на <http://localhost/>. Замечаем что Apache отображает `htdocs/index.html` когда вы пытаетесь открыть `htdocs/`. Это особенность протокола HTTP. Apache не может передать директорию но может показать содержимое. Список (index) файлов в директории может быть передан. Apache ищет страницу со специальным именем `index` либо генерирует список файлов в директории. Если у вас есть страница с именем `'index'` то будет отображена она; в этом случае вы не сможете посмотреть список файлов в директории.

## Модули

Apache is extremely versatile. Он может сёрфить файлы используя HTTP либо сёрфить файлы и помощью FTP. Он может передать файл с жесткого диска, либо вывод PHP скрипта. Для реализации этого Apache использует модули. Для приложения используют для этого плагины. При добавлении модули добавляют функциональность. Их также можно устанавливать, удалять, пересобирать (перекомпилировать).

Модули Apache обычно называются `mod_something`. Некоторые уже включены в Апахь а некоторые нужно добавлять отдельно. В портяжах содержатся множество модулей. ТПроцесс установки выкладит примерно так:

```
emerge module
```

отредактируйте `/etc/conf.d/apache2` для активации добавьте `-D MOD`

опционально /etc/apache2/modules.d/xy\_module

Добавьте директивы конфигурации в httpd.conf либо .htaccess

emerge mod\_perl

nano /etc/conf.d/apache2

# change APACHE\_OPTS="" to APACHE\_OPTS="-D PERL"

Документацию по конкретным модулям вы найдете Apache Index in this wiki. Вы также можете почитать the documentation для дополнительной информации о модулях Apache.

## Конфигурация

В файле httpd.conf, (/etc/apache2/httpd.conf,) который поставляется с Gentoo хранятся большинство настроек Apache. However, it probably does both more and less than you need it to. Apache configuration files have a consistent syntax.

Любые строки начинающийся с # игнорируются

# Apache не анализирует написанное здесь

# это комментарии

Некоторые строки начинаются с директивы и могут иметь один или несколько аргументов.

SomeDirective one or more arguments

Директивы могут быть объединены в секции. Разделы обычно заключены в угловые скобки.

<Section>

# Will only apply when the section matches

AnotherDirective

</Section>

В разделе могут быть подразделы. Вот часть файла httpd.conf:

# If mod\_alias is loaded

<IfModule mod\_alias.c>

# Alias is a directive and it only applies if mod\_alias is loaded

Alias /icons/ "/usr/share/httpd/icons/"

# If the file is in the directory

<Directory "/usr/share/httpd/icons">

# Options will only apply if:

```
# mod_alias is loaded AND
# the file is in the directory
Options Indexes MultiViews
</Directory>
</IfModule>
```

Вы можете прочитать подробнее configuration files и sections в официальной документации Apache.

## Common Problems

### SSI Not Working

When configuring for SSI (Server Side Includes), an error may occur:

```
mod_include: Options +Includes (or IncludesNoExec) wasn't set, INCLUDES filter
removed
```

The problem is that setting Options +Includes in either .htaccess or httpd.conf is overwritten by the additional configuration file as defined at the end of httpd.conf.

```
Include /etc/apache2/vhosts.d/*.conf
```

You need to edit this additional configuration file such that

```
AllowOverride None
```

Is replaced by

```
AllowOverride Options
```

### Could Not Open Error Log

While starting Apache, it prints:

```
Error while starting apache: (2)No such file or directory: apache2: could not open error log
file /usr/lib/apache2/logs/error_log.
```

/usr/lib/apache2/logs should be a symlink pointing to /var/log/apache2 . Check it using:

```
ls -la /usr/lib/apache2/logs
```

(note the lack of a slash on the end). If /var/log/apache2 is missing, create it and make sure you give apache ownership:

```
mkdir /var/log/apache2
```

```
chown apache:apache /var/log/apache2
```

If the symlink `/usr/lib/apache2/logs` is missing, you can create it:

```
ln -s /var/log/apache2 /usr/lib/apache2/logs
```

You don't need to set permissions on the symlink.

## Check the Logs

See `/var/log/apache2/error_log` for errors, especially towards the end of the file. You may find `tail` useful because it displays only the last few lines of a file:

```
tail /var/log/apache2/error_log
```

If you wish to keep an eye on the log the `-f` option for `tail` may be useful:

```
tail -f /var/log/apache2/error_log
```

Here's one error you might see:

```
Error: [alert] (EAI 2)Name or service not known: mod_unique_id: unable to find IPv4 address of ""
```

With the base installation "mod\_unique\_id" is turned on, this can cause problems, notably the server not starting. Simply comment out this module in `/etc/apache2/httpd.conf` and the problem will be solved.

(Your config file might be `/etc/apache2/conf/apache2.conf`)

## Forbidden User Directories

If the server is returning "403 Forbidden" while accessing `http://server/~username/` Make sure Apache (usually user `apache` and group `apache`) has read access to username's home directory and `public_html` (or equivalent). You can grant everyone read access using:

```
chmod 755 ~username/ ~username/public_html/
```

## Not Enough Entropy

If Apache2

accepts connections

does not respond to clients

creates exactly one process

is not stopped by  
`/etc/init.d/apache2 stop`

Check to see how much entropy is available using:  
`cat /proc/sys/kernel/random/entropy_avail`

If little entropy (less than 100) is available, Apache2 is probably waiting for more so it can generate the secret for digest authentication (`mod_auth_digest`). To generate more entropy, just do something else for a little while. Grepping the kernel or emerging a package usually works well.

The `video-entropyd` and `audio-entropyd` supply `/dev/random` with entropy gathered from your video and audio devices, respectively. If you have a hardware random number generator (RNG), you can emerge `rng-tools` and run `rngd`.

If there's still a shortage of entropy, you can enable the `urandom` USE flag and re-emerge APR and Apache2. This makes APR use `/dev/urandom`, which falls back to a pseudorandom number generator when there isn't enough entropy. The program gets a number immediately, but it is cryptographically weaker. This is okay for some things (e.g. `solitaire`), but completely unacceptable for others (like PGP key generation).

Confusing config files

If you start the Apache2 server with the startup script `/etc/init.d/apache2` check to see if the line

```
local myconf="/etc/apache2/httpd.conf"
```

from `/etc/init.d/apache2` points to your configuration script. If it points to `apache.conf` and you use `httpd.conf`, make the necessary adjustments.

See Also

Configure LAMP (Linux, Apache, MySQL, and Python/PHP/Perl) - A popular web server combination

Apache Installation & Configuration

How to install `mod_security` for Apache

## **Настройка iptables для начинающих**

Введение

В сети мало документации по iptables рассчитанной на новичков. Мы же попытаемся восполнить этот пробел. Рассмотрим основы составления правил, а также некоторые дополнительные модули которые помогут сделать жизнь легче.

Прежде чем двигаться дальше - убедитесь, что ...

Всё ваше аппаратное обеспечение работоспособно. То есть Вы подключили все оборудование, модули грузятся, устройства видны в системе. Полезно в начале проверить, что соединение с интернетом возможно и без всяких там iptables. Нет ничего хуже, чем в течении многих часов возиться с упрямой программой, ругая её почём зря, а потом понять, что сетевая карта не работает или модем сконфигурирован неправильно.

Вы имеете представление о сетевых технологиях и Вы знакомы с администрированием Linux и Gentoo Linux в частности. То есть необходимы навыки работы с такими базовыми вещами как ifconfig, rc-update, /etc/conf.d/net, и так далее. Если для Вас это пока пустые звуки, то, пожалуйста предварительно изучите Настольную книгу Gentoo и Linux Help's Networking Basics 101

## Конфигурация ядра ОС Linux

Все что вам нужно - это включить поддержку iptables.

Networking --->

Networking Options---->

Network Packet Filtering (replace Ipchains)--->

Netfilter Configuration

Включим все опции как модули (хотя с точки зрения безопасности модули следует вообще отключить, монолитное ядро надежнее, хотя и медленнее).

## Установка iptables

Далее вы должны установить пакет iptables: emerge iptables

## Проверка сети

Предположим, что у нас есть 2 сетевых интерфейса: eth0 - локальная сеть и ppp0 - интернет соединение.

Проверим работоспособность сети командой ping: Code: ping

ping www.google.com

ping 192.168.1.78

ping 192.168.2.77



## Запуск iptables

Запустим iptables: `/etc/init.d/iptables start`

Эта команда загрузит основные модули и создаст цепочки в ядре Linux. Теперь добавим iptables в автозагрузку: `rc-update add iptables default`

## Использование /etc/init.d/iptables

Скрипт `/etc/init.d/iptables` понимает несколько команд (`/etc/init.d/iptables <команда>`), некоторые из них:

`start` - запуск iptables. Восстанавливает все правила (правила хранятся в `/var/lib/iptables/rules-save`);

`stop` - сброс всех цепочек;

`save` - сохранение всех правил.

## Создание правил

Практически все правила можно привести к виду: `iptables -A ЦЕПОЧКА ПАРАМЕТРЫ_ПАКЕТА -j ДЕЙСТВИЕ`

## Цепочки

Все изменения будем проводить над таблицей `filter`, именно она отвечает за фильтрацию пакетов. В таблице `filter` существует 3 цепочки: `INPUT`, `OUTPUT` и `FORWARD`. В каждой цепочки свой "тип" пакетов:

`INPUT` - пакеты пришедшие к Вам. То есть входящий трафик.

`FORWARD` - пакеты которые предназначены для другого узла, то есть транзитный трафик.

`OUTPUT` - пакеты, которые уходят от нас, или исходящий трафик.

Работают с цепочками так: `iptables <опция> <цепочка>`

Для работы с цепочками предусмотрены следующие опции:

`-A` - добавление нового правила в цепочку. Правило будет добавлено в конец цепочки.

`-I` - добавление правила не в конец, а туда куда вы укажете. Например команда:

iptables -I INPUT 2 bla-bla-bla - сделает наше правило вторым.

-D - удаление правила. Например для удаления пятого правила введите:

```
iptables -D INPUT 5
```

-F - сброс всех правил цепочки. Нужно, например, при удалении ненужной цепочки.

-N - создание пользовательской цепочки. Если не хотите создавать кашу в каждой цепочке, то создайте несколько дополнительных цепочек. Синтаксис такой: iptables -N ЦЕПОЧКА. Только русские буквы, конечно, использовать нельзя.

-X - удаление пользовательской цепочки.

ПРИМЕЧАНИЕ: Удалить цепочки INPUT, OUTPUT и FORWARD нельзя.

-P - установка политики для цепочки. Например:

```
iptables -P ЦЕПОЧКА ПОЛИТИКА
```

## Параметры пакетов

Итак по каким параметрам можно фильтровать пакеты? Рассмотрим самые основные.

### Источник пакета

Для фильтрации по источнику используется опция -s. Например запретим все входящие пакеты с узла 192.168.133.133: iptables -A INPUT -s 192.168.133.133 -j DROP

Можно использовать доменное имя для указания адреса хоста. То есть: iptables -A INPUT -s test.host.jp -j DROP

Также можно указать целую подсеть: iptables -A INPUT -s 192.168.133.0/24 -j DROP

Также вы можете использовать отрицание (знак !). Например так - все пакеты с хостов отличных от 192.168.133.156 будут уничтожаться: iptables -A INPUT -s ! 192.168.133.156 -j DROP

### Адрес назначения

Для этого нужно использовать опцию -d. Например запретим все исходящие пакеты на хост 192.168.156.156: iptables -A OUTPUT -d 192.168.156.156 -j DROP

Как и в случае с источником пакета можно использовать адреса подсети и доменные имена. Отрицание также работает.

## Протокол

Опция -p указывает на протокол. Можно использовать all, icmp, tcp, udp.

## Порт источника

Указывает на порт с которого был прислан пакет. Вот синтаксис: `iptables -A INPUT -p tcp --sport 80 -j ACCEPT`

Для указания порта необходимо указать протокол (tcp или udp). Можно использовать отрицание.

## Порт назначения

Порт назначения. Синтаксис: `iptables -A INPUT -p tcp --dport 80 -j ACCEPT`

Как и в случае с портом источника нужно указать протокол. Можно использовать отрицание.

## Действия над пакетами

Проку от того, что мы укажем параметры пакета нет. Нужно указать, что надо с ним делать. Для этого служит опция -j. Рассмотрим основные действия:

ACCEPT - разрешить пакет.

DROP - уничтожить пакет.

REJECT - будет отправлено ICMP сообщение, что порт недоступен.

LOG - информация об этом пакете будет добавлена в системный журнал (syslog).

В качестве действия можно указать и имя пользовательской цепочки. Например перекинем все пакеты с локальной сети в цепочку, где будет производиться дополнительная проверка: `iptables -A INPUT -s 192.168.200.0/24 -j LOCAL_NET`

## Пример правил

В большинстве случаев пользователю достаточно выполнить такую последовательность команд: `iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT` `iptables -A INPUT -i lo -j ACCEPT` `iptables -P`

## INPUT DROP

Вот собственно и вся настройка. На первый взгляд непонятно, что мы тут вообще сделали. Поэтому ознакомимся с основами составления правил.

### Модули

Может сложиться впечатление, что возможностей у iptables маловато. Однако с использованием модулей iptables получит просто безграничные возможности. Для указания модуля используется опция -m. Например: iptables -A INPUT -m модуль bla-bla

-m owner

Добавляет следующие опции (опции только для цепочки OUTPUT):

--uid-owner UID - UID программы пославшей пакет.

--gid-owner GID - GID программы пославшей пакет.

--pid-owner PID - PID программы пославшей пакет.

--sid-owner SID - SID (идентификатор сессии) производится проверка SID пакета, значение SID наследуются дочерними процессами от "родителя".

--cmd-owner NAME - имя программы пославшей пакет.

-m multiport

Позволяет указывать не по одному порту, а сразу несколько:

--source-ports порт1,порт2 - список портов, с которых пришел пакет;

--sports порт1,порт2 - укороченный аналог --source-ports;

--destination-ports порт1,порт2 - список портов назначения;

--dports порт1,порт2 - укороченный аналог --destination-ports;

--ports порт1,порт2 - проверяет как исходящий так и входящий порт пакета.

-m state

Предназначен для указания состояния пакета с помощью опции --state. Доступны следующие типы пакетов:

NEW - пакет устанавливающий новое соединение.

ESTABLISHED - пакет от уже установленного соединения.

RELATED - новый пакет уже установленном соединении.

-m mac Проверяет соответствие мак-адреса в пакете, например: `iptables -A INPUT -s 192.168.0.1 -m mac --mac-source 00:65:3F:ED:12:98 -j DROP` Также допускается отрицание при помощи !

## Подробная настройка iptables

### Введение

Документация по iptables в сети не рассчитана на новичков. В этой статье будет кратко и быстро описаны команды, затем, возможно, будут правки и добавления для расширенного объяснения. Так что это будет минимальная установка которую мы в дальнейшем расширим и упрочним с помощью правил.

Так же примите во внимание, что будет использовано рррое соединение и 2.6.x ядро. Для настройки сетевой карты надо будет заменить rpp0 на eth0 (или подходящую по смыслу вашу сетевую карточку глядящую в интернет)

Прежде чем двигаться дальше - убедитесь, что ...

1.Всё ваше аппаратное обеспечение работоспособно. То есть вы все подключили, модули грузятся, устройства видны в системе. Полезно вначале проверить, что соединение с интернетом возможно и без всяких там iptables. Нет ничего хуже, чем в течении многих часов возиться с упрямой программой, ругая её почём зря, а потом понять, что сетевая плата не работает или модем сконфигурирован неправильно.

2.Вы читали ману по теме. Предполагается, что пока вы изучаете это руководство, man iptables постоянно открыт в соседнем терминале для точного понимания и уточнения, что же та или иная команда означает на самом деле.

3.Вы имеете представление о сетевых технологиях и Вы знакомы с администрированием Linux и Gentoo Linux в частности. То есть необходимы навыки работы с такими базовыми вещами как ifconfig, rc-update, /etc/conf.d/net, и так далее. Если для вас это пока пустые звуки, то, пожалуйста предварительно изучите The Gentoo Handbook и Linux Help's Networking Basics 101

### Конфигурация ядра

Все что вам нужно - это включить поддержку iptables. Linux Kernel Configuration: Включение IPTables

Device Drivers--->

Networking Support--->

Networking Options---->

Network Packet Filtering (replace Ipchains)--->

Netfilter Configuration

Я включил все опции как модули (с тем расчетом, что я захочу попробовать другие опции позже) и добавил `ip_tables` в `modules.autoload`. Это загрузит еще несколько модулей в качестве зависимостей. Модуль `ip_conntrack` необходим для "statefull" фильтрация, то есть для отслеживания соединений. Для запуска скриптов выполните команду :

```
# modprobe ip_tables
```

Необходимые утилиты

Далее вы должны установить пакет `iptables`:

```
# emerge iptables
```

Конфигурация интерфейсов

Сетевые протоколы Прикладной уровень HTTP, SMTP, SSH, SNMP, FTP, NNTP, NTP, LDAP, ...

Сеансовый уровень TLS, SSL, RPC, WSP...

Транспортный уровень TCP, UDP, SCTP, ICMP, OSPF, RSVP, VRRP, RTP, DCCP ...

Сетевой уровень IPv4, IPv6, ARP, RARP, MPLS, IPX ...

Канальный уровень Ethernet, 802.11, xDSL, Fibre Channel, FDDI, ATM, ISDN ...

В моем случае имеется 3 сетевых адаптера. Один подключен к WAN через `pppoe`. Другие два - к моей внутренней сети. Для того, чтобы не было проблем с `iptables` и маскардингом (NAT'ом), они должны быть сконфигурированы для различных подсетей. Для примера, 2 сетевых адаптера подключены к моим внутренним компьютерам (внутренние сетевые интерфейсы). Им присвоены IP-адреса: 192.168.1.1 и 192.168.2.1.

Следует заметить, что будет лучше если подключать эти внутренние адаптеры в любое сетевое устройство, такие как свитч и хаб. Для `pppoe` подключений мы должны убедиться, что сетевой адаптер подключен к внешнему миру, то есть внешним интерфейсам не присвоены никакие IP-адреса. Его запись в `/etc/conf.d/net` должна оставаться пустой. Это делается потому, что `pppoe` выступает в качестве виртуального устройства, которое включается вслед за сетевым интерфейсом. Мы также должны присвоить правильные сетевые маски и широковещательные адреса для этих интерфейсов. Ваш `conf.d/` должна выглядеть примерно так:

СерверФайл: `/etc/conf.d/net`

```
# Для pppoe подключений вы не должны указывать значения для [[eth0]],
```

```
# просто добавьте net.ppp0 или rc-pppoe в default уровень загрузки.
```

```
iface_eth0="192.168.1.98 broadcast 192.168.1.255 netmask 255.255.255.0"
```

```
iface_eth1="192.168.2.98 broadcast 192.168.2.255 netmask 255.255.255.0"
```

Заметьте, что не было указано никаких шлюзов.

Клиент1Файл: /etc/conf.d/net

```
iface_eth0="192.168.1.77 broadcast 192.168.1.255 netmask 255.255.255.0"  
gateway="eth0/192.168.1.1"
```

Клиент2Файл: /etc/conf.d/net

```
iface_eth0="192.168.2.77 broadcast 192.168.2.255 netmask 255.255.255.0"  
gateway="eth0/192.168.2.1"
```

Шлюз для клиентов установлен на внутренний IP сетевого интерфейса сервера, что и логично. Теперь добавьте все интерфейсы в default уровень загрузки и перезапустите подключения:

```
# rc-update add net.eth1 default && rc-update add net.eth2 default && rc-update add  
net.ppp0 default
```

и

```
# /etc/init.d/net.eth1 start && /etc/init.d/net.eth2 start && /etc/init.d/net.ppp0 start
```

Для клиентов:

```
# /etc/init.d/net.eth0 restart
```

Проверка настроек

Теперь убедитесь в том, что ваш сервер подключен к интернету, а также все интерфейсы могут пинговать друг друга. Для сервера: Code: ping

```
ping www.google.com;
```

```
ping 192.168.1.78
```

```
ping 192.168.2.78
```

```
ping 192.168.1.77
```

```
ping 192.168.2.77
```

Убедитесь что у клиентов правильно указаны DNS-сервера в /etc/resolv.conf

## Scripting

Теперь интересная часть... iptables и NAT(трансляция адресов). Для начала сделаем простое перенаправление адресов с минимальными правилами, чтобы убедиться что можем выходить в сеть. Предупреждение: Если вы параноик, то это не самое секретное, что можно сделать... мы открываемся в сеть с мизерной защитой. Однако будем считать, что сеть настраиваем для дома или для игрового класса.

Файл: /var/lib/iptables/rules-save

```
#!/bin/bash
```

```
IPTABLES='/sbin/iptables'
```

```
# Определяем интерфейсы
```

```
EXTIF='ppp0'
```

```
INTIF1='eth1'
```

```
INTIF2='eth2'
```

```
# Включаем форвардинг ip в ядре.
```

```
/bin/echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
# Сбросить правила и удалить цепочки
```

```
$IPTABLES -F
```

```
$IPTABLES -X
```

```
# Включаем маскардинг для разрешения доступа в интернет
```

```
$IPTABLES -t nat -A POSTROUTING -o $EXTIF -j MASQUERADE
```

```
# Форвардить сетевой трафик с $INTIF1 на интернетовский интерфейс $EXTIF
```

```
$IPTABLES -A FORWARD -i $INTIF1 -o $EXTIF -m state --state NEW,ESTABLISHED -j ACCEPT
```

```
# Форвардить сетевой трафик с $INTIF2 на интернетовский интерфейс $EXTIF
```

```
$IPTABLES -A FORWARD -i $INTIF2 -o $EXTIF -m state --state NEW,ESTABLISHED -j ACCEPT
```

```
#echo -e "      - Разрешаем доступ к SSH серверу"
```

```
$IPTABLES -A INPUT --protocol tcp --dport 22 -j ACCEPT
```

```
#echo -e "      - Разрешаем доступ к HTTP серверу"
```

```
$IPTABLES -A INPUT --protocol tcp --dport 80 -j ACCEPT
```



```
# Блокируем все прочие попытки доступа на $EXTIF
$IPTABLES -A INPUT -i $EXTIF -m state --state NEW,INVALID -j DROP
$IPTABLES -A FORWARD -i $EXTIF -m state --state NEW,INVALID -j DROP
```

Проверка на работоспособностьПримечание: Этот скрипт написан кем-то в сетевом форуме... Адрес автора утерян, имя... В общем, спасибо ему, пусть и без имени. (Надеюсь, он не обидится).

Теперь проверяем могут ли наши пользователи выйти в интернет или подключиться к серверу по ssh. Если все нормально, то можно переходить ниже по тексту. Если нет - проверьте синтаксические ошибки и прочее... Удостоверьтесь, что IP-адреса и маски клиентов и сервера введены правильно... Ну или...

Если все работает, как задумали, сохраняем конфигурацию:

```
# /etc/init.d/iptables save
```

И бэкапим вашу рабочую конфигурацию для возможного восстановления "как было":

```
# cp /var/lib/iptables/rules-save /var/lib/iptables/rules.working
```

Проверим iptables start-up скрипт перед тем как добавить iptables в default runlevel:  
Code: Проверка скрипта

```
/etc/init.d/iptables start
```

```
/etc/init.d/iptables stop
```

```
/etc/init.d/iptables start
```

Смысл в запуске-остановке-запуске в том, что у нас нет скрипта запуска iptables... поэтому нужно "инициализировать" статус перед остановкой. Остановка, по существу, обнуляет настройки и возвращает все к исходному. Перезапуск покажет нам работает ли наша сеть после перезагрузки. Если все в порядке, то добавляем iptables в default runlevel:

```
rc-update add iptables default
```

Не забудем также установить в /etc/sysctl.conf:

```
net.ipv4.ip_forward = 1
```

Часть II рассматривает правила установок и политики для обороны Вашего маршрутизатора.

### ЧАСТЬ III - Оборона Firewall

Далее мы сделаем наш, уже работающий файрвол безопасным, т.е. защищающим нашу систему от проникновений извне. На самом деле нам придется настроить

файрвол таким образом, чтобы он не только защищал нас, но и защищал внешнюю сеть от нас. :) Защита такого рода является обязательной, она нужна для того, чтобы, в том случае, если наша система все-таки была взломана, взломщик не смог воспользоваться нашими ресурсами для дальнейшей атаки любого рода. Этот аспект является важнейшим в сетях класса SOHO, т.е. небольших офисах. Обыкновенно вирусы не оказывают никакого влияния на малые сети и заражение ими редко приводит к потере данных. Для нас, пользователей \*nix, этой проблемы практически не существует. В любом случае, т.к. небольшие сети, как правило, защищены гораздо хуже больших, кракеры стараются использовать их в качестве "опорной базы" для DoS атак, или другой своей вредоносной активности.

В последующем я опишу всю конфигурацию по кусочкам, дабы мы смогли бы проверить пошагово каждый фрагмент. Каждый шаг может потребовать от вас вставки чего-либо до, после или в середину указываемого скрипта. Все действия производятся так, чтобы (я надеюсь) ваша сеть не работала только лишь короткий промежуток времени во время настройки. Это сделано мною потому как я предположил что у многих из вас (таких как я) выделен сервер под Firewall. И так как я предпочитаю настраивать свой сервер через SSH, отключение сети может иметь пренеприятные последствия, такие как ползания под столами или хуже того. Если же вы смелы, то вы можете скопировать скрипт в конце этого документа и запустить его на своей машине. Этот скрипт должен быть работоспособен на все сто, но тестировал я его только на своей машине, так что уmmv.

#### Установка переменных окружения

Вы можете установить необходимые переменные окружения следующим скриптом:  
Code: '

```
#!/bin/sh
```

```
#
```

```
# Внешний интерфейс
```

```
EXTIF="ppp0"
```

```
# Внутренний интерфейс
```

```
INTIF="eth1"
```

```
# Loop device/localhost
```

```
LPDIF="lo"
```

```
LPDIP="127.0.0.1"
```

```
LPDMSK="255.0.0.0"
```

```
LPDNET="$LPDIP/$LPDMSK"
```

```
# Необходимые утилиты
```

```
IPT="/sbin/iptables"
```

```
IFC="/sbin/ifconfig"
```

```
G="/bin/grep"
SED="/bin/sed"
AWK="/usr/bin/awk"
ECHO="/bin/echo"
```

# Последующие команды могут работать некорректно при локализации.

# Установка переменных окружения внешнего интерфейса

```
EXTIP=""$IFC $EXTIF | $AWK /$EXTIF/{next}/{split($0,a,":");split(a[2],a," ");print a[1];exit}""
EXTBC="255.255.255.255"
#EXTMSK=""$IFC $EXTIF | $G Mask:|$SED 's/.*Mask:\([^\ ]*\)/\1/'""
EXTMSK=""$IFC $EXTIF | $AWK /$EXTIF/{next}/{split($0,a,":");split(a[4],a," ");print a[1];exit}""
EXTNET="$EXTIP/$EXTMSK"
$ECHO "EXTIP=$EXTIP EXTBC=$EXTBC EXTMSK=$EXTMSK EXTNET=$EXTNET"
```

# Due to absence of EXTBC I manually set it to 255.255.255.255

# this (hopefully) will serve the same purpose

# Установка переменных окружения внутреннего интерфейса

```
INTIP=""$IFC $INTIF | $AWK /$INTIF/{next}/{split($0,a,":");split(a[2],a," ");print a[1];exit}""
INTBC=""$IFC $INTIF | $AWK /$INTIF/{next}/{split($0,a,":");split(a[3],a," ");print a[1];exit}""
INTMSK=""$IFC $INTIF | $AWK /$INTIF/{next}/{split($0,a,":");split(a[4],a," ");print a[1];exit}""
INTNET="$INTIP/$INTMSK"
$ECHO "INTIP=$INTIP INTBC=$INTBC INTMSK=$INTMSK INTNET=$INTNET"
```

iptables ACCEPTS

Теперь мы должны установить АССЕРТЫ, так, чтобы мы могли соединяться с нашим сервером. На самом деле это очень большой вопрос. Правила для надежного маршрутизатора должны по умолчанию запрещать нежели разрешать. Однако, если вы сделаете это, то потеряете все соединения. Продолжайте тестирование пока не будете уверены что ваши АССЕРТЫ работают как надо. Однако думаю, что сперва мы введем следующее и это будет предпоследним правилом в окончательном скрипте. Code: '

```
$IPT -t nat -A PREROUTING -j ACCEPT
```

```
# $IPT -t nat -A POSTROUTING -o $EXTIF -s $INTNET -j SNAT --to $EXTIP
```

```
# Закомментируйте последующие строки (которые содержат "MASQUERADE")
# для сетей без трансляции адресов (NAT)
$IPT -t nat -A POSTROUTING -o $EXTIF -s $INTNET1 -j MASQUERADE
$IPT -t nat -A POSTROUTING -o $EXTIF -s $INTNET2 -j MASQUERADE
$IPT -t nat -A POSTROUTING -j ACCEPT
$IPT -t nat -A OUTPUT -j ACCEPT
$IPT -A INPUT -p tcp --dport auth --syn -m state --state NEW -j ACCEPT
$IPT -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

## iptables DROP & REJECT

Теперь мы определим пару цепочек (chains) которые будут фиксировать события DROP и REJECT. Таким образом нам не придется вводить отдельные строки для каждой введенной команды. Сообщения о событиях будут отправлены сервису syslog, (обычно они фиксируются в /var/log/messages). Позже я (переводчик не имеет к этому никакого отношения) собираюсь написать скрипт на sed/grep по разбору событий для облегченного просмотра и установлю его как ежедневную работу для сервиса cron.

Эти строки следует вставить сразу после текста выше, в тот же скрипт. Когда вы это сделаете, запустите скрипт снова. Это не окажет влияния на вашу сеть, вы пока просто устанавливаете правила. Но это поможет убедиться что мы не сделали ошибок на данном этапе. Code: '

```
# ***** Цепочки журналирования событий *****
#
# Теперь мы определяем несколько цепочек которые служат для записи
# событий о сбрасываемых пакетах. Это позволит нам избежать ввода
# команд для каждого правила. Сперва мы фиксируем DROP, а потом REJECT.
# Не жалуйтесь, если цепочки уже существуют (однако это не приведет к
# ошибкам???)
$IPT -N DROPI 2> /dev/null
$IPT -A DROPI -m limit --limit 3/minute --limit-burst 10 -j LOG --log-prefix 'FIREWALL
DROP BLOCKED:'
$IPT -A DROPI -j DROP
$IPT -N REJECTI 2> /dev/null
$IPT -A REJECTI -m limit --limit 3/minute --limit-burst 10 -j LOG --log-prefix 'FIREWALL
REJECT BLOCKED:'
$IPT -A REJECTI -j REJECT
$IPT -N DROP2 2> /dev/null
```

```
$IPT -A DROP2 -m limit --limit 3/second --limit-burst 10 -j LOG --log-prefix 'FIREWALL  
DROP UNKNOWN:'
```

```
$IPT -A DROP2 -j DROP
```

```
$IPT -N REJECT2 2> /dev/null
```

```
$IPT -A REJECT2 -m limit --limit 3/second --limit-burst 10 -j LOG --log-prefix 'FIREWALL  
REJECT UNKNOWN:'
```

```
$IPT -A REJECT2 -j REJECT
```

# Для тестирования фиксируем события ACCEPT

```
$IPT -N ACCEPT1 2> /dev/null
```

```
$IPT -A ACCEPT1 -m limit --limit 10/second --limit-burst 50 -j LOG --log-prefix 'FIREWALL  
ACCEPT:'
```

```
$IPT -A ACCEPT1 -j ACCEPT
```

### Сброс правил

Теперь, когда мы видим наши устройства, правильно определенные, вставляем команду сброса правил. Однако все уже назначенные правила будут сброшены. Эти строки должны быть вставлены после определения утилит, которые заканчиваются строкой: ECHO='/bin/echo' Code: '

# Сброс всех существующих и очистка персональных цепочек.

```
CHAINS=`cat /proc/net/ip_tables_names 2>/dev/null`
```

```
for i in $CHAINS
```

```
do
```

```
    $IPT -t $i -F
```

```
done
```

```
for i in $CHAINS
```

```
do
```

```
    $IPT -t $i -X
```

```
done
```

### Локальные интерфейсы

Теперь мы готовы для того, чтобы описать некоторые правила. Для начала мы разрешим все пакеты с loopback интерфейса, имеющие в качестве адреса назначения один из адресов наших интерфейсов. Code: '

```
$IPT -A INPUT -i $LPDIF -s $LPDIP -j ACCEPT
```

```
$IPT -A INPUT -i $LPDIF -s $EXTIP -j ACCEPT
```

```
$IPT -A INPUT -i $LPDIF -s $INTIP1 -j ACCEPT
$IPT -A INPUT -i $LPDIF -s $INTIP2 -j ACCEPT
```

#### Блокировка широковещательных пакетов

Теперь мы должны заблокировать все входящие и исходящие широковещательные пакеты. Это предотвратит DoS атаки против нас, и не позволит нашим клиентам проводить DoS атаки против кого либо другого. Если бы все системные администраторы следовали этим правилам, тогда много суровых и дорогих DoS атак не состоялись или были максимально ограниченными. Code: Блокировка широковещательных пакетов

```
$IPT -A INPUT -i $EXTIF -d $EXTBC -j DROP1
$IPT -A INPUT -i $INTIF1 -d $INTBC1 -j DROP1
$IPT -A INPUT -i $INTIF2 -d $INTBC2 -j DROP1
$IPT -A OUTPUT -o $EXTIF -d $EXTBC -j DROP1
$IPT -A OUTPUT -o $INTIF1 -d $INTBC1 -j DROP1
$IPT -A OUTPUT -o $INTIF2 -d $INTBC2 -j DROP1
$IPT -A FORWARD -o $EXTIF -d $EXTBC -j DROP1
$IPT -A FORWARD -o $INTIF1 -d $INTBC1 -j DROP1
$IPT -A FORWARD -o $INTIF2 -d $INTBC2 -j DROP1
```

Теперь проверим скрипт еще раз, чтобы убедиться в том, что мы не наделали синтаксических ошибок. Также отмечу, что мы используем определенные нами DROP1 цепочки (chains). Это означает, что отбрасываемые пакеты будут отмечены в журнале событий (log file).

#### Блокировка доступа в локальную сеть из глобальной

Теперь мы блокируем доступ из глобальной сети в нашу локальную сеть, если мы не хотим что бы интернет провайдер назначал IP адреса для нашей внутренней сети. Code: '

```
# Блокировать внешний доступ к локальной сети
# Это позволит остановить боевых кракеров от использования
# нашей сети как стартовой точки для других атак.
#
# Нижеприведенная строчка на человеческом языке будет выглядеть как
# "если входящий пакет, пришедший на наш внешний интерфейс,
# имеет адрес назначения, отличный от адреса нашего внешнего интерфейса,
# то этот пакет не будет пропущен."
$IPT -A INPUT -i $EXTIF -d ! $EXTIP -j DROP1
```

## Изолирование локальных сетей

Теперь мы предпримем некоторые действия для наших локальных сетей. Другими словами - все пакеты не относящиеся к локальным сетям должны быть заблокированы. Code: '

```
# Теперь мы должны заблокировать все пакеты не относящиеся к
# адресному пространству наших локальных сетей.
# Запомните, если вы подключите свой ноутбук к другому разъему,
# вам надо убедиться, что ваш сетевой адрес соответствует адресам этой сети.
#
# Первая локальная сеть
$IPT -A INPUT -i $INTIF1 -s ! $INTNET1 -j DROPI
$IPT -A OUTPUT -o $INTIF1 -d ! $INTNET1 -j DROPI
$IPT -A FORWARD -i $INTIF1 -s ! $INTNET1 -j DROPI
$IPT -A FORWARD -o $INTIF1 -d ! $INTNET1 -j DROPI
# Вторая локальная сеть
$IPT -A INPUT -i $INTIF2 -s ! $INTNET2 -j DROPI
$IPT -A OUTPUT -o $INTIF2 -d ! $INTNET2 -j DROPI
$IPT -A FORWARD -i $INTIF2 -s ! $INTNET2 -j DROPI
$IPT -A FORWARD -o $INTIF2 -d ! $INTNET2 -j DROPI
```

Дальше мы сделаем некоторые дополнительные проверки исходящих пакетов и остановим все icmp пакеты кроме ping. Code: '

```
# Дополнительная проверка
$IPT -A OUTPUT -o $EXTIF -s ! $EXTNET -j DROPI
# Блокируем исходящие ICMP (кроме PING)
$IPT -A OUTPUT -o $EXTIF -p icmp --icmp-type ! 8 -j DROPI
$IPT -A FORWARD -o $EXTIF -p icmp --icmp-type ! 8 -j DROPI
```

Замечательно. Двигаемся дальше и проверяем скрипт на ошибки.

## Ports

Предполагая что у нас все сработало мы заткнем еще несколько портов, доступ по которым может представлять для нас серьезную опасность:

```
# COMmon ports:
# 0 is tcpmux; SGI had vulnerability, 1 is common attack
# 13 is daytime
# 98 is Linuxconf
# 111 is sunrpc (portmap)
```

```

# 137:139, 445 is Microsoft
# SNMP: 161,2
# Squid flotilla: 3128, 8000, 8008, 8080
# 1214 is Morpheus or KaZaA
# 2049 is NFS
# 3049 is very virulent Linux Trojan, mistakable for NFS
# Common attacks: 1999, 4329, 6346
# Common Trojans 12345 65535
COMBLOCK="0:1 13 98 111 137:139 161:162 445 1214 1999 2049 3049 4329 6346
3128 8000 8008 8080 12345 65535"
# TCP ports:
# 98 is Linuxconf
# 512-515 is rexec, rlogin, rsh, printer(lpd)
# [very serious vulnerabilities; attacks continue daily]
# 1080 is Socks proxy server
# 6000 is X (NOTE X over SSH is secure and runs on TCP 22)
# Block 6112 (Sun's/HP's CDE)
TCPBLOCK="$COMBLOCK 98 512:515 1080 6000:6009 6112"

# UDP ports:
# 161:162 is SNMP
# 520=RIP, 9000 is Sangoma
# 517:518 are talk and ntalk (more annoying than anything)
UDPBLOCK="$COMBLOCK 161:162 520 123 517:518 1427 9000 9 6346 3128 8000
8008 8080 12345 65535"

После определения переменных окружения нам останется только пробежаться по
ним циклом:
echo -n "FW: Blocking attacks to TCP port"
for i in $TCPBLOCK;
do
echo -n "$i "
$IPT -A INPUT -p tcp --dport $i -j DROPI
$IPT -A OUTPUT -p tcp --dport $i -j DROPI
$IPT -A FORWARD -p tcp --dport $i -j DROPI
done
echo ""
echo -n "FW: Blocking attacks to UDP port "
for i in $UDPBLOCK;

```



```
do
echo -n "$i "
$IPT -A INPUT -p udp --dport $i -j DROP
$IPT -A OUTPUT -p udp --dport $i -j DROP
$IPT -A FORWARD -p udp --dport $i -j DROP
done
echo ""
```

Ну что ж, теперь каждый раз, когда мы запускаем скрипт, эти строчки просто добавляются к уже существующим... что создаёт небольшой бардак. По этой причине мы собираемся перепрыгнуть в начало скрипта... сразу после переменных окружения для sed и grep, но перед переменными для EXTIP и EXTBC - там мы добавляем цикл, который производит очистку. Так мы будем уверены, что работаем в чистой среде. Нам не приходилось задумываться об этом ранее, потому что мы не имели возможности оттестить скрипт без обрыва соединения либо закрытия файрвола. Этот скрипт сначала выставляет всю политику в DROP, после чего очищает и удаляет наши цепочки. Чтобы убедиться, что мы по-прежнему можем залогиниться по ssh на наш сервер после того, как скрипт перезапустится, мы добавим цепочку INPUT для ssh. Пока что поставим её в конец скрипта. Это делается для того чтобы избежать открытия дыры в момент установки новых правил, что является довольно распространённой ошибкой:

```
# Отказываем, потом принимаем: это уберёт дыру в момент,
# когда мы закрываем порты
$IPT -P INPUT DROP
$IPT -P OUTPUT DROP
$IPT -P FORWARD DROP
# Очищаем все существующие цепочки и стираем дополнительные
CHAINS=`cat /proc/net/ip_tables_names 2>/dev/null`
for i in $CHAINS;
do
$IPT -t $i -F
done
for i in $CHAINS;
do
$IPT -t $i -X
done
$IPT -A INPUT -i $INTIF1 -p tcp --dport 22 --syn -m state --state NEW -j ACCEPT
```

## Sysctl'ы

Сразу после этого мы запустим sysctl'ы для tcp\_syncookies, icmp\_echo\_ignore\_broadcasts, rp\_filter и accept\_source\_route. До этого момента многие правила, которые мы "проверяли", фактически не выполнялись. По сути, мы просто делали проверку синтаксических ошибок. Теперь наши правила будут

выполняться в полной мере:

```
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
# Проверка адреса источника
for f in /proc/sys/net/ipv4/conf/*/rp_filter;
do
    echo 1 > $f
done
# Отключаем роутинг источника IP и ICMP-редиректы
for f in /proc/sys/net/ipv4/conf/*/accept_source_route;
do
    echo 0 > $f
done
for f in /proc/sys/net/ipv4/conf/*/accept_redirects;
do
    echo 0 > $f
done
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Теперь мы собираемся добавить трекинг ftp-соединения, так что нам не доведётся наблюдать ошибки PASV при установке паков:

```
# Запускаем трекинг ftp-соединения
MODULES="ip_nat_ftp ip_conntrack_ftp"
for i in $MODULES;
do
    echo "Добавляем модуль $i"
    modprobe $i
done
```

### Базовая конфигурация NAT

А сейчас возвратимся к концу нашего скрипта, поскольку мы собираемся открыть сервисы для хостов, находящихся за файерволом. Я включил следующие сервисы: IRC, MSN, ICQ, and NFS, FTP, domain,time и некоторые другие. Самое главное здесь то, что эти сервисы могут использоваться ТОЛЬКО ЗА файерволом. Таким образом никто не сможет их использовать по ftp внутри Вашей локальной сетки:

```
IRC='ircd'
MSN=1863
ICQ=5190
NFS='sunrpc'
# Мы пользуемся sync!!
```

```

PORTAGE='rsync'
OpenPGP_HTTP_Keyserver=11371
# Все порты сервисов считываются из /etc/services
TCPSERV="domain ssh http https ftp ftp-data mail pop3 pop3s imap3 imaps imap2 time
$PORTAGE $IRC $MSN $ICQ $OpenPGP_HTTP_Keyserver"
UDPSERV="domain time"
echo -n "FW: Allowing inside systems to use service:"
for i in $TCPSERV;
do
    echo -n "$i "
    $IPT -A OUTPUT -o $EXTIF -p tcp -s $EXTIP --dport $i --syn -m state --state NEW -j
    ACCEPT
    $IPT -A FORWARD -i $INTIF1 -p tcp -s $INTNET1 --dport $i --syn -m state --state NEW -j
    ACCEPT
    $IPT -A FORWARD -i $INTIF2 -p tcp -s $INTNET2 --dport $i --syn -m state --state NEW -j
    ACCEPT
done
echo ""
echo -n "FW: Allowing inside systems to use service:"
for i in $UDPSERV;
do
    echo -n "$i "
    $IPT -A OUTPUT -o $EXTIF -p udp -s $EXTIP --dport $i -m state --state NEW -j
    ACCEPT
    $IPT -A FORWARD -i $INTIF1 -p udp -s $INTNET1 --dport $i -m state --state NEW -j
    ACCEPT
    $IPT -A FORWARD -i $INTIF2 -p udp -s $INTNET2 --dport $i -m state --state NEW -j
    ACCEPT
done
echo ""

```

Теперь сделаем так, чтобы фаервол позволил нам пинговать внешний мир. Для этого разрешим прохождение icmp-пакетов через внешний интерфейс:

```

# Разрешаем внешнее пингование
$IPT -A OUTPUT -o $EXTIF -p icmp -s $EXTIP --icmp-type 8 -m state --state NEW -j
ACCEPT
$IPT -A FORWARD -i $INTIF1 -p icmp -s $INTNET1 --icmp-type 8 -m state --state NEW -j
ACCEPT
$IPT -A FORWARD -i $INTIF2 -p icmp -s $INTNET2 --icmp-type 8 -m state --state NEW -j
ACCEPT

```

# Разрешим фаерволу пинговать наши внутренние сетки:

```
$IPT -A OUTPUT -o $INTIF1 -p icmp -s $INTNET1 --icmp-type 8 -m state --state NEW -j ACCEPT
```

```
$IPT -A OUTPUT -o $INTIF2 -p icmp -s $INTNET2 --icmp-type 8 -m state --state NEW -j ACCEPT
```

Теперь мы по умолчанию будем записывать в журнал все прочие запросы, но никак не будем на них реагировать. Все что нам надо было принять мы описали в начале наших правил. Так что заканчиваем правила блокировкой всего, что специально не разрешено:

# Заблокируем все, что осталось:

```
$IPT -A INPUT -j DROP
```

```
$IPT -A OUTPUT -j REJECT
```

```
$IPT -A FORWARD -j DROP
```

Итак, все сделано. я имею дружественные nmap и nessus для моих соединений с вышеописанным набором правил и ничего не препятствует использованию IRC, MSN, ICQ, и emerge sync.

The full script

А сейчас, полноценный скрипт во всей своей красе (Заодно я поместил форвардинг ssh в более подходящее для него место):

# Внешний интерфейс

```
EXTIF=ppp0
```

# Внутренний интерфейс

```
INTIF1=eth1
```

```
INTIF2=eth2
```

# Loop-устройство/localhost

```
LPDIF=lo
```

```
LPDIP=127.0.0.1
```

```
LPDMSK=255.0.0.0
```

```
LPDNET="$LPDIP/$LPDMSK"
```

# Переменные текстовых инструментов

```
IPT='/sbin/iptables'
```

```
IFC='/sbin/ifconfig'
```

```
G='/bin/grep'
```

```
SED='/bin/sed'
```

# Последнее (но немаловажное) - пользователи

```
JAMES=192.168.1.77
```

```
TERESA=192.168.2.77
```

# Deny вместо ассерт: предотвращает открытие "дыр"

# в то время, как мы закрываем порты и все такое

```
$IPT -P INPUT DROP
```

```

$IPT -P OUTPUT DROP
$IPT -P FORWARD DROP
# Сброс всех существующих цепочек и стирание персональных цепочек
CHAINS=`cat /proc/net/ip_tables_names 2>/dev/null`
for i in $CHAINS;
do
    $IPT -t $i -F
done
for i in $CHAINS;
do
    $IPT -t $i -X
done
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
# Проверка адреса источника
for f in /proc/sys/net/ipv4/conf/*/rp_filter;
do
    echo 1 > $f
done
# Запрет маршрутизации IP от источника и редиректов ICMP
for f in /proc/sys/net/ipv4/conf/*/accept_source_route;
do
    echo 0 > $f
done
for f in /proc/sys/net/ipv4/conf/*/accept_redirects;
do
    echo 0 > $f
done
echo 1 > /proc/sys/net/ipv4/ip_forward
# Установка переменных среды для внешнего интерфейса
EXTIP=""$IFC $EXTIF|$G addr:|$SED 's/. *addr:\([^ ]*\) .*^1/'""
#EXTBC=""$IFC $EXTIF|$G Bcast:|$SED 's/. *Bcast:\([^ ]*\) .*^1/'""
EXTBC="255.255.255.255"
EXTMSK=""$IFC $EXTIF|$G Mask:|$SED 's/. *Mask:\([^ ]*\)^1/'""
EXTNET="$EXTIP/$EXTMSK"
#echo "EXTIP=$EXTIP EXTBC=$EXTBC EXTMSK=$EXTMSK EXTNET=$EXTNET"
echo "EXTIP=$EXTIP EXTBC=$EXTBC EXTMSK=$EXTMSK EXTNET=$EXTNET"

```

```

# Так как EXTBC отсутствует, я устанавливаю ее вручную как it to 255.255.255.255
# Это (надеюсь) послужит тем же целям
# Установка переменных среды для первого внутреннего интерфейса
INTIP1=""$IFC $INTIF1|$G addr:|$SED 's/. *addr:\([^\ ]*\) .*^1/'""
INTBC1=""$IFC $INTIF1|$G Bcast:|$SED 's/. *Bcast:\([^\ ]*\) .*^1/'""
INTMSK1=""$IFC $INTIF1|$G Mask:|$SED 's/. *Mask:\([^\ ]*\)^1/'""
INTNET1="$INTIP1/$INTMSK1"
echo "INTIP1=$INTIP1 INTBC1=$INTBC1 INTMSK1=$INTMSK1 INTNET1=$INTNET1"
# Установка переменных среды для второго внутреннего интерфейса
INTIP2=""$IFC $INTIF2|$G addr:|$SED 's/. *addr:\([^\ ]*\) .*^1/'""
INTBC2=""$IFC $INTIF2|$G Bcast:|$SED 's/. *Bcast:\([^\ ]*\) .*^1/'""
INTMSK2=""$IFC $INTIF2|$G Mask:|$SED 's/. *Mask:\([^\ ]*\)^1/'""
INTNET2="$INTIP2/$INTMSK2"
echo "INTIP2=$INTIP2 INTBC2=$INTBC2 INTMSK2=$INTMSK2 INTNET2=$INTNET2"
# Сейчас мы собираемся создать несколько собственных цепочек, результатом
работы
# которых будет логгинг отброшенных пакетов. Это поможет нам избежать
необходимости
# вводить команду log перед каждым отбрасыванием пакета, что мы хотим
запротоколировать.
# Первыми идут лог отброшенных пакетов и собственно отброс, затем лог пакетов с
отказами
# и собственно отказы.
# Отключаем сообщения о том, что цепочки уже существуют (чтобы перезапуск был
без мусора)
$IPT -N DROPI 2> /dev/null
$IPT -A DROPI -j LOG --log-prefix 'DROPI:'
$IPT -A DROPI -j DROP
$IPT -N REJECTI 2> /dev/null
$IPT -A REJECTI -j LOG --log-prefix 'REJECTI:'
$IPT -A REJECTI -j REJECT
# Весь трафик от устройства loopback принимается
# если IP совпадает с любым из наших интерфейсов.
$IPT -A INPUT -i $LPDIF -s $LPDIP -j ACCEPT
$IPT -A INPUT -i $LPDIF -s $EXTIP -j ACCEPT
$IPT -A INPUT -i $LPDIF -s $INTIP1 -j ACCEPT
$IPT -A INPUT -i $LPDIF -s $INTIP2 -j ACCEPT
# Широковещательные пакеты блокируем
$IPT -A INPUT -i $EXTIF -d $EXTBC -j DROPI

```

```

$IPT -A INPUT -i $INTIF1 -d $INTBC1 -j DROPI
$IPT -A INPUT -i $INTIF2 -d $INTBC2 -j DROPI
$IPT -A OUTPUT -o $EXTIF -d $EXTBC -j DROPI
$IPT -A OUTPUT -o $INTIF1 -d $INTBC1 -j DROPI
$IPT -A OUTPUT -o $INTIF2 -d $INTBC2 -j DROPI
$IPT -A FORWARD -o $EXTIF -d $EXTBC -j DROPI
$IPT -A FORWARD -o $INTIF1 -d $INTBC1 -j DROPI
$IPT -A FORWARD -o $INTIF2 -d $INTBC2 -j DROPI
# Блокируем доступ к внутренней сети из WAN
# Это также призвано не дать нечестивым крякерам использовать нашу сетку
# в качестве отправной точки для атак на других людей
# Перевод с языка iptables:
# "если пришедшие на наружный интерфейс пакеты были отправлены не с
выданного
# nefarious адреса, выкинуть их как горячую картошку"
$IPT -A INPUT -i $EXTIF -d ! $EXTIP -j DROPI
# А сейчас мы блокируем внутренние адреса, кроме двух, присвоенных нашим двум
# внутренним интерфейсам.....только помните, что если вы воткнете свой лэптоп или
# какой другой pc в напрямую в одну из этих сетевых карт, то нужно удостовериться,
# что они имеют именно эти IP-адреса или добавить соответствующий адрес
отдельно.
# Первый интерфейс/первая внутренняя сеть
$IPT -A INPUT -i $INTIF1 -s ! $INTNET1 -j DROPI
$IPT -A OUTPUT -o $INTIF1 -d ! $INTNET1 -j DROPI
$IPT -A FORWARD -i $INTIF1 -s ! $INTNET1 -j DROPI
$IPT -A FORWARD -o $INTIF1 -d ! $INTNET1 -j DROPI
# Второй интерфейс/вторая внутренняя сеть
$IPT -A INPUT -i $INTIF2 -s ! $INTNET2 -j DROPI
$IPT -A OUTPUT -o $INTIF2 -d ! $INTNET2 -j DROPI
$IPT -A FORWARD -i $INTIF2 -s ! $INTNET2 -j DROPI
$IPT -A FORWARD -o $INTIF2 -d ! $INTNET2 -j DROPI
# Дополнительная Egress-проверка
$IPT -A OUTPUT -o $EXTIF -s ! $EXTNET -j DROPI
# Блокируем исходящие пакеты ICMP (за исключением PING)
$IPT -A OUTPUT -o $EXTIF -p icmp --icmp-type ! 8 -j DROPI
$IPT -A FORWARD -o $EXTIF -p icmp --icmp-type ! 8 -j DROPI
# печально известные порты:
# 0 - tcpmux; у SGI есть уязвимость, через которую можно атаковать

```

```

# 13 - daytime
# 98 - Linuxconf
# 111 - sunrpc (portmap)
# 137:139, 445 - Microsoft
# SNMP: 161,2
# Флотилия Squid: 3128, 8000, 8008, 8080
# 1214 - Morpheus или KaZaA
# 2049 - NFS
# 3049 - очень заразный троян для Linux, часто путаемый с NFS
# Часто атакуемые: 1999, 4329, 6346
# Частые трояны 12345 65535
COMBLOCK="0:1 13 98 111 137:139 161:162 445 1214 1999 2049 3049 4329 6346
3128 8000 8008 8080 12345 65535"
# Порты TCP:
# 98 - Linuxconf
# 512-515 - rexec, rlogin, rsh, printer(lpd)
# [очень серьезные уязвимости; продолжаются ежедневные атаки]
# 1080 - прокси-серверы Socks
# 6000 - X (ЗАМЕЧАНИЕ. X через SSH - безопасен, и работает на порту TCP 22)
# Блокировка 6112 (CDE у Sun и HP)
TCPBLOCK="$COMBLOCK 98 512:515 1080 6000:6009 6112"
# Порты UDP:
# 161:162 - SNMP
# 520=RIP, 9000 - Sangoma
# 517:518 - talk и ntalk (самые надоедливые)
UDPBLOCK="$COMBLOCK 161:162 520 123 517:518 1427 9000"
echo -n "FW: Blocking attacks to TCP port"
for i in $TCPBLOCK;
do
echo -n "$i "
$IPT -A INPUT -p tcp --dport $i -j DROPI
$IPT -A OUTPUT -p tcp --dport $i -j DROPI
$IPT -A FORWARD -p tcp --dport $i -j DROPI
done
echo ""
echo -n "FW: Blocking attacks to UDP port "
for i in $UDPBLOCK;
do

```



```

echo -n "$i "
$IPT -A INPUT -p udp --dport $i -j DROPI
$IPT -A OUTPUT -p udp --dport $i -j DROPI
$IPT -A FORWARD -p udp --dport $i -j DROPI
done
echo ""
# Открываем отслеживание соединений по ftp
MODULES="ip_nat_ftp ip_conntrack_ftp"
for i in $MODULES;
do
    echo "Inserting module $i"
    modprobe $i
done
# Защищаем некоторые распространенные клиенты для чата.
# Уберите из списка допустимых для пущей безопасности.
IRC='ircd'
MSN=1863
ICQ=5190
NFS='sunrpc'
# Нам нужно синхронизировать данные!!
PORTAGE='rsync'
OpenPGP_HTTP_Keyserver=11371
# Все порты сервисов читаются из /etc/services
TCPSERV="domain ssh http https ftp ftp-data mail pop3 pop3s imap3 imaps imap2 time
$PORTAGE $IRC $MSN $ICQ $OpenPGP_HTTP_Keyserver" UDPSERV="domain time"
echo -n "FW: Allowing inside systems to use service:"
for i in $TCPSERV;
do
    echo -n "$i "
    $IPT -A OUTPUT -o $EXTIF -p tcp -s $EXTIP --dport $i --syn -m state --state NEW -j
ACCEPT
    $IPT -A FORWARD -i $INTIF1 -p tcp -s $INTNET1 --dport $i --syn -m state --state NEW -
j ACCEPT
    $IPT -A FORWARD -i $INTIF2 -p tcp -s $INTNET2 --dport $i --syn -m state --state NEW -
j ACCEPT
done
echo ""
echo -n "FW: Allowing inside systems to use service:"
for i in $UDPSERV;

```

```

do
echo -n "$i "
$IPT -A OUTPUT -o $EXTIF -p udp -s $EXTIP --dport $i -m state --state NEW -j
ACCEPT
$IPT -A FORWARD -i $INTIF1 -p udp -s $INTNET1 --dport $i -m state --state NEW -j
ACCEPT
$IPT -A FORWARD -i $INTIF2 -p udp -s $INTNET2 --dport $i -m state --state NEW -j
ACCEPT
done
echo ""
# Разрешается ping наружу
$IPT -A OUTPUT -o $EXTIF -p icmp -s $EXTIP --icmp-type 8 -m state --state NEW -j
ACCEPT
$IPT -A FORWARD -i $INTIF1 -p icmp -s $INTNET1 --icmp-type 8 -m state --state NEW -j
ACCEPT
$IPT -A FORWARD -i $INTIF2 -p icmp -s $INTNET2 --icmp-type 8 -m state --state NEW -j
ACCEPT
# Файерволу разрешается ping внутренних систем
$IPT -A OUTPUT -o $INTIF1 -p icmp -s $INTNET1 --icmp-type 8 -m state --state NEW -j
ACCEPT
$IPT -A OUTPUT -o $INTIF2 -p icmp -s $INTNET2 --icmp-type 8 -m state --state NEW -j
ACCEPT
$IPT -A INPUT -i $INTIF1 -p tcp --dport 22 --syn -m state --state NEW -j ACCEPT
$IPT -t nat -A PREROUTING -j ACCEPT
$IPT -t nat -A POSTROUTING -o $EXTIF -s $INTNET1 -j MASQUERADE
$IPT -t nat -A POSTROUTING -o $EXTIF -s $INTNET2 -j MASQUERADE
$IPT -t nat -A POSTROUTING -j ACCEPT
$IPT -t nat -A OUTPUT -j ACCEPT
$IPT -A INPUT -p tcp --dport auth --syn -m state --state NEW -j ACCEPT
$IPT -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
# Заблокировать и запротоколировать все, что мы могли забыть.
$IPT -A INPUT -j DROP
$IPT -A OUTPUT -j REJECT
$IPT -A FORWARD -j DROP

```

## Установка почтовой системы

### Установка почтовой системы

Установка почтовой системы на Gentoo linux для небольшой конторы

Используется postfix + cyrus-imap, авторизация пользователей через cyrus-sasl без прикручивания какой-либо БД (подходит для контор с несколькими десятками почтовых ящиков - хранить что-либо в mysql или postgres в данном случае нет смысла). Пользователи хранятся в базе sasl\_db, поэтому нет нужды заводить в системе реальных пользователей.

Возможно при установке указанных пакетов понадобится установить все зависимости, которые они за собой тянут. Предоставим системе самой разобраться, что же ей не хватает - она с этим вполне успешно справляется.

Предпочитаю использовать для /var reiserfs, но это сугубо личные предпочтения.

```
emerge -pv cyrus-sasl
```

```
[ebuild N ] dev-libs/cyrus-sasl-2.1.20 -authdaemond +berkdb -debug +gdbm +java  
-kerberos -ldap +mysql +pam -postgres +ssl -static 0 kB
```

На всякий случай проверяем флаги - все ОК

```
emerge cyrus-sasl
```

Этот пакет поставили, едем дальше

```
emerge -pv postfix
```

These are the packages that I would merge, in order:

Calculating dependencies ...done!

```
[ebuild N ] mail-mta/postfix-2.1.5-r1 +ipv6 -ldap -mailwrapper -mbox +mysql +pam  
-postgres -sasl*(-selinux) +ssl -vda 0 kB
```

Здесь нам может пригодиться флаг sasl, хотя в моем случае пока нет нужды авторизоваться на smtp, возможно в дальнейшем здесь будет описана авторизация через cyrus-sasl на smtp - вообще-то выставляйте флаги по вкусу кому что надо

```
USE="sasl" emerge postfix
```

Постфикс собрался

```
emerge -pv cyrus-imapd
```

These are the packages that I would merge, in order:

Calculating dependencies ...done!

```
[ebuild N ] net-mail/cyrus-imapd-2.2.10 -afs -drac -idled -kerberos +pam -snmp +ssl  
+tcpd 0 kB
```

После проверки и установки нужных нам флагов пересобираем openssl и imap-сервер

```
emerge cyrus-imapd
```

Программа для администрирования имаповских ящиков

```
emerge cyrus-imap-admin
```

После того как все нужные нам пакеты поставлены, можно приступить к настройке

сначала cyrus-sasl

```
passwd cyrus
```

```
pwconv
```

```
chown -R cyrus:mail /etc/sasl2 - доступ cyrus к базе /etc/sasl2/sasldb2
```

```
saslpasswd2 cyrus - заводим в sasldb2
```

```
sasldblistusers2 - для проверки
```

Разбираемся с postfix

Содержимое /etc/postfix/main.cf, ниже приведена примерная конфигурация почтового сервера подключенного напрямую к инету (не через relayhost), естественно для этого случая должна быть MX-запись в ДНС. Подчеркиваю, что умышленно опускаю многие параметры в main.cf, чтобы не раздувать описание. Добавьте все остальное руководствуясь документацией postfix.

```
queue_directory = /var/spool/postfix
```

```
command_directory = /usr/sbin
```

```
daemon_directory = /usr/lib/postfix
```

```
mail_owner = postfix
```

```
myhostname = mail.domain.tld
```

```
mydomain = mail.domain.tld
```

```
myorigin = $mydomain
```

```
inet_interfaces = all
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
local_recipient_maps =
unknown_local_recipient_reject_code = 550
mynetworks_style = subnet
mynetworks = 192.168.1.0/24, 127.0.0.0/8
relay_domains = $mydestination
НА ЭТУ СТРОЧКУ ОБРАТИТЕ ВНИМАНИЕ!!!
mailbox_transport = lmtp:unix:/var/imap/socket/lmtp
Если надо подключить procmail то вместо нее пишем
mailbox_transport = procmail
```

Когда добавили все, что нужно в main.cf, идем в /etc/postfix/master.cf

Ищем такую строку

```
# Also specify in main.cf: cyrus_destination_recipient_limit=1
cyrus    unix    -    n    n    -    -    pipe
user=cyrus argv=/cyrus/bin/deliver -e -r ${sender} -m ${extension} ${user}
```

И заменяем путь на

```
# Also specify in main.cf: cyrus_destination_recipient_limit=1
cyrus    unix    -    n    n    -    -    pipe
user=cyrus argv=/cyrus/deliver -e -r ${sender} -m ${extension} ${user}
```

При использовании procmail, добавляем

```
procmail unix    -    n    n    -    -    pipe
flags=R user=cyrus argv=/usr/bin/procmail -p /etc/procmailrc USER=${user}
```

Где файл /etc/procmailrc :

```
DELIVERMAIL=/usr/lib/cyrus/deliver
LOGFILE=/var/log/procmaillog
IMAP="$DELIVERMAIL -e -a $USER -m user.$USER"
(если надо подключить spamassassin)
:0fw : spamassassin.lock
```

```
* < 90000
| /usr/bin/spamassassin
и далее обязательно
:0
| $IMAP
:0w
{
EXITCODE=$?
HOST
}
```

Все, с постфиксом разобрались, теперь cyrus-imapd

Редактируем /etc/cyrus.conf

```
# $Header: /var/cvsroot/gentoo-x86/net-mail/cyrus-imapd/files/cyrus.conf,v 1.4 2004/07/18
04:02:23 dragonheart Exp $
# Standard standalone server configuration.
START {
# Do not delete this entry!
recover      cmd="ctl_cyrusdb -r"
# This is only necessary if using idled for IMAP IDLE.
#idled       cmd="idled"
}
# UNIX sockets start with a slash and are put into /var/imap/socket.
SERVICES {
# Add or remove based on preferences.
imap         cmd="imapd" listen="imap2" prefork=0
pop3         cmd="pop3d" listen="pop-3" prefork=0
# Don't forget to generate the needed keys for SSL or TLS
# (see doc/html/install-configure.html).
imaps        cmd="imapd -s" listen="imaps" prefork=0
pop3s        cmd="pop3d -s" listen="pop3s" prefork=0
sieve        cmd="timsieved" listen="sieve" prefork=0
# at least one LMTP is required for delivery
#lmtp        cmd="lmtpd" listen="lmtp" prefork=0
##ОБРАТИТЕ ВНИМАНИЕ НА ЭТУ СТРОКУ
lmtpunix     cmd="lmtpd" listen="/var/imap/socket/lmtp" prefork=0
# this is only necessary if using notifications
```

```

#notify      cmd="notifyd" listen="/var/imap/socket/notify" proto="udp" prefork=1
}
EVENTS {
# This is required.
checkpoint  cmd="ctl_cyrusdb -c" period=30
# This is only necessary if using duplicate delivery suppression.
delprune    cmd="ctl_deliver -E 3" period=1440
# This is only necessary if caching TLS sessions.
tlsprune    cmd="tls_prune" period=1440
}

```

Вот так делаем сертификаты

```

openssl req -new -nodes -out req.pem -keyout key.pem
openssl rsa -in key.pem -out new.key.pem
openssl x509 -in req.pem -out ca-cert -req \
-signkey new.key.pem -days 999
cp new.key.pem /etc/ssl/cyrus/server.pem
rm new.key.pem
cat ca-cert >> /etc/ssl/cyrus/server.pem
chown cyrus:mail /etc/ssl/cyrus/server.pem
chmod 600 /etc/ssl/cyrus/server.pem # Your key should be protected

```

Проверяем все-ли директории созданы

```

/var/imap
cd /var
mkdir imap
chown cyrus:mail imap
chmod 750 imap
/var/spool/imap
cd /var/spool
mkdir imap
chown cyrus:mail imap
chmod 750 imap
/usr/sieve
cd /usr
mkdir sieve
chown cyrus:mail sieve

```

chmod 750 sieve

Далее редактируем /etc/imapd.conf

```
configdirectory:    /var/imap
partition-default:  /var/spool/imap
auto_transition:    yes
tls_ca_path:        /etc/ssl/cyrus
tls_cert_file:      /etc/ssl/cyrus/server.pem
tls_key_file:       /etc/ssl/cyrus/server.pem
admins:             cyrus
hashimapspool:      yes
allowanonymouslogin: no
allowplaintext:     yes
sasl_pwcheck_method: auxprop
sasl_auxprop_plugin: sasldb
sasldb_path:        /etc/sasl2/sasldb2
sasl_mech_list:     LOGIN PLAIN
```

Авторизация через базу sasldb, механизмы авторизации LOGIN,PLAIN Если мы будем использовать обычные открытые пароли, без шифрования (например внутри организации) то изменим в /etc/imapd.conf

```
allowanonymouslogin: yes
```

вроде бы с настройкой cyrus покончено

Теперь займемся почтовыми ящиками юзеров, для этого используем программу cyradm

```
/etc/init.d/cyrus start
```

```
cyradm -user cyrus -server localhost
```

```
localhost> cm user.testuser
```

```
localhost> help - выводит все команды для руления юзерскими ящиками
```

```
localhost> exit
```

Не забудьте забить в sasldb2 нового юзера:

```
saslpaswd2 testuser
```

Поднимаем сервисы



```
/etc/init.d/postfix start
```

```
/etc/init.d/cyrus start
```

Настраиваем почтового клиента (проверено с thunderbird 1.0 и с KMail 1.7.1, с Outlook Express тоже не должно быть проблем).

Пробуем отправить письмо, если что-то не работает - смотрим логи, прежде всего проверьте права на доступ в /var/imap/socket/lmtp для пользователей postfix и cyrus, чаще всего проблемы связаны именно с этим. Также проверьте права на доступ к /etc/sasl2/saslauth2 для пользователя cyrus, который должен быть в группе mail.

Желательно поставить антивирус например clamav используем для этого связку clamav + clamsmtp хотя возможны и другие варианты например amavisd-new. Установка прекрасно описана здесь [http://www.nixp.ru/articles/clamav\\_postfix](http://www.nixp.ru/articles/clamav_postfix), необходимо руководствоваться ей, естественно с адаптацией под реалии Gentoo

```
emerge clamav
```

```
ACCEPT_KEYWORDS="" emerge clamsmtp (пока замаскирован в портаджах)
```

Затем в файле /etc/conf.d/clamd START\_CLAMD = yes (для версии старше 0.85 уже не актуально - все настройки только в clamd.conf и clamsmtpd.conf)

Редактируем файлы /etc/clamd.conf и /etc/clamsmtpd.conf в соответствии с рекомендациями приведенными выше по ссылке Обратите внимание на параметры LocalSocket: /var/run/clamav/clamd.sock в /etc/clamav.conf и ClamAddress: /var/run/clamav/clamd.sock в /etc/clamsmtpd.conf - путь и имя файла должен одинаковым для обоих конфигов

В файл main.cf необходимо добавить две строчки:

```
content_filter = scan:127.0.0.1:10025
```

```
receive_override_options = no_address_mappings
```

Первая говорит postfix'у о том, что необходимо пересылать всю почту через сервис (фильтр) 'scan' на 10025-ый порт, который, как раз, открыт clamsmtpd. Вторая строчка говорит о том, чтобы postfix не делал никаких манипуляций с адресами до того, как они дойдут до content\_filter. Так что получается, что фильтр работает с реальными почтовыми адресами, а не с результатами перевода в виртуальные псевдонимы, маскардингом и т.п.

В файл master.cf необходимо добавить следующие строки:

```
# AV scan filter (used by content_filter)
```

```
scan    unix    -        -        n        -        16    smtp
```

```
        -o smtp_send_xforward_command=yes
```

# For injecting mail back into postfix from the filter

```
127.0.0.1:10026 inet n - n - 16 smtpd
```

```
-o content_filter=
```

```
-o receive_override_options=no_unknown_recipient_checks,no_header_body_checks
```

```
-o smtpd_helo_restrictions=
```

```
-o smtpd_client_restrictions=
```

```
-o smtpd_sender_restrictions=
```

```
-o smtpd_recipient_restrictions=permit_mynetworks,reject
```

```
-o mynetworks_style=host
```

```
-o smtpd_authorized_xforward_hosts=127.0.0.0/8
```

Примечание: вокруг знака '=' пробелы не ставить. Значение 127.0.0.1:10026 открывает 10026-порт для возвращения почты обратно от clamsmtpd.

```
rc-update add clamd default
```

```
rc-update add clamsmtpd default
```

```
/etc/init.d/postfix reload
```

все - антивирус должен проверять всю почту проходящую через Ваш почтовик

P.S. Все вышеописанное проверялось на моей рабочей машине и на серваке небольшой конторы. Автор (или авторы) не несут ответственности за безграмотное и необдуманное применение данного руководства. Ваша безопасность в ваших руках и в голове! Данное описание не претендует на полное и развернутое и не является единственно возможным вариантом конфигурации. Также выражаю признательность разработчикам вышеописанных замечательных программ за прекрасное исполнение и хорошую документацию. Практически все вышенаписанное подчерпнуто из документации к самим пакетам, а также некоторые мысли родились из отрывочных данных, блуждающих в рунете - большое спасибо за это их авторам.

PS. Некоторая путаница возникает с переадресацией почты - самый простой вариант использование файла aliases - но до его использования в отличие от sendmail, ящик который должен быть переадресован должен быть все-таки создан в системе через cyradm -user cyrus -auth login -server localhost команда см и не забыть что etc/postfix/main.cf

```
alias_maps = hash:/usr/local/etc/postfix/aliases
```

```
alias_database = hash:/usr/local/etc/postfix/aliases
```

# Настройка vsftpd

## Введение

Это перевод статьи en:HOWTO vsftpd. Кое-что изменено и добавлено от себя для более понятного, как мне кажется, изложения

Взято из en:HOWTO Plan, setup and run a high school Gentoo Club и модифицировано.

Это руководство описывает процесс установки VSFTP и настройки его для анонимного доступа с правами только для чтения. FTP (File Transfer Protocol) - старый, но надежный протокол, используемый для быстрой передачи файлов в сети.

## Установка VSFTPD

Зайдите в систему под пользователем root и введите команду: Code: Установка  
emerge vsftpd

## Настройка

Это очень легко. Откройте файл /etc/vsftpd/vsftpd.conf вашим любимым текстовым редактором и внесите в него вот эти изменения:

Общие настройки Файл: /etc/vsftpd/vsftpd.conf

```
dirmessage_enable=YES
# banner_file=/etc/vsftpd/vsftpd.banner # edit banner first
chown_uploads=NO
xferlog_enable=YES
idle_session_timeout=600
data_connection_timeout=120
ascii_upload_enable=NO
ascii_download_enable=NO
chroot_list_enable=YES
background=YES
listen=YES
ls_recurse_enable=NO
```

Анонимный пользователь (Anonymous), только для чтения  
Файл: /etc/vsftpd/vsftpd.conf  
anonymous\_enable=YES  
anon\_upload\_enable=NO  
anon\_mkdir\_write\_enable=NO

Отключаем локальных пользователей

Это плохая идея разрешать локальным пользователям доступ через ftp (проще sftp/ssh), потому мы отключаем им доступ. Файл: /etc/vsftpd/vsftpd.conf  
local\_enable=NO  
write\_enable=NO

Запуск службы

Gentoo предоставляет централизованное место для служб, которые называются init scripts или скрипты инициализации. Эти скрипты находятся в каталоге /etc/init.d/ и имеют имя сервиса, который они стартуют. Инит скрипты используются для управления сервисами. Для запуска VSFTPD сервера наберите Code: Запуск vsftpd как службы

```
/etc/init.d/vsftpd start
```

Вы можете узнать больше о том что вы можете сделать этим скриптом инициализации набрав "/etc/init.d/vsftpd" без аргументов ("start" это аргумент, еще есть "stop" и "restart") )

Настройка скриптов конфигурации

Вы не хотите набирать /etc/init.d/vsftpd start каждый раз когда вы включаете свой компьютер? Gentoo предоставляет простой путь запуска скриптов инициализации при включении вашего компьютера. Чтобы настроить VSFTPD для запуска при каждой загрузке, наберите: Code: запуск vsftpd как службы загрузке

```
rc-update add vsftpd default
```

Если вы хотите узнать больше о rc-update наберите "man rc-update".

Также можно и через chkconfig: Code: запуск vsftpd как службы загрузки  
chkconfig vsftpd on

Где размещать файлы

Файлы анонимного пользователя находятся в домашнем каталоге специального пользователя ftp (пользователя в системе). По умолчанию это - /home/ftp. Если вы хотите сменить его например на /var/ftp, вам нужно сделать следующие действия: /var/ftp Code: Смена размещения

```
rmdir /home/ftp  
mkdir /var/ftp  
chown ftp:ftp /var/ftp  
ln -s /var/ftp /home/
```

Теперь можно что-то разместить в этом каталоге. Например, если вы хотите сделать доступными для других каталоги disfiles и packages, можно сделать следующие действия: Code: '

```
mv /usr/portage/disfiles /var/ftp/  
ln -s /var/ftp/disfiles /usr/portage/  
mv /usr/portage/packages /var/ftp/  
ln -s /var/ftp/packages /usr/portage/
```

Примечание: VSFTPD автоматически запускается в chroot окружении, поэтому символические ссылки не работают. Именно для этого пришлось перенести каталоги /usr/portage/disfiles и /usr/portage/packages и создать символические ссылки в тех местах где они должны находиться

Но можно например монтировать нужные каталоги в .../ftp при помощи fstab. Для этого нужно создать каталоги: /var/ftp/distfiles и /var/ftp/packages Code: '

```
mkdir /var/ftp/distfiles  
mkdir /var/ftp/packages
```

После чего пропишите в /etc/fstab следующее: Code: '

```
/usr/portage/distfiles /var/ftp/distfiles none ro,bind 0 0  
/usr/portage/packages /var/ftp/packages none ro,bind 0 0
```

## Клиенты

Есть множество программ, которые позволяют получать доступ к ftp серверу. Вот некоторые из них:

ftp

lftp - лучший консольный клиент

mozilla-firefox

nautilus

gftp

## Jabber Server

### Введение

Jabber - это открытый протокол, использующий XML, для быстрого обмена сообщениями и информацией о присутствии между любыми двумя абонентами в интернете. Благодаря своей уникальной расширяемости и гибкости, jabber способен поддерживать множество протоколов - ICQ, IRQ, MSN, RSS, Yahoo и др.

### Установка пакетов

В качестве платформы для развертывания протокола jabber возьмем пакет jabberd. Он поддерживает множество возможностей - основные IM-протоколы ICQ, MSN, Yahoo, поддержку SSL-соединений и даже IPv6.

Сначала посмотрим, какие USE-флаги требует jabber: Code: Зависимости

```
# emerge -pv jabberd
```

These are the packages that I would merge, in order:

Calculating dependencies ...done!

```
[ebuild N ] net-im/jabberd-1.4.3-r5 -icq -ipv6 -ldap -msn -oscar -ssl -yahoo 0 kB
```

Total size of downloads: 0 kB

Включаем соответствующие флаги в установку (неважно, через переменную USE или через файл make.conf) и собираем: USE="icq ssl" emerge jabberd В качестве зависимости вытянется пакет jit - Jabber ICQ Transport, необходимый для поддержки протокола ICQ.

## Конфигурация

Сначала добавьте пользователей, ответственных за администрирование сервера jabber, в группу jabber: `grasswd -a имя_пользователя jabber`

Все конфигурационные файлы, связанные с jabber, имеют формат XML и находятся в каталоге `/etc/jabber`.

Обязательно надо прописать имя хоста и РАЗРЕШИТЬ регистрации.

## Запуск демона

Добавляем запуск сервера при старте системы: `rc-update add jabber default`

Стартуем наш сервер: `/etc/init.d/jabber start`

Все должно работать :)

## Выделенный сервер Counter-Strike

Рано или поздно у большинства админов локальных сетей возникает необходимость настраивать игровые сервера для пользователей. Одной из самых популярных сетевых игр на сегодняшний день является Counter-Strike.CS:Source. Которая, к сожалению пока не приобрела достаточной популярности у нас в России. Мы рассмотрим установку сервера для игры CS 1.6. Более популярной в среде российских геймеров.

## Необходимое программное обеспечение

Для установки CS-сервера нам понадобится только одна программа: `halflife-steam`  
Code: `emerge -pv halflife-steam`

Calculating dependencies ...done!

[ebuild N ] games-server/halflife-steam-2.0 3,431 kB

Total size of downloads: 3,431 kB

Steam - это утилита от Valve, позволяющая обновлять любую их игру через steam-network.

Установка и настройка

Ставим steam

Установка сводится к простому emerge halflife-steam.

Настройка установка hl

Steam установится в каталог /opt/halflife

ВАЖНО: в ноябре 2006 года в пакете была бага. Что характерно для всех продуктов получаемых через стим, они иногда забывают unix специфику. Непосредственно у меня, на steam не стоял бит исполняемости. И в cs было имя типа xxx.WAD (нужно wad, различается регистр) Code: команды

```
cd /opt/halflife
```

```
chmod +x steam
```

Code: команды

```
./steam -command update -game cstrike -dir ./
```

Наблюдаем лог. Если никаких слов типа error нет - значит сервер установился. Code: output

HLDS installation up to date

Предупреждение: Этот сервер работает только в LAN (класс C) и только со steam клиентами. Все остальное Вarez



## Файл сервер Samba в домене Win2k

### Задача

Организация файлового сервера

Авторизация пользователей через домен контроллер

### Дано

Red Hat 9.0

Samba 3.0.13

DC win 2003 server

### Вступление

Взяться за написание данной статьи меня побудило то, что когда возникла задача о которой написано выше , то статьи описывающей решение данной проблемы полностью я не нашел вот и решил написать такую статью в которой было бы готовое решение.

### Основная часть

Первым делом устанавливаем самбу. Решено было использовать последнюю версию 3.0.13. Установлено все это было из стандартного RPM пакета. Думаю с этим сложностей ни у кого не возникает, док по данному вопросу полно и потому начнем сразу править конфиг самбы. Ниже приведен окончательный работающий конфиг, может там и есть что лишнее, но после того как все заработало я убирать оттуда ничего не стал.

```
[global]
```

```
realm = bryusov.iasnet.ru
```

```
# Workgroup = имя NT-домена (или рабочей группы):
```

```
workgroup = DOMAIN
```

```
# NetBIOS-имя, под которым будет виден сервер остальным машинам сети.
```

```
netbios name = NAU
```

```
# Комментарий, появляющийся рядом с именем машины в "Сетевом окружении" Windows.
```

```
server string = Samba Server
```

```
# Следующий параметр влияет на безопасность. Hosts allow разрешает машинам с
```

```
# указанными IP-адресами присоединяться к Samba-серверу.
```

```
hosts allow = 172.18. 172.17. 127.
```

```

# если подставить %m то для каждой машины подключенной к Samba-серверу
будет
# использоваться свой log-файл.
log file = /var/log/samba/log.smbd

# это кому скока не жалко
max log size = 500

# определяет, каким образом будет осуществляться проверка пароля (нам надо
через DC)
security = domain

# Параметр Password server используется только совместно с опцией security =
domain
password server = <IP домен контролера>

# для репликации всех доменов входящих в траст с вашим доменом
allow trusted domains = yes

# включаем поддержку шифрованных паролей.
encrypt passwords = yes

# Используя следующий параметр можно создать отдельную конфигурацию для
каждой машины домена.
# Вместо пары символов %m при входе подставляется NetBIOS-имя машины.
# Я Такого не делал хотя поэкспериментировать можно.
# include = /usr/local/samba/lib/smb.conf.%m

# данные строчки можно не включать в работающий конфиг они определяют место
# хранения , порядок обновления Unix паролей и какой программой все это
производится
smb passwd file = /etc/samba/smbpasswd
unix password sync = Yes
passwd program = /usr/bin/passwd %u
passwd chat = *New*UNIX*password* %n\n *ReType*new*UNIX*password* %n\n
*passwd:*all*authentication*tokens*updated*successfully*

# В документации говорится, что с помощью этого параметра

```

# можно повысить производительность Samba-сервера.

```
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
```

# по дефолту слушаются все интерфейсы, здесь можно указать конкретно

```
interfaces = <ip или название интерфейса>
```

# строки которых не было в стандартном конфиге и были добавлены руками для репликации NT-юзеров:

```
winbind uid = 10000-20000
```

```
winbind gid = 10000-20000
```

```
winbind enum groups = yes
```

```
winbind enum users = yes
```

# Если вам нужны русские буквы в именах файлов, то раскомментируйте

# следующие 3 строки (заменяв KOI8-R на свою локаль)

```
# dos charset = CP866
```

```
# unix charset = KOI8-R
```

```
# display charset = KOI8-R
```

# описываем шары

```
[FILES]
```

```
comment = share
```

```
path = /share/FILES
```

```
public = no
```

```
writable = yes
```

```
valid users = DOMAIN\users
```

```
create mask = 0744
```

# настройка кириллицы по желанию

Все с конфигом самбы закончили, далее привожу конфиг /etc/krb5.conf

```
[logging]
```

```
default = FILE:/var/log/krb5libs.log
```

```
kdc = FILE:/var/log/krb5kdc.log
```

```
admin_server = FILE:/var/log/kadmind.log
```

[libdefaults]

```
ticket_lifetime = 24000
default_realm = PDC.DOMAIN.NAME.RU
dns_lookup_realm = false
dns_lookup_kdc = false
```

[realms]

```
DOMAIN.NAME.RU = {
    kdc = pdc.domain.name.ru:88
    admin_server = kerberos.domain.name.ru:749
    default_domain = domain.name.ru
}
```

[domain\_realm]

```
.domain.name.ru = DOMAIN.NAME.RU
domain.name.ru = DOMAIN.NAME.RU
```

[kdc]

```
profile = /var/kerberos/krb5kdc/kdc.conf
```

[appdefaults]

```
pam = {
    debug = false
    ticket_lifetime = 36000
    renew_lifetime = 36000
    forwardable = true
    krb4_convert = false
}
```

Все остальное оставлено без изменений.

Теперь заводим самба сервер в домен, делаем со стороны домен контролера двусторонние доверительные отношения, запускаем getent group и видим что все работает.

Вот и все надеюсь полезной инфы достаточно для быстрой и эффективной настройки файл сервера.

Огромное спасибо всем кто писал статьи про самбу почти все их прочитал и почерпнул много полезного , в том числе не только для решения своих задач.

## **Руководство по настройке Samba в режиме PDC с использованием LDAP**

Устанавливаемые пакеты

Установим нужные пакеты:

```
# USE="ldap acl ldapsam pam gdbm samba ssl tcpd winbind" emerge samba openldap  
acl nss_ldap pam_ldap
```

ВАЖНО: Версия nss\_ldap должна быть не ниже 250-r1 (>=sys-auth/nss\_ldap-250-r1)

OpenLDAP

Конфигурирование OpenLDAP

Для примера используется домен amber.global.com который является дочерним доменом домена global.com, который работает под управлением Win2003 Server.

Примечание: Не решенной задачей осталось создание доверительных отношений между доменами и включение домена amber в лес global.com

Файл: /etc/openldap/slapd.conf

```
include /etc/openldap/schema/core.schema  
include /etc/openldap/schema/cosine.schema  
include /etc/openldap/schema/inetorgperson.schema  
include /etc/openldap/schema/misc.schema  
include /etc/openldap/schema/nis.schema  
include /etc/openldap/schema/openldap.schema  
include /etc/openldap/schema/samba.schema  
pidfile /var/run/openldap/slapd.pid
```

argsfile                /var/run/openldap/slapd.args

access to dn.base=""

by self write

by \* auth

access to attr=userPassword

by self write

by \* auth

access to attr=shadowLastChange

by self write

by \* read

access to \*

by \* read

by anonymous auth

#loglevel 1

database        ldbm

suffix        "dc=amber,dc=global,dc=com"

rootdn        "cn=Manager,dc=amber,dc=global,dc=com"

# Пароль rootpw лучше всего указывать в зашифрованном виде.

# Для генерации шифрованного пароля используйте утилиту slappasswd

# Например: slappasswd -h {MD5}

# rootpw = secret

rootpw        {MD5}Xr4ilOzQ4PCOq3aQ0qbuaQ==

directory     /var/lib/openldap-ldbm

index objectClass        eq

index cn                eq,subinitial

index sn                eq,subinitial

index uid                eq,subinitial

index displayName        eq,subinitial

index uidNumber        eq

index gidNumber        eq

index memberUID        eq

index sambaSID        eq

index sambaPrimaryGroupSID eq

index sambaDomainName    eq

Файл: /etc/openldap/ldap.conf

HOST 127.0.0.1

BASE dc=sanaa,dc=global,dc=com

Проверяем разрешения на каталоги /var/lib/openldap-\*

ls -la /var/lib/

ВАЖНО:

```
drwx----- 2 ldap  ldap   104 Июн  8 18:31 openldap-data
drwx----- 2 ldap  ldap    72 Июн  8 18:31 openldap-ldbm
drwx----- 2 ldap  ldap    72 Июн  8 18:31 openldap-slurp
```

Запуск сервера OpenLDAP

пока работаем без шифрования трафика, так как сервер LDAP и SAMBA работают на одном сервере, подправляем конфиг чтобы LDAP слушал 389 порт только на localhost. Файл: /etc/conf.d/sldap

# conf.d file for the openldap-2.1 series

#

# To enable both the standard unciphpered server and the ssl encrypted

# one uncomment this line or set any other server starting options

# you may desire.

#

# OPTS="-h 'ldaps:// ldap:// ldapi://%2fvar%2frun%2fopenldap%2fslapd.sock'"

OPTS="-h 'ldap://127.0.0.1'"

и запускаем OpenLDAP

/etc/init.d/slapd start

SAMBA

Миграция OpenLDAP

Существует 2 пути создания учетных записей пользователей: домен уже существует и создание нового домена. В первом случае самбу нужно настроить в режиме BDC, перенести все учетные записи, и потом изолировав PDC, перезапустить самбу в

режиме PDC. Во втором случае все еще проще, самбу сразу запускаем в режиме PDC и создаем стандартные учетные записи при помощи замечательного пакета smbldap-tools.

emerge smbldap-tools

/etc/init.d/samba start

Code: configure.pl

-----

smbldap-tools script configuration

-----

Before starting, check

- . if your samba controller is up and running.
- . if the domain SID is defined (you can get it with the 'net getlocalsid')
  
- . you can leave the configuration using the Ctrl-c key combination
- . empty value can be set with the "." character

-----

Looking for configuration files...

Samba Configuration File Path [/etc/samba/smb.conf] >

The default directory in which the smbldap configuration files are stored is shown.

If you need to change this, enter the full directory path, then press enter to continue.

Smbldap-tools Configuration Directory Path [/etc/smbldap-tools/] >

-----

Let's start configuring the smbldap-tools scripts ...

- . workgroup name: name of the domain Samba act as a PDC  
workgroup name [amber] >
- . netbios name: netbios name of the samba controler  
netbios name [neptun] >
- . logon drive: local path to which the home directory will be connected  
(for NT Workstations). Ex: 'H:'  
logon drive [U:] >
- . logon home: home directory location (for Win95/98 or NT Workstation).



```

(use %U as username) Ex:"\\neptun\\%U'
logon home (press the "." character if you don't want homeDirectory) [\\%L\\users\\%U] >
. logon path: directory where roaming profiles are stored. Ex:"\\neptun\\profiles\\%U'
logon path (press the "." character if you don't
want roaming profile) [\\%L\\Profiles\\%a\\%U] >
. home directory prefix (use %U as username) [/home/%U] >
. default users' homeDirectory mode [700] >
. default user netlogon script (use %U as username) [] >
  default password validation time (time in days) [45] > 900
. ldap suffix [dc=amber,dc=global,dc=com] >
. ldap group suffix [ou=Groups] >
. ldap user suffix [ou=Users] >
. ldap machine suffix [ou=Users] >
. ldmap suffix [ou=ldmap] >
. sambaUnixIdPoolDn: object where you want to store the next uidNumber
and gidNumber available for new users and groups
  sambaUnixIdPoolDn object (relative to ${suffix}) [sambaDomainName=amber] >
. ldap master server: IP address or DNS name of the master (writable) ldap server
  ldap master server [127.0.0.1] >
. ldap master port [389] >
. ldap master bind dn [cn=Manager,dc=amber,dc=global,dc=com] >
. ldap master bind password [] >
. ldap slave server: IP address or DNS name of the slave ldap server: can also
be the master one
  ldap slave server [127.0.0.1] >
. ldap slave port [389] >
. ldap slave bind dn [cn=Manager,dc=amber,dc=global,dc=com] >
. ldap slave bind password [] >
. ldap tls support (1/0) [0] >
. SID for domain amber: SID of the domain (can be obtained with 'net getlocalsid neptun')
  SID for domain amber [S-1-5-21-1918777035-593721947-2697221154] >
. unix password encryption: encryption used for unix passwords
  unix password encryption (CRYPT, MD5, SMD5, SSHA, SHA) [SSHA] > MD5
. default user gidNumber [513] >
. default computer gidNumber [515] >
. default login shell [/bin/bash] >
. default skeleton directory [/etc/skel] >

```

. default domain name to append to mail address [] >

-----

backup old configuration files:

/etc/smbldap-tools/smbldap.conf->/etc/smbldap-tools/smbldap.conf.old

/etc/smbldap-tools/smbldap\_bind.conf->/etc/smbldap-tools/smbldap\_bind.conf.old

writing new configuration file:

/etc/smbldap-tools/smbldap.conf done.

/etc/smbldap-tools/smbldap\_bind.conf done.

Если мы не хотим, что бы у всех пользователей профили были перемещаемыми, то в файле /etc/smbldap-tools/smbldap.conf установим следующее значение:  
Файл: /etc/smbldap-tools/smbldap.conf

...

userProfile=""

...

Инициализируем каталоги самбы в LDAP:

smbldap-populate -a Administrator -k 0 -m 0

Конфигурация SAMBAФайл: /etc/samba/smb.conf

[global]

workgroup = amber

netbios name = neptun

realm = amber.global.com

nt acl support = yes

acl compatibility = win2k

map acl inherit = yes

server string = Samba Server %v

interfaces = eth0

bind interfaces only = yes

hosts allow = 192.168.7. 127.

```
log file = /var/log/samba/log.%m
debug level = 9
max log size = 500
socket options = TCP_NODELAY SO_SNDBUF=8192 SO_RCVBUF=8192
security = user
os level = 250
passdb backend = ldapsam:"ldap://127.0.0.1/"
enable privileges = yes
```

```
passwd program = /usr/sbin/smbldap-passwd "%u"
passwd chat = *new*password* %n\n *new*password* %n\n *successfully*
passdb expand explicit = no
unix password sync = no
ldap passwd sync = no
```

```
ldap suffix = dc=amber,dc=global,dc=com
ldap admin dn = cn=Manager,dc=amber,dc=global,dc=com
ldap user suffix = ou=Users
ldap group suffix = ou=Groups
```

```
# Т.к. для самбы компьютеры и пользователи - одно и то же,
# и искать она в дальнейшем записи компьютеров будет в пользователях,
# то для избежания дальнейших проблем при добавлении рабочих станций
# к домену мы вместо следующей строки
```

```
#
# ldap machine suffix = ou=Computers
#
```

```
# напишем другую:
```

```
ldap machine suffix = ou=Users
ldap idmap suffix = ou=Idmap
idmap backend = ldapsam:ldap://127.0.0.1/
idmap uid = 10000-20000
idmap gid = 10000-20000
```

```
ldap delete dn = Yes
ldap ssl = no
```

add user script = /usr/sbin/smbldap-useradd -n -a "%u"

delete user script = /usr/sbin/smbldap-userdel "%u"

add group script = /usr/sbin/smbldap-groupadd -p "%g"

delete group script = /usr/sbin/smbldap-userdel "%g"

add user to group script = /usr/sbin/smbldap-groupmod -m "%u" "%g"

delete user from group script = /usr/sbin/smbldap-groupmod -x "%u" "%g"

set primary group script = /usr/sbin/smbldap-usermod -g "%g" "%u"

add machine script = /usr/sbin/smbldap-useradd -w "%u"

#PDC

domain master = yes

preferred master = yes

#BDC

# domain master = no

# preferred master = no

domain logons = Yes

logon script =

# Если хотите, что бы профили всех пользователей были перемещаемыми

# и хранились на сервере (со всеми гигабайтами фильмов и личных фотографий)

# то укажите такое значение следующего параметра:

#

# logon path = \\%L\Profiles\%a\%U

#

# Если вы не хотите гонять профили по сети, оставьте значение пустым,

# (но ни в коем случае не комментируйте эту строку, она просто получит

# значение по умолчанию), вот так:

logon path =

logon drive = U:

logon home = \\%L\users\%U

#=====

Share

Definitions

[netlogon]

comment = Network Logon Service  
path = /var/lib/samba/netlogon  
browseable = yes  
guest ok = yes  
writable = no  
share modes = no

[Profiles]

admin users = admin  
create mode = 600  
directory mode = 700  
path = /var/lib/samba/profiles  
browseable = yes  
guest ok = yes  
writable = yes

[homes]

comment = Home Directories  
browseable = no  
read only = no

[public]

path = /pub  
guest ok = yes  
read only = no

[users]

path = /home/users  
writable = yes  
printable = no

Добавим запуск winbind с самбой (если нужно): Файл: /etc/conf.d/samba

...

```
daemon_list="smbd nmbd winbind"
```

```
...
```

Введём пароль рабочей станции:

```
smbpasswd -w secret
```

Введём контроллер домена, собственно в домен

```
net rpc join -S neptun -U Administrator
```

Настройка системы на авторизацию в LDAP

--ladserg 14:05, 28 июля 2006 (UTC) У меня честно говоря не получилось сделать авторизацию пользователей samba через LDAP без настройки поддержки авторизации системных пользователей в LDAP, пришлось настраивать и это.

Сначала поправим файл /etc/ldap.conf, приведя его примерно к следующему виду:

Файл: /etc/ldap.conf

```
host 127.0.0.1
```

```
base dc=amber,dc=global,dc=com
```

```
ldap_version 3
```

```
rootbinddn cn=Manager,dc=amber,dc=global,dc=com
```

```
bind_timelimit 10
```

```
bind_policy soft
```

```
pam_filter objectClass=posixAccount
```

```
pam_password exop
```

```
nss_base_passwd ou=Users,dc=tty,dc=perm,dc=ru?one
```

```
nss_base_shadow ou=Users,dc=tty,dc=perm,dc=ru?one
```

```
nss_base_group ou=Groups,dc=tty,dc=perm,dc=ru?one
```

```
nss_base_hosts ou=Hosts,dc=tty,dc=perm,dc=ru?one
```

```
nss_base_services ou=Services,dc=tty,dc=perm,dc=ru?one
```

```
nss_base_networks ou=Networks,dc=tty,dc=perm,dc=ru?one
```

```
nss_base_protocols ou=Protocols,dc=tty,dc=perm,dc=ru?one
```

```
nss_base_rpc      ou=Rpc,dc=tty,dc=perm,dc=ru?one
nss_base_ethers   ou=Ethers,dc=tty,dc=perm,dc=ru?one
nss_base_netmasks ou=Networks,dc=tty,dc=perm,dc=ru?one
nss_base_bootparams ou=Ethers,dc=tty,dc=perm,dc=ru?one
nss_base_aliases  ou=Aliases,dc=tty,dc=perm,dc=ru?one
nss_base_netgroup  ou=Netgroup,dc=tty,dc=perm,dc=ru?one
ssl off
nss_reconnect_tries 4
nss_reconnect_sleeptime 1
nss_reconnect_maxsleeptime 16
nss_reconnect_maxcontries 2
```

Сим мы скажем nss\_ldap где и как искать записи пользователей и групп.

Теперь создадим файл /etc/ldap.secret и при помощи любого текстового редактора в plain/text виде занесём туда пароль пользователя, который выше у нас указан в опции rootbinddn, например пароль secret: Файл: /etc/ldap.secret

```
secret
```

Затем непременно установим на него нужные права: Code: Установка прав на файл /etc/ldap.secret

```
#chmod 600 /etc/ldap.secret
#chown root:root /etc/ldap.secret
```

Далее приведём файл /etc/pam.d/system-auth к следующему виду:  
Файл: /etc/pam.d/system-auth

```
auth    required  pam_env.so
auth    sufficient pam_unix.so likeauth nullok
auth    sufficient pam_ldap.so use_first_pass
auth    required  pam_deny.so

account sufficient pam_ldap.so
account required  pam_unix.so
```

```

password required pam_cracklib.so difok=2 minlen=8 dcredit=2 ocredit=2 retry=3
password sufficient pam_unix.so nullok md5 shadow use_authok
password sufficient pam_ldap.so use_authok
password required pam_deny.so

session required pam_limits.so
session required pam_unix.so
session required pam_mkhomedir.so skel=/etc/skel/ umask=077
session optional pam_ldap.so

```

Обратите внимание на строку:

```
session required pam_mkhomedir.so skel=/etc/skel/ umask=077
```

Она заставляет систему создавать домашние каталоги для тех пользователей у которых они ещё не созданы, при этом в новый каталог помещается содержимое директории /etc/skel/ и задаётся маска каталога 0x700

После чего правим файл /etc/nsswitch.conf, приводя его к следующему виду:  
Файл: /etc/nsswitch.conf

```

passwd:    files ldap
shadow:    files ldap
group:     files ldap

hosts:     files dns
networks:  files dns

services:  db files
protocols: db files
rpc:       ldap [NOTFOUND=return] db files
ethers:    ldap [NOTFOUND=return] db files
netmasks:  files
netgroup:  ldap [NOTFOUND=return] files
bootparams: files

automount: files

```



aliases: files

Предупреждение: Ни в коем случае не добавляйте значение ldap к следующим базам: hosts, networks, protocols, services. Иначе вы рискуете не дождаться следующей загрузки системы.

Всё, теперь мы указали системе брать пользователей как из системных файлов, так и из LDAP.

Перезагрузим наш компьютер, дабы убедиться что система грузится нормально. Если система останавливается на загрузке udev, то смотрите ошибки в файле /etc/nsswitch.conf, может вы указали использовать ldap не в той базе.

## Управление пользователями

Ранее мы установили пакет smbldap-tools, теперь рассмотрим возможность управления пользователями с его помощью.

### Создание пользователя

```
smbldap-useradd [-o] [-a] [-b] [-w] [-i] [-u uid] [-g gid ] [-G groups,...]
[-n] [-d home] [-s shell] [-c gecos] [-m [-k]] [-t] [-P] [-A 0|1] [-B 0|1]
[-C sambaHomePath] [-D sambaHomeDrive] [-E sambaLogonScript] [-F
sambaProfilePath]
[-H sambaAcctFlags] [-N surname] [-S family name] [-M local mailAddress,...]
[-T mailToAddress] [-?] user
```

Где:

user - системное имя создаваемого пользователя

-o - add the user in the organizational unit (relative to the user suffix)

-a - is a Windows User (otherwise, Posix stuff only)

-b - is a AIX User

-w - is a Windows Workstation (otherwise, Posix stuff only)

-i - is a trust account (Windows Workstation)

-u - uid

-g - gid

-G - список групп пользователя, разделённых запятой.

-n - do not create a group

-d - домашний каталог пользователя (по умолчанию /home/имя\_пользователя)

-s - оболочка пользователя (по умолчанию /bin/false)

-c - отображаемое в Windows имя пользователя

- m - создать домашний каталог и скопировать в него файлы из /etc/skel
- k - указать иной каталог, из которого будут копироваться файлы при создании домашнего каталога пользователя (используется с ключём -m)
- t - time. Wait 'time' seconds before exiting (when adding Windows Workstation)
- P - ends by invoking smbldap-passwd
- A - возможность менять пароль пользователем, значение 0 если нет, 1 если да
- B - пользователь должен поменять пароль, значение 0 если нет, 1 если да
- C - домашний каталог samba (например '\\PDC-SRV\homes')
- D - буква диска для монтирования домашнего каталога samba (например 'H:')
- E - скрипт, выполняемый при входе в систему
- F - каталог профиля пользователя (например '\\PDC-SRV\profiles\foo')
- H - sambaAcctFlags (samba account control bits like '[NDHTUMWSLKI]')
- N - настоящее имя пользователя (для русских ещё и отчество)
- S - фамилия пользователя
- M - local mailAddress (comma seperated)
- T - mailToAddress (forward address) (comma seperated)
- ? - отобразить помощь

Например создание пользователя ladserg:

```
smbldap-useradd -a -c 'Serg Alex Lad' -N 'Serg Alex' -S 'Lad' -s /bin/bash ladserg
```

К сожалению подружить smbldap-tools с русским мне не удалось, даже при использовании кодировки UTF-8.

Итак, в приведённом выше примере будет создан пользователь с системным именем ladserg, фамилией Lad, именем Serg Alex, оболочкой /bin/bash, домашним каталогом /home/ladserg. Флаг -a укажет, что пользователь также будет являться пользователем домена.

Изменение пароля

```
smbldap-passwd [-s] [-u] [-h] username
```

Где:

- username - имя пользователя
- h, -?, --help - показать помощь
- s - обновить только samba пароль
- u - обновить только UNIX пароль

Например:

```
smbldap-passwd ladserg
```

После чего дважды будет запрошен пароль.

Теперь можно попробовать зайти в систему под учётной записью только что созданного пользователя.

Модификация пользователя

```
smbldap-usermod [-a] [-c comment] [-d home_dir] [-e expiration_date]
[-g initial_group] [-r new_login_name] [-p passwd] [-s shell] [-u uid [-o]] [-x]
[-A canchange] [-B mustchange] [-C smbhome] [-D homedrive] [-E scriptpath]
[-F profilepath] [-G group,...]] [-H acct-flags] [-N canonical_name]
[-S surname] [-P] login
```

Где:

- c - Полное имя
- d - Домашний каталог
- r - новое имя пользователя (cn, sn и dn будут обновлены)
- u - uid
- o - uid может быть не уникальным
- g - gid
- G - список групп пользователя, разделённых запятой.
- s - оболочка
- N - настоящее имя пользователя (для русских ещё и отчество)
- S - фамилия пользователя
- P - ends by invoking smbldap-passwd

For samba users:

- a - add sambaSAMAccount objectclass
- e - expire date ("YYYY-MM-DD HH:MM:SS")
- A - возможность менять пароль пользователем, значение 0 если нет, 1 если да
- B - пользователь должен поменять пароль, значение 0 если нет, 1 если да
- C - домашний каталог samba (например '\\PDC-SRV\homes')
- D - буква диска для монтирования домашнего каталога samba (например 'H:')
- E - скрипт, выполняемый при входе в систему
- F - каталог профиля пользователя (например '\\PDC-SRV\profiles\foo')

- H - sambaAcctFlags (samba account control bits like '[NDHTUMWSLKI]')
- I - disable an user. Can't be used with -H or -J
- J - enable an user. Can't be used with -H or -I
- M - mailAddresses (comma seperated)
- T - mailToAddress (forward address) (comma seperated)
- ?|-h - отобразить помощь

Например команда:

```
smbldap-usermod -A 1 ladserg
```

Позволит пользователю ladserg менять пароль. А команда:

```
smbldap-usermod -a slad-adm
```

Добавит к учётной записи пользователя slad-adm класс sambaSAMAccount, что сделает его пользователем samba.

Удаление пользователя

```
smbldap-userdel [-r|-R|-?] username
```

Где:

- r удалить домашний каталог
- R удалить домашний каталог с запросом на удаление каждого файла
- ? отобразить помощь

Например команда:

```
smbldap-userdel -r slad-adm
```

удалит пользователя slad-adm, и его домашний каталог.

Управление пользователями в оффтопике

Если вы вводите в домен компьютеры под управлением Windows, то вам пригодятся пара утилит, архив которых можно скачать отсюда:

<ftp://ftp.microsoft.com/Softlib/MSLFILES/SRVTOOLS.EXE>

# Мониторинг работы системы

## Введение

В данной статье описывается, как осуществить мониторинг разных системных и физических параметров компьютеров, работающих под GNU/Linux или другими UNIX.

## Постановка задачи

Нужно осуществлять постоянный мониторинг основных параметров работы компьютеров (физических и системных) через сеть, собирать и выдавать эту информацию в удобной форме.

Для примера см: <http://monitoring.sourceforge.net/example.html>

К основным системным параметрам относят: загрузка процессоров (пользователями, системой, привилегированными программами, свободное процессорное время), использования оперативной памяти (и многие другие, всего больше 100), дисков, сетевого трафика. Для получения необходимых нам значений параметров работы системы используем программы `df`, `sar` и `iptables` или `ipfw`.

К физическим параметрам относят: температуру процессора и материнской платы, скорость вращения вентиляторов, напряжения питания. Для получения показателей датчиков материнских плат используется модуль ядра `I2C` и программа `lm_sensors`.

Для передачи информации через сеть используем клиент-серверную модель и программу `Net-Telnet`. Базу данных строим пакетом `rrdtool`, информацию предоставляем через сервер `apache` и/или по почте. С помощью скриптов постараемся максимально упростить и автоматизировать поставленную задачу, это избавит вас от рутинной работы.

Программа работает по клиент-серверной технологии. На компьютерах, которые нужно мониторить, размещается часть программы, которая снимает различные показатели работы компьютера и выдает их по запросу через сеть. На одном административном компьютере осуществляется периодический опрос всех компьютеров, за которыми ведется наблюдение, данные записываются в специально созданную базу, и на её основе формируется `html` страница, которая экспортируется через `apache` для просмотра обычным браузером.

Теперь подробно о том, что нам нужно сделать:

Загружаем последнюю версию программы для мониторинга с сайта <http://monitoring.sourceforge.net/> Файл monitoring-\*.tar.bz2 размещаем в доступном месте, например в папке локального ftp: /var/ftp/pub/linux.

"Серверная" сторона

Разархивируйте monitoring-\*.tar.bz2.

У Gentoo есть ебилд, для установки программы monitoring со всеми зависимостями. С сайта необходимо скачать monitoring-\*.ebuild.tar.bz2 и распаковать его в /usr/local. Далее выполните:

```
# emerge -pv monitoring
```

```
[ebuild N ] net-analyzer/monitoring-0.11 USE="-admin -apache2 -lm_sensors -server"
```

Для серверной стороны необходимо добавить USE="server", а для поддержки мониторинга датчиков USE="lm\_sensors"

```
# export "USE=lm_sensors server"
```

```
# emerge monitoring
```

Компиляция и установка lm\_sensors, sysstat, iptables, coreutils, xinetd в случае использования Gentoo можно пропустить, потому что эти пакеты уже установлены как зависимости, и перейти сразу к их настройке.

Мониторинг физических параметров

Определяем на базе какого чипсета сделаны платы: материнская, видеокарта (некоторые имеют датчики) и прочие, заходим на страницы: <http://secure.netroedge.com/~lm78/supported.html> и <http://www.lm-sensors.nu/~lm78/newdrivers.html> и ищем, есть ли драйверы для мониторинга этих плат под Линуксом. Если определили, что ваши чипсеты поддерживаются и драйвер стабильный, можете продолжить выполнение этого пункта.

Внимание, если версия ядра меньше за 2.4.9, нужно обновить ядро или воспользоваться старой версией программы! смотрите: i2c. Если ядро версии 2.4.\*, вам дополнительно нужен СООТВЕТСТВУЮЩИЙ! модуль i2c-\*.tar.gz, родной Линуксовый не подходит. В версии ядра 2.6.\* все хорошо - отлично работает родной модуль ядра i2c.

## Компиляция ядра

Подробную инструкцию по сборке нового ядра можно прочесть по ссылке "Ставим ядро 2.6, или Ядерная физика для домохозяйки. Версия 2.0"

Обратите особое внимание на пункты, которые добавляют к ядру GNU/Linux поддержку сенсоров:

Для поддержания ACPI:

Power management options (ACPI, APM) ---> ACPI (Advanced Configuration and Power Interface) Support --->

Для поддержки IPMI стандарта управления сенсорами.

Device Drivers ---> Character devices ---> IPMI --->

Включаем поддержку сенсоров в ядре

Device Drivers ---> I2C support --->

Выбираем алгоритмы, использующие ваши чипы

Device Drivers ---> I2C support ---> I2C Algorithms --->

Выбираем ваш чипсет (внимание, здесь нужно указать чипсет именно вашей системной платы, или видеокарты, см. ниже):

Device Drivers ---> I2C support ---> I2C Hardware Bus support --->

Device Drivers ---> I2C support ---> Miscellaneous I2C Chip support --->

Выбираем драйверы ко всем вашим сенсорам (внимание, именно тех, соответствующих вашим чипам, предыдущим пунктам):

Device Drivers ---> Hardware Monitoring support --->

Можно собрать необходимые драйверы как модули, хотя это несколько добавит работы при настройке (нужно будет запустить программу sensors-detect). Для мониторинга сети можно использовать netfilter (iptables) и прочие опции...

Если Ваше аппаратное обеспечение поддерживается, установите lm\_sensors

Загружаем с: lm\_sensors, свеженькую версию lm\_sensors-\*.tar.gz. Разархивируем lm\_sensors-\*.tar.gz Заходим в директорию, которая создастся и выполняем:

```
# make user
```

```
# make user_install
```

```
# depmod -a
```

Проверяем, есть ли строка `/usr/local/lib` в файле `/etc/ld.so.conf`, если нет, добавляем и выполняем команду:

```
# ldconfig
```

Настройка `lm_sensors`

Выполняем команду

```
# prog/mkdev/mkdev.sh
```

Чтобы правильно определить ваше оборудование и знать, какие модули нужно загрузить, выполним:

```
# prog/detect/sensors-detect
```

именно этот скрипт и указывает, какие у вас чипсеты, и какие в них сенсоры!

Добавляем строки, созданные программой для `modules.conf` в `/etc/modules.conf`

Копируем `prog/init/lm_sensors.init` в каталог `/etc/rc.d/init.d/lm_sensors` (назначить права для исполнения).

Добавляем строки, которые создала программа `prog/detect/sensors-detect` с `modprobe` и `sensors -s` например, в конец файла `/etc/conf.d/local.start` - чтобы они загружались при старте системы

или можно запускать как сервис:

для Gentoo выполняем:

```
# rc-update -a lm_sensors default
```

Только для клонов RedHat выполняем:

```
# chkconfig --add lm_sensors
```

копируем `prog/init/lm_sensors.sysconfig` в `/etc/sysconfig/lm_sensors`

копируем `etc/sensors.conf.eg` в `/etc/sensors.conf`

копируем `prog/sensors/sensors` в каталог `/usr/bin/sensors` (предоставить права на выполнение)

Все теперь выполняют `modprobe` с параметрами, которые указала программа `sensors-detect` для загрузки модулей, потом выполняем:

```
# sensors -s
```

Корректируем `/etc/sensors.conf` таким образом, чтобы названия напряжений, скорости вентиляторов и температур, именно Ваших чипов содержали



соответственно: "Volt", "Fan", "Temp". При этом, названия меток допускают только малые и большие латинские буквы, цифры, знак "\_", и не должны содержать знаков ",", "/" или "+". Выполнение этих условий обязательно!!!

Пример /etc/sensors.conf можно найти в server/etc/sensors.conf

Например, для lm85\*:

```
# cat /etc/sensors.conf
...
# Voltage inputs
label in0 "VoltA1_5" # AGP on Intel S845WD1-E
label in1 "Volt1_5"
label in2 "Volt3_3"
label in3 "Volt5"
label in4 "Volt12"
# Temperature inputs
label temp1 "TempCPU"
label temp2 "TempMB1"
label temp3 "TempMB2"
# Fan inputs
label fan1 "FanCPU"
label fan2 "FanSys1"
label fan3 "FanSys2"
label fan4 "FanSys3"
# PWM Outputs
label pwm1 "CPUF_PWM"
label pwm2 "SysF1_PWM"
label pwm3 "SysF2_PWM"
...
```

Теперь выполним команду `sensors`, и если у нас все получилось, то увидим примерно следующее:

```
# sensors
lm85b-i2c-0-2e
Adapter: SMBus I801 adapter at c800
VoltA1_5: +1.48 V (min = +1.42 V, max = +1.58 V)
Volt1_5: +1.50 V (min = +1.45 V, max = +1.60 V)
Volt3_3: +3.33 V (min = +3.13 V, max = +3.47 V)
Volt5: +5.10 V (min = +4.74 V, max = +5.26 V)
Volt12: +12.31 V (min = +11.38 V, max = +12.62 V)
```

FanCPU: 3360 RPM (min = 3000 RPM)  
TempCPU: +36C (low = +10C, high = +60C)  
TempMB1: +33C (low = +10C, high = +45C)  
TempMB2: +33C (low = +10C, high = +45C)  
CPUF\_PWM: 255  
SysF1\_PWM: 255  
SysF2\_PWM: 77  
vid: +1.525 V (VRM Version 9.1)

Если у вас названия напряжений, скорости вентиляторов и температуры не содержат соответственно: "Volt", "Fan", "Temp" - обязательно отредактируйте файл /etc/sensors.conf как показано выше.

## Мониторинг системных параметров

### Установка sysstat

Загружаем последнюю версию пакета sysstat с сайта <http://perso.wanadoo.fr/sebastien.godard/> или <ftp://ibiblio.org/pub/Linux/system/status/>

Розархивируем файл sysstat-\*.\*.tar.bz2 в /usr/src

Заходим в созданный каталог и выполняем:

```
# make config          #Отвечаем на вопросы только 'y' или 'n'
# make
# make install
```

Теперь из командной строки пробуем:

```
# sar -A 1 1 |grep Average:
```

```
Average:   proc/s
```

```
Average:    0.00
```

```
Average:  cswch/s
```

```
Average:  258.00
```

```
Average:   CPU  %user  %nice  %system  %iowait  %idle
```

```
Average:   all  1.90   0.00   0.40   0.00  97.70
```

```
.....
```

```
Average: runq-sz  plist-sz  ldavg-1  ldavg-5  ldavg-15
```

```
Average:    0    83   0.06   0.15   0.16
```

## Проверка df

# df

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/sda3	2007996	411772	1596224	71%	/
/dev/sda1	132206	10981	114399	43%	/boot
/dev/sda5	248895	4256	231789	2%	/tmp
/dev/sda6	5863496	2556028	3307468	64%	/var
/dev/sda7	6843432	5128048	1715384	75%	/usr
/dev/sda8	1976492	1048740	827348	56%	/home
/dev/sda9	5863496	4479752	1383744	77%	/usr/portage/distfiles
none	254752	0	254752	0%	/dev/shm
/usr/portage/distfiles	5863496	4479752	1383744	77%	/var/ftp/pub/linux/distfiles
/usr/portage/packages	6843432	5128048	1715384	75%	/var/ftp/pub/linux/packages

Если Вам выдало:

```
bash: df: command not found
```

установите df из пакета coreutils.

## Мониторинг сети

Мониторинг сети основывается на подсчете байт, попадающих в правила сетевых экранов - iptables или ipfw, и поэтому они достаточно гибкие, но является зависимым от конфигурации сетевого экрана.

Если Вы хотите осуществлять мониторинг сети: просмотрите `server/etc/show_network.conf` (`/etc/monitoring/show_network.conf`) – это список графиков, которые Вы получите. Он должен отвечать названиям графиков в `admin/etc/network` (`/etc/monitoring/network`).

## Linux iptables

Нужно откорректировать `server/etc/show_iptables_acc.conf` (`/etc/monitoring/show_network.conf`)

1 колонка – название графика (должна отвечать `show_network.conf`, но с “in\_\*.N” или “out\_\*.N”, где N – некоторый идентификатор, например число)

2 колонка – название таблицы

3 колонка – название цепочки с таблицы

4 колонка – порядковый номер правила с цепочки

Это всё должно отвечать Вашим правилам!

Простые примеры экранов находятся в `server/firewall/*` и если подходит Вам, можете использовать их.

Проверьте работу `/usr/bin/show_iptables.sh`. Результат найдёте в `/home/monitoring`

## FreeBSD ipfw

Для BSD, надо откорректировать `server/bin/show_ipf_acc.conf`:

1 колонка – название графика

2 колонка – номер правила

3 колонка – порядковый номер правила в `ipfw show`

Это должно отвечать Вашим правилам!

Проверьте работу `/usr/bin/show_ipfw.sh` Результат в `/home/monitoring`

## Некоторые важные системные настройки

Если нет `xinetd` – установите его. Загляните в `server/etc/host.allow (/etc/host.allow)`:

```
# BEGIN allow services for monitoring
```

```
show_hdisk.sh:127.0.0.1 # IP ком п'ютера что мониторит
```

```
show_sensors.sh:127.0.0.1 # IP ком п'ютера что мониторит
```

```
show_info.sh:127.0.0.1 # IP ком п'ютера что мониторит
```

```
show_system.sh:127.0.0.1 # IP ком п'ютера что мониторит
```

```
show_network.sh:127.0.0.1 # IP ком п'ютера что мониторит
```

```
# END allow services for monitoring
```

Также некоторые из них вы можете изменить в `server/bin/show_*` и строки "only\_from", "bind" и "disable" в `server/xinetd.d/show_*`

```
$ cat server/xinetd.d/show_hdisk
```

```
# default: on
```

```
# description: The showdisk server show disk useg on the server.
```

```
# It dont uses authentication !!!!!!!!!!!!!!!.
```

```

service df
{
    socket_type      = stream
    wait             = no
    user             = monitoring
    bind             = 127.0.0.1          # IP сетевого интерфейса, на котором
мониторят
    server           = /usr/bin/show_hdisk.sh
    only_from        = 127.0.0.1          # IP комп'ютера, который мониторит
    disable          = no                 # Вкл./Выкл.
}

```

Откройте доступ к следующим портам: 9045/tcp 9046/tcp 9047/tcp 9048/tcp 9049/tcp с административного компьютера в случае наличия сетевых экранов. Если Вы хотите осуществить установку вручную, пропустите следующий пункт.

#### Автоматическая инсталляция

Для автоматической установки отредактируйте следующие строки в monitoringinstall.sh (/usr/sbin/monitoringinstall.sh):

```

# Edit this first:
# all command will execute user:
runuser=monitoring
rungroup=monitoring
# cron cfg directory
cronpath=/etc/cron.d
# init script locations:
crond=/etc/init.d/crond
# Edit only for server side:
# init script location:
xinetd=/etc/init.d/xinetd
# program location:
dfpath=/bin/df
sarpath=/usr/bin/sar
sensorspath=/usr/bin/sensors

```

Теперь с правами суперпользователя выполните:

```
# ./monitoringinstall.sh --server
```

В Gentoo просто:

```
# monitoringinstall.sh --server
```

Вы можете пропустить пункт следующий пункт и перейти к „административной” части системы.

### Ручная установка

Копируем следующие файлы:

```
cp server/etc/show_iptables_acc.conf /etc/monitoring #for GNU/Linux
cp server/etc/show_ipfw_acc.conf /etc/monitoring      #for BSD
cp server/etc/show_network.conf /etc/monitoring
cp server/xinetd.d/show_hdisk /etc/xinetd.d/
cp server/xinetd.d/show_network /etc/xinetd.d/
cp server/xinetd.d/show_sensors /etc/xinetd.d/
cp server/xinetd.d/show_system /etc/xinetd.d/
cp server/bin/show_hdisk.sh /usr/bin
cp server/bin/show_iptables.sh /usr/bin               #for GNU/Linux
cp server/bin/show_ipfw.sh /usr/bin                   #for BSD
cp server/bin/show_network.sh /usr/bin
cp server/bin/show_sar.sh /usr/bin
cp server/bin/show_sensors.sh /usr/bin
cp server/bin/show_system.sh /usr/bin
```

Добавим следующие строки к /etc/service:

network	9045/tcp	# show network info
info	9046/tcp	# show server info
df	9047/tcp	# show disk info
sys	9048/tcp	# show system info
sensors	9049/tcp	# show sensors info

Создадим специального пользователя:

```
groupadd monitoring
useradd -g monitoring -d /home/monitoring -s /bin/bash -c monitoring monitoring
mkdir /home/monitoring
chmod 700 /home/monitoring
chown -R monitoring:monitoring /home/monitoring
```

В случае использования политик SELinux необходимо также добавить:

```
# cat /etc/security/selinux/src/policy/users
```

```
...
```

```
# BEGIN monitoring selinux:
```

```
user monitoring roles user_r;
```

```
# END monitoring selinux.
```

и дополнительно выполнить:

```
# cd /etc/security/selinux/src/policy
```

```
# make load
```

Перезгрузим xinetd:

```
# /etc/init.d/xinetd restart
```

и добавьте строки в crontab:

```
*/10 * * * * monitoring /usr/bin/show_iptables.sh      # для GNU/Linux
```

```
*/10 * * * * monitoring /usr/bin/show_ipfw.sh         # для BSD
```

```
*/1 * * * * monitoring /usr/bin/show_sar.sh
```

Перезгрузим crond:

```
# /etc/init.d/crond restart
```

Все вышесказанное нужно сделать на КАЖДОМ сервере, который вы хотите мониторить...

"Административная" сторона

Для начала убедитесь что все хорошо со стороны серверов

Для каждого сервера проверяем доступность необходимых сервисов:

```
# telnet server_name 9045
```

```
Trying serverIP...
```

```
Connected to serverIP.
```

```
Escape character is '^['.
```

```
in_ftp          0
```

```
out_ftp         0
```

```
in_http         0
```

```
out_http        0
```

```
in_other      249
out_other     27
Connection closed by foreign host.
```

```
# telnet server_name 9046
Trying serverIP...
Connected to server_name.
Escape character is '^]'.
  Intel(R) Celeron(R) CPU 2.40GHz | 504 Mb
Connection closed by foreign host.
```

```
# telnet server_name 9047
Trying serverIP...
Connected to server_name.
Escape character is '^]'.
/dev/sda3      2007996  411772  1596224  71% /
/dev/sda1      132206   10981   114399  43% /boot
/dev/sda5      248895    4256   231789  2% /tmp
/dev/sda6      5863496  2556028  3307468  64% /var
/dev/sda7      6843432  5128048  1715384  75% /usr
/dev/sda8      1976492  1048740   827348  56% /home
/dev/sda9      5863496  4479752  1383744  77% /usr/portage/distfiles
Connection closed by foreign host.
```

```
# telnet server_name 9048
Trying serverIP...
Connected to server_name.
Escape character is '^]'.
Average:      proc/s
Average:       3.98
Average:      cswch/s
Average:     308.00
Average:      CPU  %user  %nice  %system  %iowait  %idle
Average:     all  1.76   0.00   0.47   0.00  97.76

.....
Average:  runq-sz  plist-sz  ldavg-1  ldavg-5  ldavg-15
Average:      1    88    0.05   0.21   0.21
```



Connection closed by foreign host.

```
# telnet server_name 9049
```

Trying serverIP...

Connected to server\_name.

Escape character is '^['.

lm85b-i2c-0-2e

Adapter: SMBus I801 adapter at c800

Volt1\_5: +1.48 V (min = +1.42 V, max = +1.58 V)

VoltCore: +1.50 V (min = +1.45 V, max = +1.60 V)

Volt3\_3: +3.33 V (min = +3.13 V, max = +3.47 V)

Volt5: +5.10 V (min = +4.74 V, max = +5.26 V)

Volt12: +12.25 V (min = +11.38 V, max = +12.62 V)

CPU\_Fan: 3377 RPM (min = 3000 RPM)

fan2: 0 RPM (min = 0 RPM)

fan3: 0 RPM (min = 0 RPM)

fan4: 0 RPM (min = 0 RPM)

TempCPU: +32 C (low = +10 C, high = +50 C)

TempBoard: +30 C (low = +10 C, high = +45 C)

TempRemot: +30 C (low = +10 C, high = +40 C)

CPU\_PWM: 255

Fan2\_PWM: 255

Fan3\_PWM: 77

vid: +1.525 V (VRM Version 9.0)

Connection closed by foreign host.

Установка необходимых программ

Разархивируйте monitoring-\*.tar.bz2.

Для Gentoo есть ебилд, для установления программы monitoring со всеми зависимостями с сайта стоит брать monitoring-\*.ebuild.tar.bz2 и распаковать его в /usr/local дальше выполните:

```
# emerge -pv monitoring
```

```
[ebuild N ] net-analyzer/monitoring-0.11 USE="-admin -apache2 -lm_sensors -server"
```

Для административной стороны необходимо добавить USE=admin, а для поддержки apache-2\* USE=apache2

```
# export "USE=admin apache2"
```

```
# emerge monitoring
```

Компиляцию и установку rrdtool, apache, Net-Telnet в случае использования Gentoo можно пропускать, потому что эти пакеты уже будут установлены как зависимости, а переходить сразу к их настройке.

## Инсталляция rrdtool

Базой данных, для хранения информации и инструмент генерации графиков, используется пакет rrdtool (round robin database tool). Скачать пакет можно отсюда: <http://rrdtool.eu.org> или <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/> Подробно ознакомиться с этим пакетом можно здесь: <http://www.bog.pp.ru/work/rrdtool.html> (Российская). В принципе, программа самостоятельно создаст необходимые базы так, что дополнительных знаний не нужно и следующий пункт Пример расчёта настройки базы RRD можно пропустить.

## Пример расчёта настройки базы RRD

Расчёт настроек создания базы RRD файлы /etc/monitoring/\*.rrd.cf Количество "слов" в файле будет соответствовать количеству графиков в разных масштабах времени. например для расчёта подобия MRTG: 5-ти минутный - данные за последние 2 дня: x-доля=0.5 количество отсчётов на ячейку=1 количество ячеек= $2 \times 24 \times 12 = 576$  +min+max+avg+cur ~ 650

0.5:1:650

30-ти минутный - данные за последние две недели: x-доля=0.5 количество отсчётов на ячейку= $30/5=6$  количество ячеек= $2 \times 7 \times 24 \times 2 = 672$  +min+max+avg+cur ~ 750

0.5:6:750

2-ух часовой - данные за последние два месяца: x-доля=0.5 количество отсчётов на ячейку= $60 \times 2/5=24$  количество ячеек= $2 \times 31 \times 24/2 = 744$  +min+max+avg+cur ~ 850

0.5:24:850

1-но дневный - данные за последние два года:  $x\text{-доля}=0.5$  количество отсчётов на ячейку= $60*24/5=288$  количество ячеек= $2*366=732$  +min+max+avg+cur ~ 900

0.5:288:900

Таким образом имеем:

```
# cat /etc/monitoring/5min.rrd.cf
```

0.5:1:650    0.5:6:750    0.5:24:850    0.5:288:900

### Инсталляция Net-Telnet

Данная версия программы для работы через сеть требует Net-Telnet perl модуль (можно при желании использовать и netcat): из-за этого нужно установить perl (что пожалуй уже сделано). Далее Net-Telnet perl модуль берем и устанавливаем отсюда: <http://cpan.perl.org/modules/by-module/Net/> название файлу Net-Telnet-\*.tar.gz

### Инсталляция Apache

Нужно установить и настроить apache сервер, дополнительно можете добавить (admin/etc/apache/monitoring.conf):

```
# cat /etc/apache2/vhosts.d/monitoring.conf
```

```
### /etc/apache2/vhosts.d/monitoring.conf
```

```
### $Id: monitoring.conf,v 0.11 2006/09/28 16:27:12 hse Exp $
```

```
###
```

```
### For Monitoring *.shtml
```

```
AddType text/html .shtml
```

```
AddOutputFilter INCLUDES .shtml
```

```
### For authentication:
```

```
<Directory "/var/www/localhost/htdocs/administration/monitoring">
```

```
    AllowOverride All
```

```
    Options Includes
```

```
    <IfModule mod_access.c>
```

```
### For password authentication:
```

```
#    AuthUserFile /var/www/localhost/htdocs/administration/passwd_file
```

```
#    AuthGroupFile /dev/null
```

```
#    AuthName Administrative-information
```

```
#    AuthType Basic
```

```
#    Require valid-user
```

### For host/network authentication:

Order deny,allow

Deny from all

Allow from 127.0.0.1

# Allow from .cluster.linux

</IfModule>

</Directory>

Добавляем пользователя для просмотра веб страницы:

```
# htpasswd -bcm /var/www/localhost/htdocs/administration/passwd_file username  
password
```

Если все хорошо идем дальше.

Установка программ для мониторинга

В файл admin/etc/host (/etc/monitoring/host) вносим перечень серверов, которые нужно мониторить (имена должны разрешаться в IP адреса). Корректируем под свои потребности файл admin/etc/monitoring.conf (/etc/monitoring/monitoring.conf)

путь по которому установлено скрипти:

installpath=/usr/

путь к веб страницы:

apachehtmldir=/var/www/html

webdirpath=/administration/monitoring

путь к файлу с перечнем серверов для мониторинга:

confpath=\$installpath/etc/host

путь к временных файлов:

diskinfopath=/tmp/monitoring/disk.tmp

sarinfopath=/tmp/monitoring/sar.tmp

sensorsinfopath=/tmp/monitoring/sensors.tmp

путь где установлен пакет rrdtool:

rrdtoolpath=/usr/bin/rrdtool

почтовый адресов:

emailvalue=root@localhost

По желанию (или за потребностью) правим другие файлы с каталога etc/ (/etc/monitoring/):

diskusage memusage quantity cpu memory systemload - содержимое файлов должно отвечать тем параметрам которые Вы хотите мониторить. Все параметры должны выводиться командой `sar -A 1 1 |grep Average`: Названия файлов отвечают названиям графиков и баз которые создадутся и должны быть записаны в файле `system`. Значение параметров принадлежащих одному файлу должны быть одного типа!

`system` - первая колонка определяет какие графики получите и каждый ее элемент отвечать файлу из этого же каталога (см. выше). Вторая указывает через какие периоды времени база `rrdtool` будет принимать данные. Третья и четвертая колонка определяет минимальное и максимальное значение параметра которое еще может быть записано в базу. Пятая колонка определяет легенду (размерность).

`colors` - файл с номерами цветов в 16-ричной системе.

`config` - содержит имена конфигурационных файлов из этого же каталога отвечающие веб страницам что создаются (можно добавлять свои, но тогда их должны создать!).

`disk` - содержит названия графиков, периоды времени через которые база `rrdtool` будет принимать данные, минимальные и максимальные значение параметров которые мониторите, легенду (размерность).

`network` - содержит названия графиков, периоды времени через которые база `rrdtool` будет принимать данные, минимальные и максимальные значение параметров которые мониторите, легенду (размерность).

`sensors` - содержит названия графиков, периоды времени через которые база `rrdtool` будет принимать данные, минимальные и максимальные значение параметров которые мониторите, легенду (размерность).

`disk.rrd.cf network.rrd.cf sensors.rrd.cf system.rrd.cf` - описывают структуру базы данных.

`disk.msg network.msg sensors.msg system.msg` - содержит информацию которую получите почтой в критических ситуациях.

Если хотите осуществить установку вручную, пропустите следующий пункт.

Автоматическая установка программ

Теперь можете воспользоваться скриптом для инсталляции. Для этого нужно откорректировать следующие строки в файле `monitoringinstall.sh` (`/usr/sbin/monitoringinstall.sh`) или пропустить этот пункт и следовать дальнейшей инструкции для установки в ручную.

имя пользователя и группы:

`runuser=monitoring`

`rungroup=monitoring`

директория где расположены cron настройки:

`cronpath=/etc/cron.d`

путь к инициализационному скрипту

`crond crond=/etc/init.d/crond`

директория где находятся настройки apache

`apachconfdir=/etc/apache/conf`

файл конфигурации apache:

`apachconffile=apache.conf`

путь к инициализационному скрипту apache

`apached=/etc/init.d/apache`

Теперь с правами рута выполните (нужно подождать 1 минуту, если не установили мониторинг сети, и 10 минут после установки монитора сети на последнем сервере, прежде чем выполнять следующую команду):

```
# ./monitoringinstall.sh --admin
```

В Gentoo просто

```
# monitoringinstall.sh --admin
```

Если видите ошибки, наверное надо:

1 поправить файлы конфигурации в `admin/etc/` (`/etc/monitoring`)

2 посмотреть в `/home/monitoring/hostname/err/...`

3 удалить файлы с `/home/monitoring/hostname/` и перезапустить скрипт

4 удалить базы данных с `/var/db/monitoring/...` и перезапустить скрипт

5 `bin/net_telnet.pl` читает только 500 строк входа. Если в Вас больше, отредактируйте его:

```
line 32: while [$i ne 500]
```

Тогда попробуйте еще, но добавьте имена сервисов, под рутом выполните:

```
# ./monitoringinstall.sh --admin=disk
```

```
# ./monitoringinstall.sh --admin=sensors
```

```
# ./monitoringinstall.sh --admin=system
# ./monitoringinstall.sh --admin=network
```

Если где-то допустили ошибку, можно просто выполнить `monitoringuninstall.sh` и попытаться еще раз. Можете пропустить следующий пункт.

### Ручная установка программ

Устанавливаем скрипты для мониторинга, генерации базы данных и веб страницы скопировав с каталога `monitoring/admin` в каталог: `/usr/ ($INSTALL_PATH)`, используя опцию `-p` для сохранения прав. Выполняем:

```
# cp -p admin/bin/* $INSTALL_PATH/bin/
# mkdir /etc/monitoring
# cp -p admin/etc/* /etc/monitoring/
# mkdir $INSTALL_PATH/share/monitoring
# cp -p admin/share/* $INSTALL_PATH/share/monitoring/

# mkdir -p /var/www/localhost/htdocs/administration/monitoring
# chmod 755 /var/www/localhost/htdocs/administration/monitoring/
# groupadd -g 1111 monitoring
# useradd -u 1111 -g monitoring -d /home/monitoring -s /bin/bash -c monitoring monitoring
# chmod -R 700 /home/monitoring
# chown -R monitoring:monitoring /home/monitoring
# chown -R monitoring:monitoring /var/www/localhost/htdocs/administration/monitoring/
```

И тогда выполняйте (подождите 1 минуту, если не устанавливали мониторинг сети, и 10 минут после установки мониторинга сети на последнем сервере, прежде чем выполнять следующую команду):

```
# su monitoring
$ $INSTALL_PATH/bin/net.vert.1.sh
$ $INSTALL_PATH/bin/net.vert.10.sh
```

создаются базы данных в каталоге `/var/db/monitoring/имя_сервера/` для всех серверов с файлу `/etc/monitoring/host` и дополнительная конфигурация в `/home/monitoring`

Выполняем:

```
$ $INSTALL_PATH/bin/png_graph.sh disk -1week 30min
$ $INSTALL_PATH/bin/png_graph.sh disk -1month 2hour
```

```
$ $INSTALL_PATH/bin/png_graph.sh disk -1year 1day
$ $INSTALL_PATH/bin/png_graph.sh network -1week 30min
$ $INSTALL_PATH/bin/png_graph.sh network -1month 2hour
$ $INSTALL_PATH/bin/png_graph.sh network -1year 1day
$ $INSTALL_PATH/bin/png_graph.sh sensors -2day 5min
$ $INSTALL_PATH/bin/png_graph.sh sensors -1week 30min
$ $INSTALL_PATH/bin/png_graph.sh sensors -1month 2hour
$ $INSTALL_PATH/bin/png_graph.sh sensors -1year 1day
$ $INSTALL_PATH/bin/png_graph.sh system -2day 5min
$ $INSTALL_PATH/bin/png_graph.sh system -1week 30min
$ $INSTALL_PATH/bin/png_graph.sh system -1month 2hour
$ $INSTALL_PATH/bin/png_graph.sh system -1year 1day
```

Если видите ошибки, наверное нужно:

- 1 поправить файлы конфигурации в /etc/monitoring
- 2 посмотреть в /home/monitoring/hostname/err/...
- 3 удалить файлы с /home/monitoring/hostname/ и перезапустить скрипт
- 4 удалить базы данных с /var/db/monitoring/... и перезапустить скрипт
- 5 bin/net\_telnet.pl читает только 500 строк входа. Если в Вас больше, отредактируйте его:  
line 32: while [\$i ne 500]

Тогда попробуйте еще раз.

В каталоге /var/www/localhost/htdocs/administration/monitoring создаются начальная страница и каталоги с веб страницами для каждого сервера.

Если не возникло больше ошибок, используйте следующий файл для автоматизации всей работы с crond:

```
$ exit
# cat /etc/cron.d/monitoring
```

```
SHELL=/bin/bash
PATH=/bin:/usr/bin
MAILTO=root
HOME=/home/monitoring
LANG=POSIX
```



# Begin server part (may be you wont monitoring "admin" computer too)

\*/1 \* \* \* \* monitoring /usr/bin/show\_sar.sh

\*/10 \* \* \* \* monitoring /usr/bin/show\_iptables.sh #GNU/Linux

\*/10 \* \* \* \* monitoring /usr/bin/show\_ipfw.sh #BSD

# End server part

# Begin disk admin part

\*/30 \* \* \* \* monitoring /usr/bin/png\_graph.sh hdisk -1week 30min

0 \*/2 \* \* \* monitoring /usr/bin/png\_graph.sh hdisk -1month 2hour

0 0 \* \* \* monitoring /usr/bin/png\_graph.sh hdisk -1year 1day

# End disk admin part

# Begin network admin part

\*/30 \* \* \* \* monitoring /usr/bin/png\_graph.sh network -1week 30min

0 \*/2 \* \* \* monitoring /usr/bin/png\_graph.sh network -1month 2hour

0 0 \* \* \* monitoring /usr/bin/png\_graph.sh network -1year 1day

# End network admin part

# Begin system admin part

\*/5 \* \* \* \* monitoring /usr/bin/png\_graph.sh system -2day 5min

\*/30 \* \* \* \* monitoring /usr/bin/png\_graph.sh system -1week 30min

0 \*/2 \* \* \* monitoring /usr/bin/png\_graph.sh system -1month 2hour

0 0 \* \* \* monitoring /usr/bin/png\_graph.sh system -1year 1day

# End system admin part

# Begin sensors admin part

\*/5 \* \* \* \* monitoring /usr/bin/png\_graph.sh sensors -2day 5min

\*/30 \* \* \* \* monitoring /usr/bin/png\_graph.sh sensors -1week 30min

0 \*/2 \* \* \* monitoring /usr/bin/png\_graph.sh sensors -1month 2hour

0 0 \* \* \* monitoring /usr/bin/png\_graph.sh sensors -1year 1day

# End sensors admin part

# Begin vert admin part

\*/1 \* \* \* \* monitoring /usr/bin/net.vert.1.sh

\*/10 \* \* \* \* monitoring /usr/bin/net.vert.10.sh

# End vert admin part

Выполняем:

```
# /etc/init.d/crond restart
```

Если все хорошо

Теперь заходим Вашим любимым браузером на страницу <http://servername/administration/monitoring/index.html>. Здесь уже можем выбрать ссылки на сервер, который вас интересует :)

Добавление новых серверов

Если Вам когда-то придется добавить новый сервер, достаточно будет сделать следующее:

1. Выполнить для него пункт 2 („Серверная” сторона)
2. Добавить его имя в файл `/etc/monitoring/host`

Всё остальное создастся само !!! Наслаждайтесь :)))

Альтернативы

Если Вы недовольны реализацией мониторинга таким образом (по моему мнению достаточно неплохое решение ;) ) подберите себе лучшую. Вот некоторый перечень альтернатив или дополнений :=)

Angel Network Monitor <http://www.paganini.net/angel/>

Autostatus <http://www.angio.net/consult/autostatus/>

Cacti <http://cacti.net>

HiWAYs <http://www.hiways.org/>

MARS <http://www.altara.org/mars.html>

Mon <http://www.kernel.org/software/mon/>

Monit <http://www.tildeslash.com/monit/>

Nagios <http://www.nagios.org>

Netup (French) <http://www.pasteur.fr/units/sis/netup/>

NocMonitor <http://www2.discpro.org/nocmon/>

NodeWatch <http://www.skendric.com/nodewatch/>

Penemo <http://www.communityprojects.org/apps/penemo/>

PIKT <http://pikt.org/>

RITW [http://www.terravista.pt/Ancora/1883/ritw\\_e.html](http://www.terravista.pt/Ancora/1883/ritw_e.html)

RRDWorld <http://oss.oetiker.ch/rrdtool/rrdworld/index.en.html>

Scotty <http://wwwhome.cs.utwente.nl/~schoenw/scotty/>

Spong <http://spong.sourceforge.net/>

Sysmon <http://www.sysmon.org/>

ZABBIX <http://www.zabbix.com>

ZEUS <http://www.zeus.com/>

## ***Настройка X***

### **Настройка русских шрифтов в X**

#### Введение

#### Введение временное

Ну вот и начался этот проект, призванный восполнить достаточно широкий пробел в документации по настройке шрифтов в Иксах. Надеюсь многочисленные русские пользователи проявят некую сознательность и отпишут сюда пару строчек.

#### Кто принимает участие

Macil - линуксоид из Ярославля. Обитает на [yar.lancktelecom.ru/talk](http://yar.lancktelecom.ru/talk), [forum.median.ru](http://forum.median.ru), а также Macil [at] jabber.ru

Wanderer - пользователь Linux из Москвы. Всерьез занимаюсь изучением и использованием Linux последние 4 года, до этого просто приглядывался ;-). На Gentoo пересел почти год назад и ничуть не жалею и даже наоборот, очень этим доволен. Связаться со мной можно посредством e-mail ([wanderer\\_mg \[at\] mail.ru](mailto:wanderer_mg@mail.ru)) или jabber ([wanderer \[at\] jabber.ru](http://jabber.ru)).

#### Две системы рисования шрифтов

Исторически в X Window System существует 2 системы рисования шрифтов. Родная (core) и Xft.

Родная система – прямой потомок системы рисования шрифтов, введенной в X11R1, выпущенной в 1987 году. Изначально она могла рисовать только монохромные растровые шрифты, но на протяжении всех этих лет предпринимались более-менее успешные попытки заставить ее работать с векторными шрифтами.

Xft не является неотъемлемой частью X, хотя возможно будет ей в будущем. Это значит, что она распространяется не со всеми реализациями X. Она проектировалась для работы с векторными шрифтами, и делает это достаточно эффективно. В отличие от родной системы рисования, Xft поддерживает anti-aliasing и sub-pixel rasterisation. Кроме того, она позволяет приложениям полностью контролировать то, как будут рисоваться шрифты, обеспечивая возможность

создания WYSIWYG приложений.

Нужно ли уточнять что эти две подсистемы совершенно не совместимы на уровне API? Из-за достаточно анархичного развития X в последнее время, возникла большая путаница в определениях и версиях используемых библиотек.

//Источник: вольный перевод файла README.fonts

Родная система

Это — незавершённая статья. Вы можете помочь проекту, исправив и дополнив материал.

Xft

Это — незавершённая статья. Вы можете помочь проекту, исправив и дополнив материал.

Введение в Xft

Xft это библиотека, которая предоставляет клиентский API для рисования шрифтов. В настоящее время версия Xft 2.0 использует библиотеки fontconfig и freetype для выбора шрифтов и их рисования, соответственно. Xft 1.0 использовала внутренний механизм конфигурации, что было не очень хорошей идеей так как KDE и Pango "заимствовали" механизм конфигурации шрифтов и создали еще большую путаницу. Подробности можно посмотреть [\[здесь\]](#).

Нужно запомнить 3 вещи:

fontconfig - для конфигурации

freetype - для рисования

XFT - связывает все во едино

Freetype - библиотека рисования шрифтов

[Freetype] - совершенно независимый от кого-либо проект по созданию библиотеки рисования шрифтов. Библиотека предоставляет достаточно простой API, но поскольку обычные приложения заинтересованы в более высокоуровневых сервисах, в чистом виде не используется. Вместе с X идет модифицированная freetype, предназначенная для поддержки векторных шрифтов родной системой рисования. Я не вижу смысла даже пытаться ее компилировать (она компилируется

в модуль freetype), так как большинство приложений перешли на использование Xft и путанице со шрифтами приходит конец. Рендеринг векторных шрифтов очень сложный процесс, и к сожалению, freetype не может его выполнять до конца и в некоторых шрифтах появляются не очень приятные артефакты. Но ничего лучшего (открытого и бесплатного) пока нет... Как обычно, существует 2 версии freetype: freetype1 и freetype2. Отличия их в том, что freetype1 может рисовать только шрифты формата TrueType. Сложно найти современный дистрибутив, где отсутствует freetype2.

## Теория рендеринга векторных шрифтов

Это — незавершённая статья. Вы можете помочь проекту, исправив и дополнив материал.

### Что такое hinting

На английском много информации о hinting можно найти тут <http://www.myfonts.com/activity/hinting/>

Не уверен, однако, что можно тут публиковать такие ссылки.

Это — незавершённая статья. Вы можете помочь проекту, исправив и дополнив материал.

### Что такое anti-aliasing

Это методы избавления от артефактов, возникающих при отрисовке векторного изображения в растровое. Когда векторное изображение (не дискретное, к примеру, формула  $y=2*x$ ) отрисовывается в растровое, то в результате погрешностей и того, что цвет каждого пикселя строго зависит от формулы, возникают артефакты. То есть наклонную линию мы будем видеть ступенчато, а не плавно. Методы anti-aliasing, в основном, базируются на отрисовке изображения в большее разрешение и приведение к обычному путём билинейной фильтрации, т.е. цвет пикселя будет строиться из цветов нескольких пикселей большого изображения. Шрифты после применения anti-aliasing выглядят более гладко, т.к. более точно соответствуют "векторной формуле".

Это — незавершённая статья. Вы можете помочь проекту, исправив и дополнив материал.

### Шрифты из Microsoft Windows

Это — незавершённая статья. Вы можете помочь проекту, исправив и дополнив материал.

Установка шрифтов Microsoft Windows имея доступ в интернет

Если есть доступ к интернету, то можно просто установить пакет corefonts: emerge corefonts

Настройка родной подсистемы

Это — незавершённая статья. Вы можете помочь проекту, исправив и дополнив материал.

Настройка Xft

Все настройки Xft хранятся в файлах /etc/fonts/fonts.conf и /etc/fonts/local.conf. Все изменения нужно производить в файле /etc/fonts/local.conf. Изменение файла /etc/fonts/fonts.conf не рекомендуется, т.к. при обновлении пакета fontconfig он будет перезаписан и все настройки потеряны.

Это — незавершённая статья. Вы можете помочь проекту, исправив и дополнив материал.

Отключение сглаживания для шрифтов меньше определенного размера

По умолчанию сглаживание действует для всех шрифтов, но для шрифтов маленького размера его лучше отключить, т.к. смотрится это почти всегда плохо.

Ниже приведен пример конфига /etc/fonts/local.conf, в котором указана отмена сглаживания для шрифтов, размер которых меньше 14. Code: /etc/fonts/local.conf

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<!-- /etc/fonts/local.conf file for local customizations -->
<fontconfig>
```

```
<!-- Enable sub-pixel rendering -->
```

```
  <match target="font">
    <edit name="rgba" mode="assign"><const>rgb</const></edit>
  </match>
  <match target="font">
    <test name="size" compare="less_eq">
```

```
        <double>14</double>
    </test>
    <edit name="antialias" mode="assign">
        <bool>>false</bool>
    </edit>
</match>
<match target="font">
    <test name="pixelsize" compare="less_eq">
        <double>14</double>
    </test>
    <edit name="antialias" mode="assign">
        <bool>>false</bool>
    </edit>
</match>

</fontconfig>
```

Настройка Qt и Gtk

Для Qt

```
emerge qt qtconfig
```

Для Gtk

```
emerge gtk-theme-switch switch2
```

Это — незавершённая статья. Вы можете помочь проекту, исправив и дополнив материал.

Различные приложения

Это — незавершённая статья. Вы можете помочь проекту, исправив и дополнив материал.

Legacy приложения

Это — незавершённая статья. Вы можете помочь проекту, исправив и дополнив материал.

## Настройка размера шрифтов в X

### Введение

Столкнулся с проблемой (и не я один) размера шрифтов.

Процесс сталкивания выглядел так:

```
~ $ emerge xorg-x11
```

```
...
```

```
~ $ emerge kde
```

```
...
```

```
startx
```

Попал я, естессна сразу в KDE. И увидел очень мелкие шрифты, очень мелкие, похожие на точки. Примерно такие:

Естессна, первое, что я сделал - полез в центр управления и установил размер, соответствующий моему зрению. Им оказался 40. Немало. Тогда я понял, что что-то неправильно. Порылся в инете, не нашел ничего. Смирился. Однако вскоре узнал, что та же проблема у меня и в KDM, только там изменить размер немного сложнее. Изменил. Но проблему это не решило, потому что нашлись еще всякие шибко умные проги, которые пользовались собственными настройками шрифтов. Их (настройки) искать оказалось очень сложно и пришлось брать быка за рога. Итак,

### Суть проблемы

Она состоит в том, что шрифты ориентированы на определенное физическое разрешение экрана - dpi (точек на дюйм). В винде по дефолту оно идет равным 96. X11 решил, что это слишком сложно не знаю для кого и, как выяснилось позже, выставил себе аж целых 25 dots per inch. В переводе в метрическую систему это около 1 точки на миллиметр. Соответственно, экран разрешением 1024x768 должен быть шириной в метр, высотой 77 см, чтоб читалось все как надо. Ну а поскольку реально экран у нас меньше, мы видели то, что мы видели.

### Неправильный ход мысли



но полезный совет. Увеличить размер шрифта для KDM можно в этом файле:

`/usr/kde/3.5/share/config/kdm/kdmrc`

Но это не выход, как я уже говорил. Кстати, GDM настраивается в другом файле, его я не знаю, потому что прешел к следующему пункту:

Правильный ход мысли

Надо изменить разрешение, те самые dpi. Где? Как?

Меняем dpi

Тут, кстати, возможны варианты. Если Вы запускаете иксы из консоли по мере необходимости командой `startx`, то Вам сюда. Если иксы стартуют сами при загрузке системы - то сюда. В идеале стоит проделать оба варианта.

Для команды `startx`

Лезем туда, где этот стартикс лежит и правим его

`~ $ find /usr -name startx`

У меня он оказался в `/usr/bin`, и с вероятностью 99,(9)% там же он будет и у Вас.

Вписать в него надо всего ничего параметр `-dpi 96` в ключ `defaultserverargs`  
`defaultserverargs="-dpi 96"`

Те параметры, которые там есть не удаляйте, у них свое назначение, этот просто допишите к ним.

В принципе, на этом все. Можно

`~ $ startx`

Жизнь удалась.

Для пользователей KDM, XDM и тому подобного

Вооружитесь калькулятором. В файле `xorg.conf` Вам надо изменить параметр `DisplaySize`. По умолчанию, он устанавливается равным разрешению (помните, я говорил в начале статьи), но ваш-то монитор меньше. Можно померять линейкой, можно посчитать, какие значения ему присвоить:

$X = x * 25,4 / \text{dpi}$

$Y = y * 25,4 / \text{dpi}$

где X, Y - значения, которые надо присвоить DisplaySize  
x, y - разрешение монитора (x - ширина, y - высота)  
dpi - то физическое разрешение, которое должно получиться

У меня при разрешении 1024x768 получилось так:

```
~ $ vi /etc/X11/xorg.conf
```

Section "Monitor"

Identifier "Monitor0"

VendorName "HSD"

ModelName "HSD150PX17-A"

DisplaySize 270.9 203.2 # mm

EndSection

Вывод

Ерунда какая, ничего сложного, если сразу заглянуть в xorg.conf и почитать комментарии, то этого можно избежать. Но я не избежал и, кстати, долго с этим мучался.

Вывод-то в чем: будьте внимательнее!

P.S.

В Gnome есть такая настроечка, как dpi, сам не видел, но знающие люди говорят, что есть и рекомендуют ее тоже выставить на 96 или то, которое принято везде.

## **Xorg X11 и прозрачность**

Введение

Это руководство поможет в установке X.org на Linux/Gentoo машине, хотя общие настройки должны работать и на любом другом дистрибутиве.

Шаг за шагом

Установка X.org

Установите последнюю версию X.org с помощью команды `emerge xorg-x11`

После компиляции необходимо настроить X.org путем редактирования `/etc/X11/xorg.conf`

Если у вас было установлено XFree86 то рабочий `/etc/X11/XF86Config` будет автоматически использоваться для работы X.org. Для обеспечения целостности рекомендуется переименовать `/etc/X11/XF86Config` в `/etc/X11/xorg.conf`.

Для загрузки X.org выполните команду `startx`

## Установка X Composite Extension

Откройте `/etc/X11/xorg.conf` для редактирования `nano -w /etc/X11/xorg.conf` пользуйте ваш любимый редактор ;)

Добавьте следующие строки в любом месте конфиг файла Файл: `/etc/X11/xorg.conf`

Section "Extensions"

Option "Composite" "Enable"

Option "RENDER" "Enable"

EndSection

При следующей перезагрузке X.org у вас будет включен Composite Extension. Это расширение включает предварительную отрисовку вне экрана, чтобы на экран выводилось уже законченное изображение. Это позволяет использовать различные эффекты (например прозрачность и тени).

Для достоверности можно проверить лог файл `grep COMPOSITE /var/log/Xorg.0.log`

(II) Initializing built-in extension COMPOSITE

Включение теней и настоящей прозрачности

KDE 3.4 или более поздняя

Начиная с версии 3.4 KDE поддерживает прозрачность через Xorg. На заголовке любого окна щелкните правой кнопкой мыши, выберите "настройка поведения окна" и в списке слева надите пункт полупрозрачность. Поставьте флажок "использовать прозрачность/тени", после чего станут доступны множество настроек прозрачности (активного, неактивного окна, окна при перетаскивании, плавные переходы прозрачности и т.д.) и теней.

xcompmgr и transset

Многие WM еще не обладают поддержкой новых функций X.org, так что нам придется использовать дополнительные инструменты для наших эффектов.

Вначале нам надо установить xcompmgr

```
echo "x11-misc/xcompmgr ~x86" >> /etc/portage/package.keywords
```

```
emerge xcompmgr
```

Затем установим transset

```
echo "x11-misc/transset ~x86" >> /etc/portage/package.keywords
```

```
emerge transset
```

Теперь у нас есть все необходимое, чтобы включить тени, затенение и настоящую прозрачность. Запустите из Xterm `xcompmgr xcompmgr -c` Эта команда позволит вам увидеть эффекты от запущенного xcompmgr, если они вам не понравятся, то CTRL +C в консоли позволит вам убить процесс, а если они вам нравятся, то надо убить процесс и запустить его в фоновом режиме (сзади консоли): `xcompmgr -c &` Примечание: символ & позволяет запустить xcompmgr в фоновом режиме, но при этом если убить сам терминал - процесс тоже сдохнет, чтобы этого не произошло можно использовать утилиту nohup: `nohup xcompmgr -c &` При этом весь вывод будет направлен в nohup.out

Таким образом xcompmgr включит "мягкие" тени. Если будет использован флаг -s то будут "грубые" тени.

Рабочий пример для xcompmgr с тенью и затенением `xcompmgr -cCfF -r7 -o.65 -l-10 -t-8 -D7 &`

Теперь о прозрачности. Выполните в консоли transset Теперь ваш курсор изменился, вы можете кликнуть на окне которое хотите сделать прозрачным. Если запустить transset с цифровым значением от 0 до 1 то это укажет насколько прозрачным должно быть окно

0 - полностью прозрачно

1 - не прозрачно

Попробуйте выполнить transset 0.2

Для возвращения окна назад просто выполните transset и кликните по прозрачному окну.

Как это выглядит

Изображение взято на <http://www.grebowiec.net/archives/xorg4.html>

Рабочий стол Sheridan'a

Выключение эффектов

Для выключения всех эффектов выполните `killall xcompmgr`

Примечание

NVIDIA

При использовании NVIDIA видео карточек рекомендуется добавить в `/etc/X11/xorg.conf` нижеприведенные строки для того, чтобы можно было использовать расширение "Composite" и увеличить производительность при его использовании. Файл: `/etc/X11/xorg.conf`

Section "Device"

...

Option "RenderAccel" "true"

Option "AllowGLXWithComposite" "true"

...

EndSection

**ВАЖНО:** При использовании `RenderAccel` у вас могут возникнуть проблемы со стабильностью, так как эта опция экспериментальная и использовать ее вам предлагается на свой страх и риск.

При использовании `AllowGLXWithComposite` у вас могут возникнуть проблемы с `opengl` приложениями. Могут появляться артефакты, так как `glx` и `composite` не рекомендуется использовать одновременно.

## ATI

При использовании ATI видео карточек совместно с ATI драйверами, дополните /etc/X11/xorg.conf опцией backingstore, для включения поддержки backing store, механизма с помощью которого пиксельные данные закрытого окна запоминаются X сервером и снимает необходимость посылки сообщений к X клиенту когда надо обновить изображение Файл: /etc/X11/xorg.conf

Section "Device"

...

Option "backingstore" "true"

...

EndSection

## Заключение

Эти разработки находятся в самой начальной стадии и мы можем использовать их уже в X.org версии 6.8. По мере развития WM и приложений будет возможность воспользоваться новым функционалом X.org в полную меру без установки дополнительных приложений. Некоторые программы, kwin к примеру, уже портируются.

## Xorg. Переключение раскладок

Переключение USA/RUS + Scroll Lock Led

В секции InputDevice для клавиатуры надо дописать следующее:

Section "InputDevice"

Identifier "Keyboard1"

Driver "kbd"

Option "AutoRepeat" "500 30"

Option "XkbModel" "pc105"

Option "XkbLayout" "us,ru(winkeys)"

Option "XkbOptions" "grp:alt\_shift\_toggle, grp\_led:scroll"

EndSection

Раскладки будут переключаться по Alt+Shift, с индикацией на Scroll Lock Led

Если, конечно, не забыть, что в Section "ServerLayout" должно быть прописано InputDevice "Keyboard1". По умолчанию там как раз Keyboard0, так что будьте внимательны.

а если вы захотите что-бы CTRL-C CTRL-V работали во всех раскладках так-же четко как в микрософт виндовс - то этого увы вам никто не расскажет :) - надо править запутанные конфиги xkb

Возможно, вариант русской раскладки winkeys у вас не заработает. Тогда измените следующую строку:

```
Option "XkbLayout" "us,ru"
```

и добавьте

```
Option "XkbVariant" ",winkeys"
```

Обратите внимание на запятую, она указывает на то, что вариант "winkeys" применяется именно к русской раскладке!

## Установка Xgl

Введение

Проект Xgl позволяет ускорять посредством подсистемы OpenGL вывод примитивов X сервера, а также обеспечивать ускорение composite и render. На данный момент мы можем получить быструю работу с прозрачностью, тенью и отрисовкой окон.

Более подробное руководство на английском языке размещено по адресу en:HOWTO XGL

Инсталяция

Xorg

Нам понадобится последняя версия Xorg. Необходимо размаскировать нужные пакеты и установить Xorg-7.0emerge xorg-x11 Примечание: Дополнительно можно почитать статью Настройка Portage

glitz

Так же необходимо скачать из CVS последнюю версию glitz. cvs -d:pserver:anoncvs@cvs.freedesktop.org:/cvs/cairo co glitz Установить ее.autogen.sh && ./configure --prefix=/usr/ && make && make install

## glxcompmgr

Для работы с расширением composite скачаем утилиту glxcompmgr cvs -d :pserver:anoncvs@cvs.freedesktop.org:/cvs/xorg co app/glxcompmgr Скомпилируем и установимautogen.sh && ./configure --prefix=/usr/ && make && make install

## metacity

Так уж сложилось, что пока glxcompmgr может себя показать во всей красе только с оконным менеджером metacity, да, он способен работать и с kwin, но при этом запуск kdesktop вызовет прекращение нормальной работы glxcompmgr. metacity

## MesaLib

Если мы хотим использовать composite, необходимо собрать библиотеку libGL.so.1.2 из пакета MesaLib-6.4.1 с патчем mesa-glx-x11-render-texture-3.diff, который лежит в каталоге glxcompmgr.patch -p0 <mesa-glx-x11-render-texture-3.diff && ./configure --prefix=/usr/ && make Нет необходимости переустанавливать весь пакет т.к. нас интересует только файл ./lib/libGL.so.1.2. Заменим старый файл.cp ./lib/libGL.so.1.2 /usr/lib/opengl/xorg-x11/lib/

## Xgl

Теперь скачаем. cvs -d:pserver:anoncvs@cvs.freedesktop.org:/cvs/xorg login

CVS password: нажмите Enter

cvs -d:pserver:anoncvs@cvs.freedesktop.org:/cvs/xorg co -r xgl-0-0-1 xserver

Код находится в ./hw/xgl

И установим сам Xgl сервер. ./configure --prefix=/usr/ --enable-xglserver --enable-glx --enable-xkb && make && make install

Запуск сервера для видеокарт nvidia

"Голый" сервер



Убедимся, что мы используем libGL.so.1.2 от nvidiaeselect opengl set nvidia Для запуска сервера в отдельном окне откройте свой любимый терминал и выполните команду `Xgl :1 -ac -accel xv -accel glx:pbuffer -screen 800x600 &` Если сервер не запускается, то, возможно, необходимо отключить расширение xkb, добавив опцию `-kb Xgl :1 -kb -ac -accel xv -accel glx:pbuffer -screen 800x600 &` Также могут возникнуть проблемы с шрифтами, тогда путь к ним надо указать вручную `Xgl :1 -kb -ac -accel xv -accel glx:pbuffer -fp /usr/share/fonts/misc/ -screen 800x600 &` Для запуска сервера в полноэкранном режиме можно использовать опцию `-fullscreenXgl :1 -kb -ac -accel xv -accel glx:pbuffer -fp /usr/share/fonts/misc/ -fullscreen &`

Для запуска на ATI картах с установленными драйверами fglrx необходимо вместо ... `-accel xv...` Использовать ... `-accel xv:pbuffer...`

## Запускаем glxcompmgr

Сначала для примера запустим xterm и metacity из терминала. `export DISPLAY=:1xterm &metacity &` После этого необходимо переключиться на использование библиотеки libGL.so.1.2 из пакета MesaLibeselect opengl set xorg-x11 После этого можно запускать `glxcompmgr.glxcompmgr shadow wobbly &` После этого уже должен быть виден интересный эффект на запущенном xterm'e.

## KDE, GNOME

Для запуска kde или Gnome необходимо запустить сначала metacity затем kde(Gnome) и в последнюю очередь glxcompmgr.

## Возможные проблемы

Возможно у вас не получится запустить сервер т.к. файл `/usr/lib/libGL.so.1` будет указывать не на `/usr/lib/opengl/nvidia/lib/libGL.so.1.2`, а на `/usr/lib/libnvidia-tls.so.1`. Поменяйте симлинк и сервер запустится.

Для переключения раскладки подойдет программа xruskb.

## XGL

### Введение

Xgl это X сервер, который использует OpenGL для прорисовки объектов. Вместе с композитным оконным менеджерами Compiz или новой версией Metacity позволяет получить превосходное качество 3D и 2D эффектов рабочего стола. Xgl базируется на расширении Mesa, `GLX_EXT_texture_from_pixmap`, который представлен только в версиях Mesa 6.5 и старше.

На the XOrg mailing list предлагается написать спецификацию GLX\_EXT\_texture\_from\_pixmap расширения совместно с NVIDIA. Бета версия драйверов уже выпущенных NVIDIA, версия 9625, поддерживают это расширение, это означает что Xgl уже не нуждается в использовании Compiz на NVIDIA картах, кроме этого использование данной конфигурации приветствуется. Для информации об этом методе использования Xgl, пожалуйста ознакомьтесь на en:HOWTO nVidia GL Desktop Effects

Кроме того, Compiz тесно связан с GNOME-компонентами, Xgl и Compiz работают прекрасно во всех окружениях рабочих столов (KDE/GNOME/Xfce/\*box). Если Вы всё же не используете GNOME, будьте готовы к установке необходимых библиотек и утилит. Предупреждение: Данное программное обеспечение ещё не является стабильным, и хотя оно почти всегда работает, всё же используйте его с осторожностью.

## Системные требования

см. the article on video card support under Xgl.

Установка Xgl Примечание: Установка должна быть произведена на системе с работающей X сессией используя эмулятор терминала типа konsole или gterm, выполнение этого условия необходимо для запуска gconf-editor или gset-compiz для конфигурации Compiz в конце установки.

## Требования к программному обеспечению

Обновите Ваше локальное дерево портажей.

`emerge --sync`

Вам необходима модульная версия XOrg (версия 7.x). Для установки модульной версии XOrg ознакомьтесь с официальной Gentoo документацией или соответствующей статьёй wiki.

en:HOWTO Modular Xorg

Официальное руководство

Модульный XOrg (версия 7.x) был опубликован в стабильной ветке дерева портажей, поэтому Вы можете пропустить эту главу если Ваш Gentoo полностью обновлен.

Compiz использует рабочую оболочку GNOME GConf для конфигурации, и некоторые из опций конфигурации будут приведены здесь. Наиболее легким и

популярным способом конфигурации является программа gconf-editor, одна из приложений GNOME. Конечно, и менее известная KDE программа KConfigEditor также может использоваться для конфигурации. Правда для этой программы пока не существует ebuild файла, но Вы можете загрузить исходный код этой программы с [extragear.kde.org](http://extragear.kde.org). Пожалуй самый нежелательный из способов это редактирование gconf конфигурационных файлов вручную, что является скучным и неблагодарным занятием.

Если Вы хотите использовать gconf-editor, то его необходимо установить.

```
emerge -nav gconf-editor
```

Также Вам необходимо установить Subversion с включенной поддержкой webdav. Вы должны отключить флаг nowebdav. Добавьте добавьте при необходимости следующую строку в /etc/portage/package.use: Примечание: Тут используется двойное отрицание в использовании USE флага.

```
echo "dev-util/subversion -nowebdav" >> /etc/portage/package.use
```

затем,

```
emerge -nav subversion
```

### Загрузка Portage оверлея

Существуют два метода для создания требуемого оверлея; оверлей можно загрузить используя svn утилиту или использовать layman утилиту которая будет управлять оверлеями вместо Вас. Используя layman производить обновление системы проще, так что решать Вам. Вне зависимости, что Вы будете использовать результат будет одним и тем же.

Метод по умолчанию: использование Layman утилиты

Если вы ещё не установили Layman, сделайте это сейчас.

```
emerge -nav layman
```

Также Вам необходимо добавить следующую строку в Ваш make.conf.

```
echo "source /usr/portage/local/layman/make.conf" >>/etc/make.conf
```

Если Вы используете Eix Caching System, удостоверьтесь что Вы используете самую новую версию; старые версии не поддерживают чтение внешних файлов make.conf.

Измените "nocheck" переменную на значение "yes" в конфигурационном файле утилиты layman. (/etc/layman/layman.cfg)

Затем добавьте оверлей с помощью Layman утилиты:

```
layman -f  
layman -a xeffects
```

если вы используете layman версии 1.0.7 или позже, то оверлеи с неправильными атрибутами будут проигнорированы. Используйте опцию -k для обхода этой ситуации.

```
layman -k -a xeffects
```

Альтернативный метод: Создание оверлея вручную

Если Вы хотите обслуживать Ваш оверлей самостоятельно, то сделайте следующее. Иначе (если используется layman), вы можете перейти к главе Размаскирование необходимых пакетов. Примечание: Лучше всего не смешивать оверлеи и использовать их отдельно друг от друга.

Перейдите в Вашу директорию оверлеев

```
cd /usr/local/overlays
```

Загрузите последнюю версию из CoffeeBuzz's Subversion репозитория в директорию 'xgl-coffee'

```
svn co http://svn.xgl-coffee.org/xgl-coffee/trunk xgl-coffee
```

Теперь Вы создали оверлей дерева портежей в /usr/local/overlays/xgl-coffee.

Добавьте следующий путь в переменную PORTDIR\_OVERLAY в /etc/make.conf. Файл: /etc/make.conf

```
PORTDIR_OVERLAY="${PORTDIR_OVERLAY} /usr/local/overlays/xgl-coffee"
```

## Размаскирование необходимых пакетов

Много пакетов маркированы как нестабильные и должны быть размаскированы чтобы их можно было установить из дерева портажей. Добавьте следующие строки к файлу `/etc/portage/package.keywords` или `/etc/portage/package.unmask`.  
Файл: `/etc/portage/package.keywords`

# Зависимости

dev-util/git

media-libs/glitz

media-libs/mesa

x11-apps/mesa-progs

sys-apps/man

x11-libs/cairo

dev-python/pycairo

x11-libs/qt

x11-misc/util-macros

x11-libs/glpk

x11-apps/xvinfo

x11-apps/xlsclients

x11-libs/libwnck

x11-misc/xwinwrap

virtual/xft

gnome-base/gconf

gnome-base/libgnomeui

x11-libs/gtk+

dev-libs/glib

x11-libs/libdrm

#XGL пакет

x11-base/xgl

# старые версии Compiz

x11-wm/compiz-quinnsstorm

x11-misc/compiz-quinnsstorm-plugins

x11-wm/compiz

x11-wm/cgwd

```
x11-misc/csm
x11-misc/cgwd-themes
x11-misc/cgwd-themes-extra
x11-misc/gset-compiz
x11-misc/compiz-manager
```

# новая ветка Beryl для Compiz

```
x11-wm/beryl-core
x11-plugins/beryl-plugins
x11-misc/beryl-manager
x11-misc/beryl-settings
x11-wm/emerald
x11-misc/emerald-themes
```

Удостоверьтесь что флаги glitz, pdf и png включены в переменную USE для Cairo. Возможно Вам понадобится добавить следующую строку к файлу package.use: Файл: /etc/portage/package.use

```
x11-libs/cairo glitz pdf png
```

## Установка Xgl

Новые бета драйверы с закрытым исходным кодом от nVidia (версия 1.0.9625) уже включают в себя поддержку XGL. Поэтому не нужно более никаких установок. Если вы не хотите использовать новейшие бета драйвера, то следуйте к главе "не-nVidia пользователи". Данный способ работает на nVidia картах при этом установка и использование новых nVidia драйверов намного проще. Если Вы хотите использовать новейшие бета драйвера тогда Вы должны перейти к главе "nVidia Пользователи".

## Не-nVidia пользователи

Во-первых Вы должны установить Cairo. Используйте опцию --oneshot чтобы не записывать этот пакет в список world (не захламляй свой world список).

```
# emerge --oneshot --ask --verbose cairo
```

Переинсталируй rangod и gtk+ с использованием новоустановленного Cairo пакета.

```
# emerge --oneshot --ask --verbose pango gtk+
```

Далее у Вас есть выбор между beryl и compiz ветками:

compiz это оригинальное приложение написанное Давидом Равеманом (David Reveman) из Новел (Novell). Это приложение уже включено в дерево портажей.

beryl это снимок Quinnstorm's Ubuntu кодовой базы с некоторыми экспериментальными плагинами. Формально эта ветка известна под названием compiz-quinnstorm. Использует emerald как window decorator. это отдельный пакет не включенный в дерево портажей.

Проверьте что флаги dbus и svg включены в переменную USE для Compiz. Возможно Вам потребуется добавить следующую строку в package.use:  
Файл: /etc/portage/package.use

```
x11-wm/compiz dbus svg
```

Проверьте что пакеты glproto и libdrm установлены. Установите их если это требуется:

```
# emerge --ask --verbose --oneshot --noreplace glproto libdrm
```

В конечном итоге Вы должны будете установить Xgl и Mesa, которые автоматически скомпилируют и установят Compiz а также все его плагины и Xgl сервер.

```
# emerge --ask --verbose --oneshot --noreplace mesa
```

Опция 1: Для использования vanilla compiz:

```
# emerge --ask --verbose xgl compiz
```

Опция 2: Для использования beryl мета пакета:

```
# emerge --ask --verbose xgl beryl-core
```

Примечание: Установка beryl также устанавливает декоратор окон emerald и его темы. Таким образом emerald не надо устанавливать отдельно.

Примечание: NVIDIA пользователи пожалуйста переустановите nvidia-drivers после установки Xgl

Примечание: Если не удастся скомпилировать xgl, проверьте заплатку на

<http://forums.xgl-coffee.org/viewtopic.php?t=178>

Примечание: Если не удается скомпилировать beryl из за dbus, попробуйте обновить пакет dbus

Примечание: Если не удается скомпилировать xgl и компиляция обрывается на fbmmx.c или fbedge.c, попробуйте удалить флаг "-fforce-addr" из переменной CFLAGS. Также ознакомьтесь с документацией на форуме

Если всё прошло удачно, то Cairo должен присутствовать в системе как зависимость. Примечание: Для следующей проверки необходим установленный пакет gentoolkit.

Проверте с помощью следующей команды

```
# equery d -o -p cairo
```

Предупреждение: Если команда не возвращает никаких пакетов, то значит чтото прошло не правильно!

После произведенных действий Вы должны получить функционирующую установку Xgl. Пожалуйста ознакомьтесь с Глава работающий Xgl.

nVidia пользователи

nVidia драйвер версии 9625 BETA и старше содержат XGL и AIGLX-подобные расширения и могут быть использованы совместно с другими композитными менеджерами окон. см. en:HOWTO nVidia GL Desktop Effects для более детальной информации.

Обновление Xgl

Из-за быстрой разработки Xgl и Compiz, оверлей обновляется как минимум один раз в неделю. Используйте один из приведённых методов для обновления оверлея.

Обновление с помощью Layman утилиты

Обновление оверлея с помощью Layman утилиты очень просто:

```
layman -s xeffects
```



## Обновление вручную

Для обновления оверлея вручную, перейдите в директорию оверлея и введите `./svnup.sh`. Этот скрипт также выведет список изменений.

```
cd /usr/local/overlays/xgl-coffee/  
./svnup.sh
```

## Запуск Xgl

## Конфигурация Compiz

Для начала надо сконфигурировать Compiz и его плагинов. Если никакие плагины не указаны, тогда окна останутся без декораций! (не будет кнопок `minimize/maximize/close`, невозможно будет изменять размеры окна, ничего вообще толком не будет.)

Существуют два пути указания плагинов. Вы можете указать плагины в командной строке или использовать оконный редактор конфигурации например с помощью программ `gconf-editor` или `gset-compiz`. Но вы не можете использовать оба способа одновременно! Если `gconf` указан в командной строке, тогда Compiz проигнорирует все другие плагины указанные в командной строке и возьмёт список плагинов из вашей конфигурации `gconf`.

Таким образом если Вы впервые запускаете Xgl и Вы хотите конфигурировать в оконном редакторе, наступило время сделать это, до того как Вы выйдете из Вашей X сессии.

Способ "По умолчанию": Использование `gconf-editor`

```
# gconf-editor
```

Перейдите в `apps --> compiz --> general --> allscreens --> options` и отредактируйте узел `active_plugins` находящийся справа. Примечание: Если в `gconf-editor` нет вкладок Compiz, тогда это можно исправить таким образом:

```
# export GCONF_CONFIG_SOURCE="xml:merged:/etc/gconf/gconf.xml.defaults"  
# gconftool-2 --makefile-install-rule /etc/gconf/schemas/compiz.schemas
```

Способ "По умолчанию" для `compiz-quinnstorm`: Использование `csm`

```
# csm
```

Плагины могут быть активированы кликая в checkbox после имени. Индивидуальная конфигурация плагинов осуществляется после щелчка мышью на имени плагина. Примечание: Compiz-quinnstorm поставляется вместе с Compiz Manager (compiz-manager). Это оконное приложение находящееся в системной панели и включает в себя Compiz Settings Manager (csm) и Compiz Theme Manager (gcompizthemer). Запустив на выполнение эти две утилиты (управляйте compiz и темами), перестаруйте оконный менеджер или декоратор из системной панели и установите запасной оконный менеджер который будет использоваться в случае неудачного старта.

```
# compiz-manager  
or (if any dbus problem) :  
# dbus-launch compiz-manager
```

Альтернативный способ: Использование gconftool-2

```
# gconftool-2 -s /apps/compiz/general/allscreens/options/active_plugins  
"[gconf,decoration,wobbly,fade,minimize,cube,switcher,move,resize,place,rotate,zoom,scale]" -t list --list-type=string
```

Укажите активные плагины внутри квадратных скобок.

Альтернативный способ: Использование gset-compiz Предупреждение:

gset-compiz больше не пытается использовать все возможности Compiz. Используйте лучше gconf-editor или более новые программы для возможностей которыми не располагает gset-compiz.

```
# gset-compiz
```

Плагин можно активировать поставив галочку сразу возле его имени. Конфигурация индивидуальных настроек плагина доступна после нажатия кнопки Plugins находящейся сразу под меню.

Как минимум, потребуются пакеты gconf и decoration. Также, если Вы хотите включить все доступные эффекты, надо также добавить wobbly, fade, switcher, move, resize, place, minimize, cube, rotate, zoom, и scale. Из имени пакета Вы сами можете догадаться, что выполняет каждый из плагинов, так что не торопитесь включать их все. Это как предпочтения "сладостей для глаз", установите плагины move и resize - у Вас появится возможность работать с их функциями.

Using Quinnstorm's compiz & cgwd

Quinnstorm's new compiz doesn't use gconf plugin anymore (so settings). To use it, replace any instance of compiz --replace gconf found in this document with dbus-launch compiz --replace dbus csm. And set it by csm or dbus-launch csm.

Quinnstorm's new window decorator must be called instead of gnome-window-decorator in order to function. If you chose to emerge cgwd, just replace any instance of gnome-window-decorator found in this document with dbus-launch cgwd.

## Оконный режим

Запустив Xgl таким способом мы создадим новое окно с включенным Xgl сервером. Это отличный способ протестировать правильность установки прежде чем запускать XGL в полноэкранном режиме.

Чтобы попробовать его работу в оконном режиме при уже запущенном X сервере:  
Code: Для ati пользователей

```
Xgl :1 -ac -accel glx:pbuffer -accel xv:pbuffer  
-fp /usr/share/fonts/misc,/usr/share/fonts/other_fonts
```

Code: Для nVidia пользователей

```
Xgl :1 -ac -accel glx:pbuffer -accel xv -fp /usr/share/fonts/misc,/usr/share/fonts/other_fonts
```

Примечание: Обсуждение использования pbuffer или fbo как аргументов accel, можно почитать здесь <http://forums.gentoo.org/viewtopic-t-455153-highlight-.html>

Примечание: Xgl не будет считывать значение FontPath из xorg.conf, по-этому, лучше в конце строки после опции -fp прописать пути к шрифтам разделяя их запятой. Используйте этот скрипт для получения путей к шрифтам прописанным в файле xorg.conf:

```
grep -i fontpath /etc/X11/xorg.conf | egrep -v "[:space:]*#" | sed "s/\.\\+\"(.\\+\\)\"/1,/g" | xargs  
echo | sed "s/ //g" | sed "s/,\\$/"
```

Протестируйте Вашу конфигурацию:

Запустите композитный оконный менеджер (compiz or compiz-quinnstorm), запустите декоратор окон (gnome-window-decorator or cgwd) и запустите приложение (xterm).  
Code: Для большинства: Если Вы используете compiz совместно с gnome-window-decorator

Учтите что название gnome-window-decorator изменилось на gtk-window-decorator 22 Сентября 2006 года.

```
LD_LIBRARY_PATH=/usr/lib/opengl/xorg-x11/lib/ \
```

```
DISPLAY=:1 compiz gconf
```

```
DISPLAY=:1 gnome-window-decorator
```

```
DISPLAY=:1 xterm
```

Code: Альтернатива 1: если Вы используете compiz-quinnsorm совместно с gnome-window-decorator

```
LD_LIBRARY_PATH=/usr/lib/opengl/xorg-x11/lib/ \
```

```
DISPLAY=:1 dbus-launch compiz dbus csm
```

```
DISPLAY=:1 gnome-window-decorator
```

```
DISPLAY=:1 xterm
```

Code: Альтернатива 2: Если вы используете compiz-quinnsorm с cgwd

```
LD_LIBRARY_PATH=/usr/lib/opengl/xorg-x11/lib/ \
```

```
DISPLAY=:1 dbus-launch compiz dbus csm
```

```
DISPLAY=:1 dbus-launch cgwd
```

```
DISPLAY=:1 xterm
```

Если всё прошло успешно, мы можем начать конфигурировать Вашу систему для запуска Xgl с Вашим графическим менеджером аутентификации. Следуйте инструкциям которые помогут Вам настроить менеджер аутентификации. Это последний шаг необходим для удобного использования Xgl.

startx (Для пользователей startx)

Если вы обычно загружаетесь в консоль и Xorg сервер запускается через startx, а .xinitrc загружает ваш менеджер окон на DISPLAY:0, то вам нужно: написать скрипт startxgl для запуска сервера Xgl, а так же файл .xglnitrc для загрузки менеджера окон на DISPLAY:1. Эти действия не изменят вашей текущей конфигурации и вы сможете в любой момент вернуться в нормальный Xorg сервер.

1) Скопируйте startx (cp /usr/bin/startx /usr/local/bin/startxgl) и отредактируйте 2 строки:  
Файл: /usr/local/bin/startxgl

```
## userclientrc=$HOME/.xinitrc
```

```
userclientrc=$HOME/.xglnitrc
```

```
## xinit $clientargs -- $serverargs -deferglyphs 16 &
```

```
# ATI
```

```
xinit $clientargs -- /usr/bin/Xgl :1 $serverargs -ac -accel xv -accel glx:pbuffer -deferglyphs 16 &
```

```
# NVIDIA
```

```
xinit $clientargs -- /usr/bin/Xgl :1 $serverargs -ac -accel xv -accel glx:fbo -deferlyphs 16
&
```

2) создайте файл ~/.xglnitrc (vi ~/.xglnitrc) Файл: ~/.xglnitrc

```
sleep 2 # give Xgl some extra time to start
```

```
DISPLAY=:1 KDEWM=compiz-decorator startkde
```

Примечание: Пользователи Gnome должны вписать gnome-session вместо startkde и WINDOW\_MANAGER вместо KDEWM. Пользователи других DM вписывают соответствующие им команды запуска. Для уточнения проверьте ваш ~/.xinitrc.

3) создайте файл compiz-decorator (vi /usr/local/bin/compiz-decorator)  
Файл: /usr/local/bin/compiz-decorator

```
## Start compiz or compiz-quinnstorm
```

```
## compiz --replace gconf &
```

```
## dbus-launch compiz --replace dbus csm &
```

```
compiz --replace gconf &
```

```
sleep 2
```

```
## Start gnome-window-decorator or cgwd
```

```
## gnome-window-decorator &
```

```
## dbus-launch cgwd &
```

```
gnome-window-decorator &
```

Примечание: Команды для запуска и использования compiz-quinnstorm и cgwd в скрипте закоментированны. Если вы хотите пользоваться ими то не забудьте закоментировать текущие

Смените права доступа на /usr/local/bin/compiz-decorator что бы сделать его исполняемым:

```
# chmod +x /usr/local/bin/compiz-decorator
```

4) Для запуска вашего оконного менеджера в XOrg используйте startx, для запуска в Xgl используйте startxgl.

Скрипты тестировались на связке ATI и KDE. --Ash 17:48, 11 Сентября 2006 (UTC)

XDM (Базовая графическая система аутентификации)

Измените ссылку на X сервер: Файл: /etc/X11/xdm/Xservers  
:0 local /usr/bin/Xgl vt7

KDM (для пользователей KDM)

Замените строку с ServerCmd в kdmrc на одну из следующих: Файл: /usr/kde/<your version>/share/config/kdm/kdmrc

# NVidia Command

ServerCmd=/usr/bin/Xgl -br -ac -accel glx:pbuffer -accel xv

# ATI Command

ServerCmd=/usr/bin/Xgl -br -ac -accel glx:pbuffer -accel xv:pbuffer

Вы также должны поменять значение ServerTimeout так как Xgl запускается немного дольше, чем нормальный X сервер: Файл: /usr/kde/<your version>/share/config/kdm/kdmrc

ServerTimeout=30

Если kdm не смог загрузится, измените значение ServerTimeout на 600. Учтите, что в этом случае время загрузки Xgl значительно увеличится.

Не используйте скрипт /usr/kde/3.5/share/config/kdm/Xstartup для запуска Compiz. Для его выполнения нужны права пользователя root.

Скопируйте скрипт в другое место, например сюда: Файл: /usr/local/bin/compiz-decorator

## Start compiz or compiz-quinnstorm

## compiz --replace gconf &

## dbus-launch compiz --replace dbus csm &

compiz --replace gconf &

sleep 2

## Start gnome-window-decorator or cgwd

```
## gnome-window-decorator &  
## dbus-launch cgwd --replace &  
gnome-window-decorator --replace &
```

Примечание: Команды для запуска и использования compiz-quinnstorm и cgwd в скрипте закомментированы. Если вы хотите пользоваться ими то не забудьте закомментировать текущие

Сделайте вновь созданный скрипт /usr/local/bin/compiz-decorator исполняемым:  
# chmod +x /usr/local/bin/compiz-decorator

Далее измените: Файл: /etc/env.d/99kde-env  
KDEWM=compiz-decorator

И наконец, выполните команду:  
# env-update

Теперь после перезагрузки или выполнения команды /etc/init.d/xdm start вы запустите полнофункциональный Xgl со всеми эффектами без пользовательского вмешательства и аутентификацией с помощью KDM.

GDM (для пользователей GNOME)

Вы должны сделать 3 вещи, чтобы заставить Xgl работать с GDM и использовать Compiz для автоматической загрузки при авторизации пользователя.

Сконфигурируйте GDM для Xgl с отложенным запуском, чтобы при запуске нельзя было сказать, что она не работает

Сконфигурируйте GDM для использования с Xgl-сервером

Установите сессию GNOME для автоматического запуска из Compiz Примечание: Конфигурирование gdm для отложенного запуска Xgl доступно с gdm-2.14.1 и поздних версий. Если Вы используете раннюю версию, начните с шага 2, но, возможно, Вам не удастся заставить gdm работать с Xgl. Также использование gdmsetup для конфигурирования gdm на использовании Xgl-сервера (Раздел 2a) может быть недоступно для версий gdm ранее 2.14.0. Если недоступны необходимые Вам функции, самостоятельно отредактируйте конфигурационный

файл GDM (Раздел 2b).

## Конфигурирование Gdm на отложенный запуск для запуска Xgl

Gdm сконфигурирована по умолчанию на завершение (kill) X-сервера, если загрузка заняла более 10 секунд. К сожалению, новые версии Xgl могут загружаться дольше. Пользователи Xgl в Gentoo, могут сконфигурировать Gdm (пример по версии 2.14.1) на ожидание более 10 секунд.

```
# emerge --ask --verbose >=gdm-2.14.1
```

Добавьте/измените следующие команды секции [daemon]  
файла /etc/X11/gdm/custom.conf. Строка: Файл: /etc/X11/gdm/custom.conf

```
GdmXserverTimeout=30
```

Если /etc/X11/gdm/custom.conf не существует, попробуйте /etc/X11/gdm/gdm.conf-custom или /etc/X11/gdm/gdm.conf.

## Конфигурирование Gdm для использования сервера Xgl

### Использование gdmsetup

Для начала сделайте резервную копию файла gdm.conf

```
cp /etc/X11/gdm/gdm.conf /etc/X11/gdm/gdm.conf.bak
```

Запустите gdmsetup

Выберите раздел Security

Щёлкните кнопку Configure X Server, расположенную снизу-справа раздела

Для начала мы удалим значения для стандартного X-сервера

Нажмите по стандартному серверу, который находится в списке Servers to Start

Нажмите Remove Button

Сейчас мы создадим значения для сервера Xgl

Нажмите кнопку Add/Modify

Задайте использование нового сервера VT 1. Используйте Standard для Server опцию — она отображает, какой профиль использовать для сервера, в идеале мы должны создать новый "Xgl"-профиль, а в настоящее время при помощи gdmsetup это сделать невозможно, так что отредактируем настройки для профиля Standard. Не вводите здесь каких-либо опций. Нажмите Ok по завершении.

В Server Settings измените Server Name на "Xgl"

Введите команду Xgl в поле Command, например, для пользователей ATI:

```
/usr/bin/Xgl :1 -ac -accel glx:pbuffer -accel xv:pbuffer
```



NVIDIA users should use:

```
/usr/bin/Xgl -br -ac -accel glx:pbuffer -accel xv
```

Измените в выпадающем списке Launch значение на Greeter

Установите Logins are handled by this computer

Установите Flexible (on demand)

Save и Close

Перезапустите gdm

Ручная настройка

Для начала сделайте резервную копию файла gdm.conf

```
cp /etc/X11/gdm/gdm.conf /etc/X11/gdm/gdm.conf.bak
```

Now open /etc/X11/gdm/gdm.conf in a text editor and search for the [servers] section. This should be near the end of the file.

Comment out 0=Standard and add a new line that reads 0=inactive and one that reads 1=Xgl.

Add the server information for the Xgl server, remember to substitute the correct Xgl server command! The one below is for ATI cards.

The modified /etc/X11/gdm/gdm.conf:

```
[servers]
```

```
#0=Standard
```

```
1=Xgl
```

# Definition of the xgl X server.

```
[server-Xgl]
```

```
name=Xgl
```

```
command=/usr/bin/Xgl :1 -ac -accel xv:pbuffer -accel glx:pbuffer
```

```
flexible=true
```

```
chooser=false
```

```
handled=true
```

```
priority=0
```

Restart gdm Примечание: For i810 users (not necessarily i915; this would kill my xorg sessions, I had to use the ATI command), the command for the above should be:

```
[server-xgl]
```

```
name=Xgl server
```

```
command=/usr/bin/Xgl -accel xv -accel glx:pbuffer -accel xv:pbuffer -ac -audit 0 -br -dpms -  
dpi 72
```

```
flexible=true
```

Via xsession

You can add another session for Xgl on display :1 (leaving the standard server on :0). It is a 2 step setup and imho the best option, first create a file to start Xgl and second create a new xsession file.

Create a new file startxgl.sh in /usr/bin which starts Xgl.

```
vim /usr/bin/startxgl.sh
```

And add one of the four options below.

NVIDIA (using GNOME)

```
#!/bin/bash
```

```
Xgl -fullscreen :1 -audit 0 -ac -br -accel glx:pbuffer -accel xv:fbo &  
sleep 2 && DISPLAY=:1 gnome-session
```

NVIDIA (using KDE)

```
#!/bin/bash
```

```
Xgl -fullscreen :1 -audit 0 -ac -br -accel glx:pbuffer -accel xv:fbo &  
sleep 2 && DISPLAY=:1 startkde
```

ATI and Intel (using GNOME)

```
#!/bin/bash
```

```
Xgl -fullscreen :1 -audit 0 -ac -br -accel glx:pbuffer -accel xv:pbuffer &  
sleep 2 && DISPLAY=:1 gnome-session
```

ATI and Intel (using KDE)

```
#!/bin/bash
```

```
Xgl -fullscreen :1 -audit 0 -ac -br -accel glx:pbuffer -accel xv:pbuffer &  
sleep 2 && DISPLAY=:1 startkde
```

Create a new xsession file: Файл: /usr/share/xsessions/xgl.desktop

[Desktop Entry]

Encoding=UTF-8

Name=Xgl

Comment=Start an Xgl Session

Exec=/usr/bin/startxgl.sh

Icon=

Type=Application

Now you will be able to select a new session via gdm's option menu. Options -> Select Session

The benefit is that it will not replace your normal XOrg server. So when you "break" your Xgl you can return to the normal XOrg server.

Adding Compiz to your GNOME session (replacing Metacity)

Here is a little startscript, which I've placed in /usr/bin/. Useful if you do not always run Xgl. Change or remove the setxkbmap line according to your keyboard layout.  
Файл: /usr/bin/compizrc

```
#!/bin/bash
#
# Start compiz within gnome-session
#
if [ `ps -A -o comm | grep -c '^Xgl$'` == "1" ]; then
    DISPLAY=:1 LD_LIBRARY_PATH=/usr/lib/openssl/xorg-x11/lib/ compiz --replace
gconf &
    DISPLAY=:1 gnome-window-decorator &
#    DISPLAY=:1 setxkbmap -model pc105 -layout it -variant basic # Change according
to your needs, and uncomment if needed

#    Or like that, if you need to switch between layouts.
#        setxkbmap -model pc105 -layout "us,ru(winkeys)" -variant winkeys -option
"grp:alt_shift_toggle,grp_led:scroll"
else echo "${0}: Error: Compiz and g-w-d not launched. Xgl not running?"
fi
Файл: /usr/bin/compizrc.modified
#!/bin/bash
#
# Start compiz within gnome-session
#
if [ `ps -A -o comm | grep -c '^Xgl$'` == "1" ]; then
# For old versions
#    DISPLAY=:0 LD_LIBRARY_PATH=/usr/lib/openssl/xorg-x11/lib/ compiz --replace
gconf &
# For Compiz-Quinnstorm with csm
    DISPLAY=:0 LD_LIBRARY_PATH=/usr/lib/openssl/xorg-x11/lib/ compiz --replace dbus
csm &
    DISPLAY=:0 dbus-launch cgwd --replace &
```

```
# DISPLAY=:0 setxkbmap -model br-abnt2 -layout br -variant br
else echo "${0}: Error: Compiz and cgwd not launched. Xgl not running?"
fi
```

Code: Make the script executable

```
chmod 755 /usr/bin/compizrc
```

After logging into GNOME, start System->Settings->Sessions. Here you go to the 3rd tab Startup Programs and press the Add button. Enter the path to your Compiz startscript (e.g., /usr/bin/compizrc).

After logging out and back in Compiz will start, but you'll see no decorations and effects. Thats because the Compiz gconf tree for the current user has just been created - without any plugins!

So start your configuration editor and go to apps/compiz/general/allscreens/options and edit the active\_plugins key. This should contain the following items in this order: gconf decoration wobbly fade minimize cube rotate zoom scale move resize place menu switcher

If the key is not there, make it with a type of List

After restarting your Xgl Server should be accessible with GDM & gnome-session.

Xsession (Modifying startx)Примечание: This is for NVIDIA users.

Предупреждение: This currently will not work for ATI users because of the display issues, requiring to be run on DISPLAY=:1.

Примечание: I use simmilar configuration (see startx section) on ATI. Consider /usr/bin/Xgl :1 instead of /usr/bin/Xgl and starting Gnome at :1 as well, if DISPLAY:0 is problematic. --Ash 17:57, 11 September 2006 (UTC)

Copy /usr/bin/startx to /usr/bin/startxgl and change the following line towards the end of the file: Файл: /usr/bin/startxgl

```
xinit $clientargs -- $serverargs -deferglyphs 16 &
```

to

```
xinit $clientargs -- /usr/bin/Xgl $serverargs -ac -accel xv -accel glx:pbuffer -deferglyphs 16
&
```

This does not start Compiz so create a small start script and add it to the gnome-session manager (Once GNOME is up. The first time, you will have to open a terminal and call the script manually.) Файл: /usr/bin/compizrc

```
#!/bin/bash
#
# Start compiz within gnome-session
#
if [ `ps -A | grep Xgl | wc -l` == "1" ]; then
    LD_LIBRARY_PATH=/usr/lib/opengl/xorg-x11/lib/ compiz --replace gconf &
    gnome-window-decorator &
#    setxkbmap -model pc105 -layout it -variant basic # Change according to your needs,
#    and uncomment if needed

#    Or like that, if you need to switch between layouts.
#    setxkbmap -model pc105 -layout "us,ru(winkeys)" -variant winkeys -option
#"grp:alt_shift_toggle,grp_led:scroll"
```

fi

Примечание: This should start the desktop environment configured via XSESSION in /etc/rc.conf

## Xfce4

With xfce4-session

xfce4-session is installed by default as part of the xfce4 meta-package. If you installed xfce4 this way, the following will work.

First, it is necessary to create a script that will start Compiz and gnome-window-decorator with the correct libraries and switches. Файл: /usr/bin/compizrc

```
LD_LIBRARY_PATH=/usr/lib/opengl/xorg-x11/lib compiz --replace gconf &
gnome-window-decorator
```

Code: Make the script executable

```
chmod 755 /usr/bin/compizrc
```

Now, edit the xfce4-session settings to point to this rather than xfwm4. Файл: /etc/xdg/xfce4-session/xfce4-session.rc

[Failsafe Session]

Count=4

Client0\_Command=compizrc

```
Client0_PerScreen=False
Client1_Command=xfce4-panel
Client1_PerScreen=True
Client2_Command=xftaskbar4
Client2_PerScreen=True
Client3_Command=xfdesktop
Client3_PerScreen=False
```

Примечание: Instead of editing system-wide file you may also add above section to ~/.config/xfce4-session/xfce4-session.rc file. Note also that you may need to delete content of the ~/.cache/sessions directory, if you still have xfwm4 rather than Compiz running.

Finally, create a script to start Xgl with Xfce4. Файл: /usr/bin/startxgl for NVIDIA

```
#!/bin/bash
Xgl -ac -accel xv:fbo -accel glx:pbuffer &
xfce4-session
Файл: /usr/bin/startxgl for ATi
#!/bin/bash
Xgl -ac -accel xv:pbuffer -accel glx:pbuffer &
xfce4-session
Code: Make the script executable
chmod 755 /usr/bin/startxgl
```

You will notice that xfce4-panel is now transparent, just like if you were using xfwm4's compositor. The pager will be distorted, but otherwise everything works beautifully. Windows even minimize to the correct spot on the taskbar.

Without xfce4-session

If you compiled a minimal Xfce (without xfce4-session, xffm4, etc.), then you can use this altered startxgl script from the gnome script above:

```
#!/bin/bash
echo ">Starting XGL at Display: $1"
echo "===== XGL ====="
Xgl :$1 -ac -accel xv -accel glx:pbuffer &
sleep 3
```

```
echo "===== COMPIZ ====="
```

```
DISPLAY=: $1 LD_LIBRARY_PATH=/usr/lib/opengl/xorg-x11/lib/ compiz --replace switcher  
decoration wobbly fade minimize cube rotate zoom scale move resize place &
```

```
sleep 3
```

```
echo "===== XFCE ====="
```

```
DISPLAY=: $1 xfce-mcs-manager
```

```
DISPLAY=: $1 gnome-window-decorator &
```

```
DISPLAY=: $1 xftaskbar4 &
```

```
DISPLAY=: $1 xfdesktop &
```

```
DISPLAY=: $1 exec xfce4-panel
```

Again, xfce4-panel will be transparent as if you had the X.org composite extension turned on (but you don't) or if you use xfce4-svn, and the pager will be distorted. Everything else should be fine.

With startxfce4

To do it the old fashioned, or 'normal' way via startx/startxfce4 I did the following. First, startxfce4 shouldn't be called I don't think. just use an .xinitrc in your homedir and startx.  
Файл: ~/.xinitrc

```
#!/bin/sh
```

```
/usr/bin/startxfce4
```

Now as root, we create compizrc, like above. Note however that for me, it worked best/only without the DISPLAY variable. Файл: /usr/bin/compizrc

```
#!/bin/bash
```

```
LD_LIBRARY_PATH=/usr/lib/opengl/xorg-x11/lib/ compiz --replace gconf > ~/compiz.log  
2>&1 &
```

```
sleep 1;
```

```
gnome-window-decorator > ~/gnome-window.log 2>&1 &
```

Obviously the output redirections are optional (but handy for debugging). The sleep 1; might also be removed to speed things up. If things go TO fast however, you might get in trouble though. Don't forget to chmod 755 /usr/bin/compizrc

Lastly we need to use this new compizrc instead of the default xfce4 window manager.

Again, as root edit /etc/xdg/xfce4-session/xfce4-session.rc and replace xfwm4 with

compizrc Файл: /etc/xdg/xfce4-session/xfce4-session.rc (old)

Client0\_Command=xfwm4

Файл: /etc/xdg/xfce4-session/xfce4-session.rc (new)

Client0\_Command=compizrc

Now only one thing is left, start Xgl. As above you can copy startx in /usr/bin/startx to /usr/bin/startxgl and make the following change: Файл: /usr/bin/startxgl (old)

xinit \$clientargs -- \$serverargs -deferglyphs 16 &

Файл: /usr/bin/startxgl (new)

xinit \$clientargs -- /usr/bin/Xgl -ac -accel xv -accel glx:pbuffer \$serverargs -deferglyphs 16 &

Примечание: Todo: Make Client0\_Command depend on whether startx or startxgl is being used as 3D apps don't appear to work as they should and switching back to the 'old way' is required.

Edit: it seems that when using the startxfce4 script, the /etc/xdg/xfce4-session/xfce4-session.rc file (and the .xinitrc) get ignored thus starting a non Xgl desktop. Making 'switchin' not needed. startxfce4 for oldschool, startxgl for new style

Entrance

Emergence latest entrance:

echo "=x11-misc/entrance-0.9.0.007 ~x86" >> /etc/portage/package.keywords

emerge -av entrance

Use the ecore\_config utility to change the xserver string (choose according to your video card):

# ATI

ecore\_config -k "/entranced/xserver" -s "/usr/bin/Xgl -ac -accel xv:pbuffer -accel glx:pbuffer -nolisten tcp" -c /etc/entrance\_config.cfg

# NVIDIA

ecore\_config -k "/entranced/xserver" -s "/usr/bin/Xgl -ac -accel xv:fbo -accel glx:pbuffer -nolisten tcp" -c /etc/entrance\_config.cfg

Restart entrance.

Qingy



It is quite easy to change Qingy to start Xgl instead of a regular X server. All you need to do to make it start Xgl is change 2 lines in /etc/qingy/settings Файл: /etc/qingy/settings

```
...
# x_server = "/usr/X11R6/bin/XFree86"
x_server = "/usr/bin/Xgl"
# x_args = "-nolisten tcp"
x_args = "-ac -accel xv -accel glx:pbuffer -nolisten tcp"
...
```

Now Qingy should start Xgl instead of regular X. To make KDE and GNOME load Compiz you need to export a variable. KDEWM for KDE and WINDOW\_MANAGER for GNOME. It is easiest to do this in /etc/X11/Sessions.

You need to have followed the instructions for adding plugins to gconf-editor. Then make a /usr/bin/compizrc as shown and chmod +x it. Файл: /usr/bin/compizrc

```
#!/bin/bash
LD_LIBRARY_PATH=/usr/lib/opengl/xorg-x11/lib compiz --replace gconf &
sleep 2
gnome-window-decorator &
setxkbmap -model itouch -layout gb & ## Change/Remove as appropriate
```

Then edit the appropriate file (or both) in /etc/X11/Sessions

For KDE: Файл: /etc/X11/Sessions/kde-3.5

```
#!/bin/sh
export KDEWM="/usr/bin/compizrc"
exec /usr/kde/3.5/bin/startkde
```

Or for GNOME add this to the top of the file : Файл: /etc/X11/Sessions/Gnome

```
#!/bin/sh

export WINDOW_MANAGER="/usr/bin/compizrc"
```

```
...
```

For Xfce4 edit /etc/xdg/xfce4-session/xfce4-session.rc as shown in a previous section. Hopefully Qingy should now start Xgl and Compiz instead of X and some other WM.

Generic startxgl script

There is a generic startup script in the overlay. For usage instructions start it without arguments.

Currently it support Xfce, Xfce-svn, GNOME and KDE.

```
cd /usr/local/overlays/xgl-coffee
```

```
./startxgl
```

## Автоматическое переключение раскладок

Для начала сделаем ebuild в portage overlay

Создаём каталог `mkdir -p /usr/local/portage/x11-misc/xneur` `cd /usr/local/portage/x11-misc/xneur`

Создаём ebuild `nano -w xneur-0.1.0_1.ebuild` Файл: `/usr/local/portage/x11-misc/xneur/xneur-0.1.0_1.ebuild`

```
# Copyright 1999-2005 Gentoo Foundation
```

```
# Distributed under the terms of the GNU General Public License v2
```

```
DESCRIPTION="It's program like Punto Switcher, but has other final aim."
```

```
SRC_URI="http://www.xneur.ru/xneur/${P}.tar.gz"
```

```
HOMEPAGE="http://www.xneur.ru/"
```

```
DEPEND="virtual/x11"
```

```
LICENSE="GPL-2"
```

```
SLOT="0"
```

```
KEYWORDS="~x86 ~amd64 ~sparc ~alpha ~ppc ~hppa ~ppc64"
```

```
src_install () {
```

```
    make DESTDIR=${D} install || die
```

```
    #dodoc AUTHORS README COPYING INSTALL NEWS ChangeLog TODO .xneurrc
```

```
}
```

```
pkg_postinst() {
    # einfo "Run:"
    # einfo "gzcat /usr/doc/${P}/.xneurrc.gz > ~/.xneurrc"
}
```

Далее: `ebuild xneur-0.1.0_1.ebuild digest` Поскольку у нас бэта: `echo "x11-misc/xneur ~x86" >> /etc/portage/package.keywords emerge xneur ;)`

А за подробностями на <http://www.xneur.ru/>

## Cedega 5.1

### Введение

Cedega - коммерческая версия ПО для реализации WinAPI на \*nix-платформах WINE, разработанная Transgaming, ранее известная как Winex. Поддерживается большинство популярных игр. В последней версии добавилась официальная поддержка Sid Meier's Civilization® IV, FIFA® 06 и Need for Speed™: Most Wanted.

Cedega 5.1 состоит из двух компонентов:

Point-to-play - графический интерфейс для простой установки игр, распространяется в виде .rpm/.deb/.tgz

Engine / engine update - модифицированный WINE, распространяется в виде файлов .cpkg - tar архивы, содержащие tgz архив и файл manifest с номером версии .

### Установка

Установка самого эмулятора, без неудобного в использовании Point-to-play выглядит так: Code: Установка Cedega

```
su
```

```
tar -zxf cedega-engine-5.1-local-update.i386.cpkg
```

```
tar -C /usr -zxf cedega_5.1-1.i386.p2p.tgz bin winex
```

```
mv /usr/share/share/doc/* /usr/share/doc
```

```
mv /usr/share/share/man/man1/* /usr/share/man1
```

```
mv /usr/bin/winex3 /usr/bin/cedega
```

```
mkdir /usr/lib/transgaming/cedega; mv /usr/winex /usr/lib/transgaming_cedega
```

```
exit
```

```
Первоначальная настройкаCode: Первоначальная настройка
tar -C $HOME -zxf cedega_5.1-1.i386.p2p.tgz .transgaming
nano ~/.transgaming/config
```

Проверка

```
cedega -version
```

## Windows-эквиваленты

Вы знаете, что хотите сделать, но не знаете, чем? Это список пакетов в портежах Gentoo и их эквиваленты в Windows. Он, конечно же, далеко не полный, но и не предназначен для этого. Он просто даёт возможность лёгкого старта в Gentoo.

Для расширения этой таблицы можно использовать статью: Аналоги Windows-программ в Linux - таблица соответствий. Также, весьма полный лист можно найти здесь: [1](англ.).

Смотрите также официальную страницу портежей(англ.) с портежами, классифицированными по категориям(англ.)

Прикладные и системные программы	Категории ПО	Программы MS Windows
Программы Linux		
Аналоги	Портежи	
Антивирусы много	ClamAV	clamav
Текстовые редакторы	Блокнот / Notepad, WordPad, TextPad, Vim, Xemacs,...	
Kedit (KDE)(грф.)		
Gedit (Gnome)		
Kate (KDE)		
Nedit		
Vim(кнс.)		
Xemacs		
Nvi kedit		
gedit		
kedit		

nedit

vim

xemacs

nvi

Запись CD/DVD    BurnAtOnce, Nero    Graveman

K3b    graveman

k3b

Работа в сети Категории ПО    Программы MS Windows Программы Linux

Аналоги    Портежи

Web-браузеры    Internet Explorer, Mozilla, Firefox, Opera...    Eipphany(грф.)

Galeon(грф.)

Konqueror

Links(кнс.)

Mozilla(грф.)

Firefox(грф.)

Nautilus(грф.)

Opera(грф.) eipphany

galeon

konqueror

links

mozilla

mozilla-firefox

nautilus

opera

Почтовые программы    Outlook Express, Netscape/Mozilla, Thunderbird, The Bat, Eudora, Becky, Datula, Sylpheed/Sylpheed-claws, Opera    Evolution(грф.)

Mozilla messenger

Sylpheed

Sylpheed-claws

Kmail

Gnus(кнс.)

Balsa

Gnumail

Opera

Thunderbird

Mutt(кнс.)

Pine(кнс.) evolution

mozilla

sylpheed

sylpheed-claws

kmail

gnus

balsa

gnumail

opera

thunderbird

mutt

pine

Клиенты быстрого обмена сообщениями (IM) CQ Lite, ICQ Corp, MSN, AIM, Yahoo, ...

Trillian ICQ (бесплатный, может заменить все 3 распространённых клиента), Miranda, Gaim Licq(ICQ) licq

Centericq(Все)(кнс.) centericq

Micq(ICQ)(кнс.) micq

GnomeICU(ICQ) gnomeicu

Gaim(Все) gaim

Aytm(Все) aytm

Kopete kopete

Simple Instant Messenger(Все) sim

aMSN(MSN) amsn

YSM(ICQ)(кнс.) ysm

Yahoo Messenger for Unix ymessenger

Kmess(MSN) kmess

Видеоредакторы и конвертеры VirtualDub, VirtualDubMod Avidemux  
avidemux

Virtual CD VirtualDrive, VirtualCD, Daemon Tools, ... mount -o loop  
"mycd.iso" /mnt/cdrom

Другое Blue Screen of Death XScreenSaver xscreensaver

Офис/БизнесКатегории ПО Программы MS Windows Программы Linux

Аналоги      Портежи

Офисный пакет      MS Office, StarOffice / OpenOffice, 602Software

OpenOffice

Koffice      openoffice

koffice

Текстовый процессор      Word, StarOffice / OpenOffice Writer, 602Text, Abiword  
OpenOffice Writer

KWord

AbiWord      openoffice

kword

abiword

Chat

amsn: alternative for MSN Messenger

gaim: alternative for Aim, ICQ, IRC, Yahoo!, MSN, Jabber (Google Talk)

irssi: (command line based) alternative for mIRC

kopete: alternative for Aim, ICQ, IRC, Yahoo!, MSN, Jabber

licq: alternative for ICQ

psi: jabber client

xchat: alternative for mIRC

konversation: alternative for mIRC

Skype: Skype has also version in Linux

WengoPhone: Wengophone also has a Linux version and is open source

SIM-IM: alternative for ICQ, Jabber, MSN, SMS, Yahoo!, AIM

File managers

krusader: alternative for Total Commander

FTP

gFTP: alternative for FileZilla, SmartFTP

Graphics

digiKam: alternative for the proprietary digital photo software bundled with your camera

gimp: alternative for Adobe Photoshop

Inkscape: alternative for Adobe Illustrator

gqview: alternative for Irfanview

gwenview: alternative for Irfanview

kuickshow: alternative for Irfanview

## Internet

Konqueror: alternative for Internet Explorer

Mozilla Firefox: there is no alternative!

Seamonkey (previously called Mozilla): alternative for Internet Explorer

Opera: free but proprietary alternative for Internet Explorer

## Mail

Evolution: alternative for Outlook

Mozilla Thunderbird: alternative for Outlook Express

Kontact / kmail: alternative for Outlook Express

## Multimedia

AmaroK: alternative for iTunes player/WMP

XMMS: alternative for WinAMP <http://komnas.com/~ashamril/tips/files/win-vs-lin-soft-EN.shtml>

beep-media-player: alternative for WinAMP

kaffeine: alternative for Windows Media Player (also plays DVDs)

MPlayer | gmpayer | kmplayer: alternative for Windows Media Player

Xine: alternative for Windows Media Player (also plays DVDs)

vlc: alternative for Windows Media Player

RealPlayer: runs natively on linux!

Totem : alternative for WMP, Menusupport for DVDs

LMMS : "Linux Music Maker Studio" Linux analog of "FL Studio"  
HomePage:<http://www.lmms.sourceforge.net/>

Rosegarden : MIDI/Audio-sequencer HomePage:<http://www.rosegardenmusic.com/>

Cheese-Tracker : one of the Trackers, use QT

## Office

OpenOffice.org: alternative for MS Word, Excel, Powerpoint and Access, great support for MS formats

KOffice : like above without MS format support. Includes more specialized apps than even the full MS Office Pro

AbiWord : Only like MS Word. Decent MS Format support. Includes grammar check

StarOffice: commercial version of OpenOffice.org, includes extras (grammar check)



kpdf: alternative for Adobe Acrobat Reader  
XPDF: alternative for Adobe Acrobat Reader  
Adobe Acrobat Reader: work natively on linux

## P2P

Azureus: java bittorrent client (or bittorrent which is written in python)  
qtorrent: bittorrent client  
Limewire + gtk-gnutella: Gnutella client  
MLDonkey + Sancho: client and interface for bittorrent, Gnutella, FastTrack, etc.  
aMule: alternative for eMule  
xmule: alternative for eMule  
nicotine: soulseek client  
Valknut, LinuxDC++: альтернативные клиенты для DC (Direct Connect)

## ***Ускорение загрузки системы***

### Введение

Одно из достоинств Gentoo Linux - это возможность настроить и оптимизировать буквально ВСЮ систему, в том числе и её загрузку. Данное направление оптимизации практически неисчерпаемо. При желании (но и при большом риске убить систему) загрузку можно довести и до нескольких секунд. Здесь же мы рассмотрим некоторые наиболее простые и безопасные методы ускорения загрузки.

### Параллельный запуск

Первое, что может дать заметный прирост скорости загрузки, это распараллеливание запуска системных сервисов. В файле `/etc/conf.d/rc` замените NO на YES в переменной `RC_PARALLEL_STARTUP`. Файл: `/etc/conf.d/rc`

...

```
RC_PARALLEL_STARTUP="YES"
```

...

Если у вас нет такой переменной, обновите пакет `baselayout` `emerge -u baselayout`

Примечание: Если эта настройка создаст проблемы при загрузке (некоторые сервисы могут пытаться загрузиться не тогда, когда это нужно), то верните значение NO.

Значительно был улучшен механизм параллельного запуска в новых (нестабильных) версиях baselayout echo "sys-apps/baselayout ~x86" >> /etc/portage/package.keywords emerge sync && emerge sys-apps/baselayout -u

Для совсем реактивного запуска используйте initng emerge sys-apps/initng

Локальная сеть

Если вы часто работаете без подключения к локальной сети, то наверняка заметили, как много времени уходит на определение настроек сети (точнее, на определение того, что настраивать нечего). В этом случае поможет демон ifplugd, который определяет, подключен ли сетевой кабель (точнее, активна ли сеть). Итак:

```
emerge -n ifplugd
```

теперь добавляем ifplugd в список запускаемых сервисов:

```
rc-update add ifplugd boot
```

Обновление зависимостей и переменных окружения

Каждый раз при загрузке Gentoo обновляет зависимости модулей и переменные окружения, разумеется, что это имеет смысл лишь тогда, когда вы действительно внесли изменения в вашу систему. Чтобы это осуществить, сделаем следующие изменения.

Меняем: Файл: /etc/init.d/modules - оригинал

...

```
ebegin "Calculating module dependencies"
    /sbin/modules-update &>/dev/null
    eend $? "Failed to calculate dependencies"
```

...

Файл: /etc/init.d/modules - замена

```
...
if [ /etc/modules.d -nt /etc/modules.conf ]
then
    ebegin "Calculating module dependencies"
    /sbin/modules-update &>/dev/null
    eend $? "Failed to calculate dependencies"
else
    einfo "Module dependencies are up-to-date"
fi
...
```

Меняем: Файл: /etc/init.d/bootmisc - оригинал

```
...
if [ -x /sbin/env-update.sh ]
then
    ebegin "Updating environment"
    /sbin/env-update.sh >/dev/null
    eend 0
fi
...
```

Файл: /etc/init.d/bootmisc - замена

```
...
if [ -x /sbin/env-update.sh ]
then
  if [ /etc/env.d -nt /etc/profile.env ]
  then
    ebegin "Updating environment"
    /sbin/env-update.sh >/dev/null
    eend 0
  else
    einfo "Environment up-to-date"
  fi
fi
...
```

Примечание: при использовании данного трюка следует быть внимательным при обновлении baselayout, т.к. обновление каждый раз будет пытаться вернуть изменения, сделанные вами.

## Монтирование локальных файловых систем

Можно подправить сервис localmount, чтобы он монтировал локальные файловые системы не одну за другой, а все сразу.

Замените: Файл: /etc/init.d/localmount - оригинал

```
...
mount -at ... >/dev/null
...
```

Файл: /etc/init.d/localmount - замена

```
...
mount -aFt ... >/dev/null
...
```

## ***Секреты командной строки***

Общие команды

поиск команды Linux, ее описания и номера секции man страниц

`apropos word`

закодировать файл `file` с помощью GnuPG

`gpg -c file`

раскодировать файл `file`

`gpg file.gpg`

быстрый поиск по словарю слов, начинающихся с `word`

`look word`

подсветить слово `word` в файле `/somefile`

`grep --color word /somefile`

запустить `command` с низким приоритетом

`nice command`

назначить низший приоритет текущему шеллу (и всем потомкам). Может быть полезно, если вашу систему сильно замедляет установка нового пакета (`emerge`). Кстати, для автоматического понижения приоритета `emerge` используется переменная `PORTAGE_NICENESS` в файле `/etc/make.conf`

`renice 19 -p $$`

посмотреть код завершения предыдущей команды

`echo $?`

скачать список новых страничек с нашего сайта в 01:00 в текущую директорию

`echo "wget http://ru.gentoo-wiki.com/Special:Newpages" | at 01:00`

в 17:45 послать пустое письмо с заголовком 'got the r00t?' на `bugs@microsoft.com`

`echo "mail -s 'got the r00t?' bugs@microsoft.com < /dev/null" | at 17:45`

напечатать 1234 в соответствии с настройками локали (в России обычно 1.234)

```
printf "%d\n" 1234
```

запускать просмотр прерываний каждую секунду

```
watch -n1 "cat /proc/interrupts"
```

посмотреть, сколько времени занимает выполнение команды

```
time command
```

удобный алиас для вывода дампа

```
alias hd='od -Ax -tx1z -v'
```

полный путь к команде command

```
which command
```

вывести в 9 колонок по ширине терминала

```
ls | pr -T9 -W$COLUMNS
```

установить время изменения для файла file (в формате YYMMDDhhmm)

```
touch -c -t 0304050607 file
```

показать иерархию запущенных процессов

```
pstree -p
```

показать процессы, использующие файл /dir/file (чаще всего использую нечто вроде 'lsof /mnt/cdrom'). Не забудьте сделать 'emerge -n lsof'

```
lsof /dir/file
```

Ввод-вывод

объединить stderr и stdout (вернее сказать перенаправить stderr в stdout)

```
gcc file.c 2>&1 | less
```

Перенаправить stderr в файл errors.log а stdout в файл compile.log для дальнейшего анализа

```
gcc file.c 2>errors.log 1>compile.log
```

Если нужно запретить вывод потока (например убрать в скрипте сообщения об ошибках) то достаточно перенаправить поток на устройство /dev/null, тогда сообщения уйдут в никуда:

`rm -r /var/tmp/portage 2>&1 >/dev/null` или `rm -r /var/tmp/portage &> /dev/null`

Порой нужно объединить выходной поток сразу нескольких команд, а поток одной из них отключить, тогда группу нужно объединить в скобки:

`( cat /etc/gentoo-release ; cat /etc/passwd >/dev/null; cat /etc/group ) |less`

Навигация по директориям.

вернуться в предыдущую директорию (не путать с 'cd ..')

`cd -`

вернуться в домашнюю директорию

`cd`

перейти в директорию `dir`, запустить `command` и автоматически вернуться назад

`(cd dir; command)`

добавить текущую директорию в стек, чтобы потом можно было сделать `popd` и вернуться к ней

`pushd .`

Дисковое пространство

показать список файлов с информацией о каждом файле (`-l`), отсортировав список по убыванию размера (`-S`) и перевернуть список (`-r`). Получим сортировку по возрастанию.

`ls -lSr`

показать, сколько места на диске занимает файл `file` и директория `dir`

`du -sh file dir`

показать свободное место на примонтированных ресурсах

`df -h`

то же самое но в инодах

`df -i`

показать геометрию размещения разделов жесткого диска (нужны права `root`)

`fdisk -l`

## Работа с CD

создать iso-образ диска и заархивировать его

```
dd bs=1M if=/dev/cdrom | gzip > cdrom.iso.gz
```

создать iso-образ из директории `dir`

```
mkisofs -r dir | gzip > cdrom.iso.gz
```

смонтировать `cdrom.iso` в `/mnt/dir` (для просмотра и правки)

```
mount -o loop cdrom.iso /mnt/dir
```

записать архивированный образ на диск

```
gzip -dc cdrom.iso.gz | cdrrecord dev=0,0,0 -
```

рипнуть дорожки с Audio-CD в текущую директорию (в формате `.wav`)

```
cdparanoia -B
```

создать Audio-CD из всех `.wav` файлов в текущей директории

```
cdrrecord dev=0,0,0 -audio *.wav
```

конвертировать файл `track.cdda.wav` формат `.ogg`

```
oggenc --tracknum="track" track.cdda.wav -o "track.ogg"
```

## Работа с архивами

создать архив директории `dir`

```
tar c dir/ | bzip2 > dir.tar.bz2
```

```
tar -cjf dir.tar.bz2 dir
```

извлечь архив в директорию `/to/dir` (без `'-C /to/dir'` в текущую директорию)

```
bzip2 -dc dir.tar.bz2 | tar x -C /to/dir
```

```
tar -xjf dir.tar.bz2 -C /to/dir
```

создать архив всех `.png` файлов в директории `dir/`



```
find dir/ -name "*.png" | xargs tar rf dir.tar; bzip2 dir.tar
```

скопировать (с сохранением прав доступа!) директорию /dir/to/copy/ в /where/to/

```
( tar cf - /dir/to/copy ) | ( cd /where/to/ && tar xf - )
```

скопировать (с сохранением прав доступа!) содержимое директории /dir/to/copy в /where/to/

```
( cd /dir/to/copy && tar cf - . ) | ( cd /where/to/ && tar xf - )
```

скопировать (с сохранением прав доступа!) директорию /dir/to/copy/ в директорию /where/to/ на удаленной машине

```
( tar cf - /dir/to/copy ) | gzip | ssh user@remote 'cd /where/to/ && gzip -dc | tar xf -'
```

создать и сохранить бэкап жесткого диска на удаленной машине

```
dd bs=1M if=/dev/hda | gzip | ssh user@remote 'dd of=hda.gz'
```

Работа с файлами

удобный листинг по команде l

```
alias l='ls -l --color=auto'
```

вывести листинг с упорядочиванием по дате

```
ls -lrt
```

показать в папке dir файлы модифицированные раньше, чем 2 дня назад

```
find dir -mtime +2
```

удаляем в папке dir файлы старше 1 часа

```
find dir -type f -mmin +60 -exec rm -f {} \;
```

удаляем в папке dir файлы старше 10 дней

```
find dir -type f -mtime +10 -exec rm -f {} \;
```

найти в текущей директории (и ниже) .c и .h файлы содержащие строку "search string"

```
find -name "*.ch" | xargs grep -E "search string"
```

искать строку "search string" только в обычных файлах

```
find -type f | xargs grep -E "search string"
```

искать строку "search string" только в текущей директории (не спускаться ниже)

```
find -type f -maxdepth 1 | xargs grep -E "search string"
```

в текущей директории найти все файлы с расширением sql содержащие USER1 и заменить в них USER1 на USER2

```
tmp="/tmp/$RANDOM$$tmp"; f="USER1"; r="USER2";
```

```
find . -name '*.sql' -exec grep -l "$f" {} \; |
```

```
xargs --replace="{}" bash -c "( sed 's/$f/$r/g' < {} > $tmp && cat $tmp > {} && rm -f $tmp )"
```

найти файл в базе данных программы slocate. Замечание: данное регулярное выражение эквивалентно маске \*file\*.txt

```
locate -r 'file[^\.]*\.'
```

Работа с файловой системой

отформатировать флоппи-диск с FAT

```
mkdosfs -c -f 16 -n "название тома" /dev/fd0 или mkfs -t fat16 /dev/fd0
```

"правильная" кодировка и права файлов для сменных носителей (floppy, CD, flash)

необходимо прописать в /etc/fstab!!!

для CDROM

```
/dev/cdrom /mnt/cdrom iso9660 ro,nosuid,noauto,exec,user,nodev 0 0
```

для "дискетки"

```
/dev/fd0 /mnt/floppy vfat icharset=koi8-r, sync, nosuid, codepage=866, user, --, noauto, nodev, unhide 0 0
```

для раздела Windows

```
/dev/hda1 /mnt/win vfat user,exec,umask=0,codepage=866,icharset=koi8-r 0 0
```

для "флешки"

```
/dev/sda1 /mnt/flash vfat user,exec,umask=0, sync, codepage=866, icharset=koi8-r 0 0
```

вышенаписанное справедливо для локали koi8-r, ваша может отличаться (см. locale), в этом случае все "koi8-r" необходимо исправить на свои, и помните, что это, лишь,

пример

Работа с календарем

вывести на экран календарь на текущий, предыдущий и следующий месяцы

```
cal -3
```

на какой день недели выпал в этом году день рождения Linux?

```
date --date='25 Aug' +%A
```

конвертировать в дату (в соответствии с локалью) - 130204800 секунд, прошедшие с начала эпохи Unix

```
date --date '1970-01-01 UTC 130204800 seconds'
```

Сколько сейчас времени на западном побережье США (используйте tzselect чтобы узнать параметр для TZ)

```
TZ="America/Los_Angeles" date
```

Работа с сетью

(Предварительно сделать emerge на net-tools, sys-apps/iproute2, net-dns/bind-tools)

показать сетевые интерфейсы

```
ip link show
```

показать статус сетевых интерфейсов

```
ethtool interface или /sbin/ifconfig
```

переименовать eth0 в wan

```
ip link set dev eth0 name wan
```

добавить ip 1.2.3.4 с маской 255.255.255.0 на eth0

```
ip addr add 1.2.3.4/24 brd + dev eth0
```

поднять интерфейс

```
ip link set dev interface up
```

опустить интерфейс

```
ip link set dev interface down
```

сделать шлюзом по умолчанию 1.2.3.254

```
ip route add default via 1.2.3.254
```

показать ip адрес для name

```
host name
```

показать прослушиваемые порты в системе (и кто их слушает)

```
netstat -lp --inet
```

показать активные соединения

```
netstat -p --inet
```

Математика

простое вычисление

```
echo "(321-123)/123" | bc -l
```

простое целочисленное вычисление с использованием bash

```
echo "$(( (51+123)/2 ))"
```

использование python для научных вычислений

```
echo "print (10E3-123)/123" | python
```

приведение систем счисления (в данном случае, десятичной к шестнадцатеричной)

```
echo "obase=16;ibase=10;123" | bc
```

Более сложное вычисление - максимальная скорость передачи (в пакетах в секунду) в Fast Ethernet сети (100Mb)

```
echo "framing=20; minsize=64; (100*10^6)/((framing+minsize)*8)" | bc
```

А здесь мы строим график зависимости скорости передачи от размера пакета всё в той же 100-мегабитной сети.

```
echo "framing=20; plot [64:1518] (100*10**6)/((framing+x)*8)" | gnuplot -persist
```

## **Тонкость работы ccache с emerge**

Вы пробовали оценить эффективность работы ccache при пересборке системы? Прodelайте опыт: очистите кэш (ccache -C) и запустите пересборку (emerge --emptytree world). После пересборки десятка-другого пакетов прервите процесс и посмотрите статистику (ccache -s) Запустите пересборку по новой и опять прервитесь на старом месте. Что показывает статистика? Много ли совпадений?

А теперь добавьте в /etc/make.conf строки

```
CC=gcc
```

```
CXX=g++
```

очистите кэш и статистику и повторите эксперимент заново. Не правда ли, что после этого кэш только и заработал?

Суть: по состоянию на апрель 2005 ccache опознает как компилятор только стандартные имена типа gcc, g++ и не опознает такие имена как i686-pc-linux-gnu-gcc. Однако именно такие имена и используют пакеты в качестве компилятора после работы скриптов конфигурации configure

## **Keymap**

Варианты раскладок клавиатуры для прописывания в /etc/conf.d/keymaps (в старых версиях baselayout переменная находится в файле /etc/rc.conf)  
Файл: /etc/conf.d/keymaps

```
...
```

```
KEYMAP="-u ru4"
```

```
...
```

Доступные варианты:

ru -- кодировка KOI-8, переключение [Right-Ctrl], одиночный символ [Right-Alt]

ru1 -- кодировка KOI-8, переключение [Right-Alt]

ru2 -- альтернативная кодировка (IBM866), переключение [Right-Alt]

ru3 -- кодировка KOI-8, 102 кнопочная клавиатура, переключение [CapsLock], [Shift +CapsLock] -- фиксация регистра

ru4 -- кодировка KOI-8, 105 кнопочная клавиатура, переключение [CapsLock], [Shift +CapsLock] -- фиксация регистра

ru-cp1251 -- кодировка CP1251, переключение [Right-Ctrl], одиночный символ [Right-Alt]

ru-ms -- раскладка как в Microsoft Cyrillic keyboard layout, переключение [Right-Ctrl]

ru\_win -- кодировка CP1251, переключение [Right-Ctrl], одиночный символ [Right-Alt]  
ru-yawerty -- кодировка KOI-8, раскладка "ЯВЕРТЫ" (это бывает удобно при работе на клавиатуре без русских символов), переключение [Right-Alt]

Вариант, которого нет в стандартной поставке, но весьма удобен ru-utf. Переключение [Right-Ctrl], одиночный символ [Right-Alt]. Брать тут: <http://mlclm.narod.ru/ru-utf.map.gz>, сохранить в каталог /usr/share/kbd/keymaps/i386/qwerty или в случае отсутствия этого, в каталог /usr/share/keymaps/i386/qwerty.

Кроме того есть вариант ru-mab. Он даёт переключение по [Ctrl+Shift]. Взять можно тут: <http://moose.ylsoftware.com/gentoo.ru/ru-mab.map.gz> Code: /etc/conf.d/keymaps

...

```
KEYMAP="ru-utf"
```

# Часто задаваемые вопросы

## 2. Введение

### Как произносится слово Gentoo, и что оно означает?

Gentoo (Хинду) это вид маленького и быстрого пингвина, произносится по-английски как "gen-too" (джин-туу). Научное название пингвина - *Pygoscelis papua*. Имя Gentoo было дано пингвину жителями Фолклендских (Мальвинских) Островов (Islas Malvinas). По-русски прижилось произношение Gentoo как "генту".

### Что делает Gentoo особенным?

Gentoo использует похожую на порты BSD систему, называемую Portage (англ.). Portage - это система управления пакетами, дающая огромную гибкость при установке и сопровождении программного обеспечения в системе Gentoo. Она предоставляет поддержку директив времени компиляции (с помощью USE-флагов), условные зависимости, сведения о пакете перед его установкой, безопасную установку (через "песочницу") и удаление ПО, системные профили, защиту конфигурационных файлов, а также многие другие возможности.

С Gentoo вы можете построить всю свою систему из исходных кодов, используя необходимую вам оптимизацию. Вы полностью контролируете, какие пакеты устанавливать, а какие нет. Gentoo предоставляет множество вариантов установки, удовлетворяющие вашим запросам, поэтому его называют метадистрибутивом.

Gentoo активно развивается. Поддерживается высокий темп разработки: заплатки быстро включаются в основное дерево проекта, документация обновляется ежедневно, часто добавляются новые функции в Portage, а официальные выпуски выходят дважды в год.

## 3. Установка

### Все работает очень нестабильно, мои флаги оптимизации "-O9 -ffast-math -fomit-frame-pointer". В чем проблема?

Не пробуйте использовать что-либо выше, чем -O3 т.к. это не поддерживается

текущими версиями gcc. Очень агрессивная оптимизация иногда заставляет компилятор изменять ассемблерный код так, что он уже выполняет несколько другие действия, чем предполагалось.

Попробуйте компилировать с флагами CFLAGS -O2 -march=<your\_arch> прежде чем сообщать об ошибках.

## Как изменить пароль root (или любого другого пользователя)?

Используйте passwd, чтобы сменить пароль пользователя, под именем которого вы вошли. Как root, вы можете сменить пароль любого пользователя в системе командой passwd username. Дополнительные параметры смотрите на странице: man passwd.

## Как добавить обычного пользователя?

Команда adduser username добавит пользователя с именем "username". Однако, такой метод не даст пользователю некоторые права которые вы, возможно, хотите ему предоставить, поэтому предпочтительней использовать:

Листинг 3.1: Использование useradd

```
# useradd -m -G users,audio,wheel username
```

В итоге в системе появится пользователь с именем "username". Параметр audio добавит его в группу audio и позволит пользователю работать со звуковыми платами. Параметр wheel добавит пользователя в группу wheel, которая позволит ему запускать команду su, чтобы получить привилегии суперпользователя.

## Почему пользователь не может получить права root командой su?

В целях безопасности, пользователи могут выполнять команду su, только если они состоят в группе wheel. Чтобы добавить пользователя в группу wheel, выполните следующую команду из-под root:

Листинг 3.2: Добавление пользователя в группу wheel

```
# gpasswd -a username wheel
```



## Как отключить devfs?

Gentoo может работать с devfs, udev, а так же поддерживает статический /dev. С появлением ядра 2.6, которое стабильно на большинстве архитектур, рекомендуется использовать udev. Информацию по настройке udev можно найти в Руководстве по udev.

Если вы хотите использовать статический /dev, установите параметр `RC_DEVICES="static"` в `/etc/conf.d/rc`.

## Могу ли я обновить Gentoo без переустановки?

На самом деле, нет разницы между разными версиями после их установки. Gentoo 1.4 и старше основаны на glibc-2.3.x. Поэтому, выполнив `emerge --sync && emerge -uDN world`, вы приведете свою систему в самое актуальное состояние (latest Gentoo). Различия между отдельными версиями Gentoo заключаются в установочных носителях и прекомпилированных пакетах. Подробнее о профилях и о их роли при обновлении смотрите в руководстве по обновлению Gentoo.

## Ядро не загружается (правильно), что мне теперь делать?

Вы не обязаны повторять каждый шаг установки, но понимание ядра и связанных с ним этапов установки необходимо. Предположим, вы установили Gentoo на `/dev/hda1` (точка монтирования `/boot`) и `/dev/hda3` (точка монтирования `/`) с разделом подкачки `/dev/hda2`:

Листинг 3.3: Конфигурация ядра

Загрузитесь с установочного CD и дождитесь приглашения командной строки

Сначала примонтируем все разделы:

```
# mount /dev/hda3 /mnt/gentoo
# mount /dev/hda1 /mnt/gentoo/boot
# swapon /dev/hda2
# mount -t proc none /mnt/gentoo/proc
```

Затем перейдем в вашу систему командой `chroot` и настроим ядро:

```
# chroot /mnt/gentoo /bin/bash
# env-update && source /etc/profile
# cd /usr/src/linux
```

```
# make menuconfig
```

Отметьте/уберите все что по вашему вы отметили/убрали неправильно в прошлый раз

Затем выходите и откомпилируйте ядро:

```
# make && make modules_install
```

Теперь замените bzImage новой его версией:

```
# cp arch/i386/boot/bzImage /boot/<kernel_name>
```

Если используете LILO, выполните lilo; пользователи GRUB - пропустите этот шаг:

```
# /sbin/lilo
```

Теперь выйдите из среды chroot и перезагрузитесь.

```
# exit
```

```
# umount /mnt/gentoo/proc /mnt/gentoo/boot /mnt/gentoo
```

```
# reboot
```

С другой стороны, проблема может быть с вашим загрузчиком; тогда поступите так же, только вместо конфигурирования/компилирования ядра перенастройте свой загрузчик (его перекомпиляция не нужна).

## **Мой прокси-сервер требует аутентификации, что делать?**

Чтобы Portage автоматически использовал реквизиты, определите их в /etc/make.conf:

Листинг 3.4: /etc/make.conf

```
HTTP_PROXY="http://username:password@yourproxybox.org:portnumber"
```

```
FTP_PROXY="ftp://username:password@yourproxybox.org:portnumber"
```

```
RSYNC_PROXY="rsync://username:password@yourproxybox.server:portnumber"
```

## **Как записать ISO файл на компакт-диск?**

Вам необходимо записать его в raw-режиме. Это означает что вам нужно НЕ просто поместить его на CD, а интерпретировать сам файл как CD.

Существует множество утилит для записи CD; рассказывать о них всех будет сизифовым трудом. Однако описать несколько популярных никогда не повредит:

Используя EasyCD Creator, выберите File, Record CD from CD image. Затем поменяйте тип файла на ISO image file. Найдите ISO-файл и кликните Open. При нажатии на Start recording, ISO образ будет записан на диск.

Используя Nero Burning ROM, закройте мастер, автоматически появляющийся при запуске, нажмите Burn Image из меню File. Выберите необходимый образ и нажмите Open. Теперь нажмите кнопку Burn и наблюдайте за процессом прожига вашего нового CD.

Используя cdrecord, просто наберите `cdrecord dev=/dev/hdc` (заменив `/dev/hdc` именем своего CD-RW привода), а затем имя ISO образа :)

Используя K3B, выберите Tools > CD > Burn CD Image. Затем найдите ISO образ, используя окно 'Image to Burn'. Наконец, Нажмите Start.

В Mac OS X Panther, запустите Disk Utility из Applications/Utilities, нажмите Open из меню Images, в главном окне выберите примонтированный образ диска и нажмите Burn в меню Images.

В Mac OS X Jaguar, запустите Disk Copy из Applications/Utilities, выберите Burn Image из меню File, найдите образ ISO и нажмите кнопку Burn.

## **Какой диск/стадию надо использовать с моим процессором?**

Для начала надо узнать, что за процессор вы используете. Предположим, что это Pentium-M. Теперь необходимо найти, с какими инструкциями процессора он совместим. Вы можете проконсультироваться на сайте производителя, однако обычно достаточно погуглить :-).

Если вы не уверены, выберите "меньший" файл CD/stage, например, от i686 или даже универсальный x86 (или эквивалентный в своей архитектуре). Так вы можете быть уверены, что ваша система точно заработает, но будет работать не так быстро, как если бы она была скомпилирована с использованием флагов оптимизации.

Учтите, что существует намного больше параметров, чем те с которыми скомпилированы бинарные пакеты Gentoo. Чтобы правильно выставить `-march`, ознакомьтесь с `gcc guide`.

## **Я не могу выйти в интернет после перезагрузки. В чем может быть дело?**

Для начала проверьте, правильно ли определилась ваша сетевая карта ядром. Выполните `ifconfig -a`, найдите `eth0` или `wlan0` (если у вас беспроводная карта). Возможно, потребуется загрузить специфичный модуль ядра, чтобы ядро правильно опознало вашу сетевую карту. Если так, то убедитесь, что эти модули ядра перечислены в `/etc/modules.autoload.d/kernel-2.6` (или `kernel-2.4`, если вы до сих пор используете ядро 2.4).

Если вы забыли включить поддержку вашей карты в ядро, придется его переконфигурировать (и пересобрать).

Если ваша сетевая карта найдена ядром, но настроена на использование DHCP, то вы могли забыть установить `dhcpcd` (`emerge dhcpcd`). Необходимо загрузиться с установочного CD, чтобы установить `dhcpcd`.

О том, как восстановить вашу систему, используя установочный CD, смотрите здесь.

## **Я хочу загрузить Windows из GRUB или LILO, но вижу только черный экран. Что делать?**

Это известная проблема. Windows отказывается грузиться и показывает черный/пустой экран, если установлен не на первый жесткий диск. Чтобы исправить ситуацию, надо, используя настройку загрузчика, "надурить" Windows, чтобы она "думала" что установлена на первый жесткий диск. Учтите, что в следующем примере Gentoo установлен на `hda` (первый диск), а Windows на `hdb` (второй). Подправьте ваш конфигурационный файл соответственно.

Листинг 3.5: Пример двойной загрузки для Windows в `grub.conf`

```
title Windows XP
    map (hd1) (hd0)
    map (hd0) (hd1)
    rootnoverify (hd1,0)
    chainloader +1
```

Листинг 3.6: Пример двойной загрузки для Windows в `lilo.conf`

```
other=/dev/hdb1
    label=WindowsXP
    table=/dev/hdb
    map-drive = 0x80
    to = 0x81
    map-drive = 0x81
    to = 0x80
```

Это заставит Windows поверить, что она установлена на первый жесткий диск, и грузиться без проблем. Дополнительную информацию можно найти в разделе Документация GRUB и в `man lilo.conf`, смотря, какой загрузчик используете.

## Как установить Gentoo, используя файл Stage1 или Stage2?

Настольная книга Gentoo только объясняет, как установить ОС, используя файл stage3. Однако, все еще осталась возможность установки из stage1 или stage2. Возможность оставлена для нужд разработчиков (команда Release Engineering начинает со stage1, чтобы получить stage3), но не обязательна к использования пользователями: файл stage3 тоже очень хорошо подходит для самогенерации системы. Вам потребуется соединение с Интернетом.

Самогенерация (bootstrapping) - построение набора инструментов (библиотека Си и компилятор) для системы, после чего устанавливаются пакеты, входящие в ядро системы. Чтобы сгенерировать систему, выполните установку со stage3. Перед главой настройка ядра, отредактируйте сценарий bootstrap.sh под ваши нужды, а затем выполните его:

Листинг 3.7: Самогенерация системы

```
# cd /usr/portage/scripts  
# vi bootstrap.sh  
  
# ./bootstrap.sh
```

Затем пересоберите все ключевые системные пакеты только что собранным набором (библиотека - компилятор). Нам необходимо пересобрать их т.к. в файле stage3 они уже есть:

Листинг 3.8: Пересборка ключевых системных пакетов

```
# emerge -e system
```

Теперь можно переходить к настройке ядра. Прекомпилированные пакеты GRP больше использовать нельзя.

## ***4. Управление пакетами***

### **В каком формате хранятся пакеты?**

Пакеты не "хранятся". Вместо этого Gentoo предоставляет набор сценариев,

которые могут разрешать зависимости, закачивать исходный код, и компилировать версию пакета именно под ваши нужды. В общем, мы собираем двоичные файлы только для выпусков (версий) и снимков (snapshots). Gentoo Ebuild HOWTO (англ.) рассказывает о содержимом сценариев ebuild в деталях.

Для полных ISO-выпусков, мы создаем полный набор двоичных пакетов в расширенном .tbz2 формате, который совместим с .tar.bz2 и включает метаинформацию, присоединенную к концу файла. Они могут использоваться для быстрой и эффективной установки работающей (правда, не очень оптимизированной) версии пакета.

Также возможно создать RPM (файлы Redhat package manager) средствами Gentoo, но на данный момент нельзя использовать уже готовые RPM для установки.

## **Я хочу выполнить сценарий ./configure сам. Возможно ли это?**

Да, но это не так просто, и не рекомендуется. Т.к. вы все равно должны понимать, как работает Portage, то лучше сделайте заплатку для ebuild-файла, выполняющую то, что вы хотите, и положите ebuild-файл в оверлей Portage (для этого он и существует). Такой подход более подходит для поддержки, и обычно легче. Чтобы узнать больше, см. Ebuild HOWTO (англ.).

## **Как использовать emerge, если я за межсетевым экраном?**

Смотрите вопросы о проху, rsync, и Закачка исходных кодов вручную.

## **Что делать, если rsync мне не подходит?**

Если вы за межсетевым экраном, который не пропускает rsync трафик, то вы можете использовать emerge-webrsync, который закачает и установит снимок дерева Portage через обычный HTTP. Обратите внимание на раздел проху этого документа, чтобы выяснить, как закачивать исходные коды и снимки Portage через прокси-сервер.

## **У меня дома лишь слабое модемное подключение. Могу ли я скачать исходные коды в другом месте, а затем добавить их в систему?**

Конечно. Выполните emerge --pretend package, чтобы понять, какие программы необходимо установить. Чтобы увидеть, какие исходные коды необходимо скачать и откуда, выполните emerge -fp package. Закачайте исходники, и принесите их на

любом носителе домой. Скопируйте их в `/usr/portage/distfiles/`, затем просто наберите `emerge package`. Но имейте в виду: это нудное занятие.

### **Исходные коды хранятся в `/usr/portage/distfiles`. Безопасно ли удалить их?**

Удаление этих файлов никак не повлияет на каждодневную производительность. Однако лучше оставить самые последние версии файлов; часто несколько файлов `ebuild` выпускаются с расчетом на одну и ту же версию исходных кодов. В таком случае, если вы удалили архивы с исходными текстами, а затем решили обновить ПО, потребуется заново скачать нужные исходники через Интернет. Существуют программы, разработанные пользователями, для удаления всех исходников, кроме самых последних версий. Заметьте, они хоть и работают, но официально не поддерживаются. Используйте их на свой страх и риск.

### **Что находится в `/var/tmp/portage`? Можно ли удалить файлы и каталоги из `/var/tmp/portage`?**

Во время компиляции, Gentoo сохраняет исходники пакетов в `/var/tmp/portage`. Эти файлы и каталоги обычно удаляются после успешной установки, но не всегда. Можете спокойно очищать этот каталог, если `emerge` не запущен. Чтобы быть уверенным, всегда выполняйте `pkgrep emerge` перед очисткой каталога.

## **5. Использование**

### **Как установить международную раскладку клавиатуры?**

Отредактируйте переменную `KEYMAP` в файле `/etc/conf.d/keymaps`. Чтобы консоль заработала правильно с расширенными знаками вашей раскладки, вам может также потребоваться настроить переменные `CONSOLETRANSLATION` и `CONSOLEFONT` у себя в `/etc/conf.d/consolefont` (за дальнейшими сведениями о локализации своей среды обращайтесь к нашему руководству по локализации (англ.)). Затем выполните перезагрузку (`reboot`), или перезапустите сценарии `keymaps` и `consolefont`:

Листинг 5.1: Перезагрузка `keymaps`

```
# /etc/init.d/keymaps restart
```

```
# /etc/init.d/consolefont restart
```

Разрешение имен DNS работает только у суперпользователя.

У файла `/etc/resolv.conf` неправильно выставлены права; Выполните `chmod`, как описано ниже:

Листинг 5.2: Смена прав `/etc/resolv.conf`

```
# chmod 0644 /etc/resolv.conf
```

## **Почему мой пользователь не может использовать свой файл `crontab`?**

Вы должны внести своего пользователя в группу `cron`.

## **Как включить `numlock` при загрузке?**

Если вы работаете в командной строке, вам всего лишь необходимо выполнить `rc-update add numlock default &&/etc/init.d/numlock start`.

Каждый GUI предоставляет собственные средства для этого; смотрите раздел справки или руководство.

## **Как сделать очистку терминала при выходе?**

Для этого добавьте `clear` в ваш сценарий `~/.bash_logout`:

Листинг 5.3: Очистка терминала при выходе

```
$ echo clear >> ~/.bash_logout
```

Если вы хотите, чтобы у новых пользователей это работало автоматически, добавьте ту же строку в `/etc/skel/.bash_logout`:

Листинг 5.4: Очистка экрана при выходе у новых пользователей



```
# echo clear >> /etc/skel/.bash_logout
```

## **Не удастся запускать X приложения из-под root'a (после su)**

Похоже, что эта проблема появляется только при графическом входе. Пользователи, использующие startx, с ней не сталкиваются. Проблема в Gentoo's PAM, однако выход довольно прост: добавьте следующую строку в файл /etc/profile.

Листинг 5.5: Экспорт XAUTHORITY

```
export XAUTHORITY="${HOME}/.Xauthority"
```

## **6. Сопровождение**

### **ReiserFS и порча файловых систем - как исправить, и т.п.**

Если раздел с ReiserFS испорчен, попытайтесь загрузиться с установочного диска Gentoo и выполните reiserfsck --rebuild-tree на испорченной файловой системе. Эта команда предназначена для восстановления файловой системы, но все же из-за ее повреждения некоторые файлы или каталоги могут быть утеряны.

## **7. Разработка**

### **Где оставлять сообщения об ошибках?**

Используйте нашу систему Bugzilla. Если вы не уверены в актуальности ошибки, можете зайти на канал #gentoo в сети IRC FreeNode

### **Как часто появляются новые стабильные выпуски?**

Пакеты Gentoo обычно обновляются вскоре после обновления основного кода программы. А если речь идет о том, как часто выходят файлы стадий/профили/выпуски ISO самого дистрибутива, смотрите страницу Release Engineering Project (англ.) Новые стабильные версии анонсируются в списке

рассылки gentoo-announce. Смотрите вопрос об обновлении.

## **Мой динамик орет как сумасшедший. Как выключить динамик в консоли?**

Консольные звуки можно выключить, используя `setterm`:

Листинг 7.1: Использование `setterm`

```
# setterm -blength 0
```

Если вы хотите, чтобы звуки выключались при загрузке, поместите эту команду в сценарий `/etc/conf.d/local.start`. Однако, это избавит вас от звуков только в текущем терминале. Чтобы выключить звук и на других терминалах, перенаправьте вывод команды на нужный терминал:

Листинг 7.2: Использование `setterm` (bis)

```
# setterm -blength 0 >/dev/vc/1
```

Нужно заменить `/dev/vc/1` на терминал, в котором вы хотите выключить звук.

## **8. Ресурсы**

### **Где найти информацию о Gentoo Linux?**

Официальная документация Gentoo находится по адресу <http://www.gentoo.org/doc/ru/>.

### **Могу ли я купить CD с Gentoo Linux?**

Установочные диски для всех поддерживаемых архитектур имеются в нашем магазине Gentoo. При покупке дисков в нашем магазине, вы поддерживаете разработку дистрибутива. Поэтому, если можно, покупайте диски у нас.

Также список продавцов находится на странице загрузки Gentoo Linux.

## **В этом списке нет ответа на мой вопрос. Что делать?**

В первую очередь, просмотрите соответствующую документацию, если это не помогло, в Google можно найти всевозможные списки рассылки Gentoo. Чтобы искать в списках рассылки Gentoo, наберите "lists.gentoo.org что-то-там" в строке поиска. Если все попытки тщетны, и вы готовы просто покончить с Gentoo, сходите на irc канал: #gentoo on irc.freenode.net.

Примечание: Также можно заглянуть на <http://forums.gentoo.org>, где, ко всему прочему, есть русскоязычное сообщество пользователей - прим. пер.

## **Наивные вопросы юных линуксоидов**

### ***Вступление***

Перейдя из ОС Windows в Linux пользователи сталкиваются с кучей непонятных им вещей. Есть много вопросов которые их тревожат, но спросить они не решаются - боятся. Так давайте же поможем им поскорее разобраться с новой замечательной системой. Может и сами узнаем чего нового. Примечание: Так как Gentoo новичкам не поставить, то этот материал носит скорее пропагандистский характер

Общие вопросы

### **Как ставить программы?**

Первая проблема которая встает перед линуксоидом это установка программ. Программы можно ставить из исходников и бинарных пакетов. На бинарниках завязаны такие дистрибутивы как RedHat, Mandrake, Fedore Core, ASP Linux, ALT Linux, Debian и прочие. С исходниками работают Gentoo Linux и \*BSD-системы.

Чтобы установить программу в Gentoo Linux нужно только ввести: emerge программа emerge скачает все необходимое с интернета, скомпилирует и установит. Если же программы нет в дереве портов, то придется ставить ее из тарболла.

Поправочка: В Gentoo тоже можно ставить прекомпиленные (заранее собранные) программы путем вызова команды: emerge --usepkg --getbinpkg программа или короче так: emerge -gk программа Но сборка из исходных текстов предпочтительнее.

## Что еще за тарболл?

Все программы в исходных текстах поставляются в виде тарболлов (файлы с расширением tar.gz, tar.bz2, tbz2, tgz, впрочем последние два - большая редкость). Тарболл по своей сути - это просто архив, который можно распаковать следующими командами: `tar -xvjf super_proga_0.1.tar.bz2` для tar.bz2(или tbz2) и `tar -xvzf super_proga_0.1.tar.gz` для tar.gz(tgz). И архив будет распакован в текущий каталог.

Кстати -v - абсолютно лишнее. :) Dron

Я бы сказал так, распаковка `tar -zxvf`, архивы bz и gz2 -jxvf, удобно запомнить, различается только j в начале. =) das

Некоторым проще запоминать xzf как eXtract Zipped File. j - для bzip2 wt

Дальнейшие инструкции по установке можно прочитать в файле INSTALL или README, но чаще всего установка выглядит так: `./configure make` Далее от пользователя root: `make install`

Но пользователям Gentoo лучше прочитать следующий раздел.

### Gentoo Way

Установка из тарболлов может показаться достаточно простой на первый взгляд, однако, она может затруднить управление пакетами и создать некоторые проблемы в дальнейшем, поэтому лучше воспользоваться мощью системой управления пакетами portage. Наверняка, о бесппроблемной сборке необходимой вам программы уже позаботилось сообщество пользователей и разработчиков Gentoo. Чтобы узнать это, попробуйте найти нужную вам программу в базе портежей набрав: `emerge -s [название программы]` или по описанию: `emerge -S [ключевое слово]` что правда занимает достаточно много времени, и лучше использовать специально для этого написанные утилиты, например `esearch emerge esearch eupdatedb esearch [название программы]` или по описанию `esearch -S [ключевое слово]` так же можно проводить поиск по категориям. Например `esearch -F mail-client` покажет все почтовые клиенты, информация о которых есть в дереве портежей. Так же при поиске можно использовать регулярные выражения.

Неплохим подспорьем может послужить GUI фронтенд к портежам, например Porthole(интерфейс GTK).Для любителей KDE есть Kentoo(встраивается в центр управления) и Guito.

Если вам потребовалось установить программу из неофициального ebuild'a, сначала прочтите Portage\_Overlay.

Если же ebuild скрипта найти не удалось или нужно наложить дополнительные патчи, то это значит, что пора освоить написание скриптов. Для этого можно воспользоваться skel.ebuild'ом который находится в каталоге portage. Так же не повредит прочтение en:HOWTO Create an Updated Ebuild.

Удобной средой разработки и правки ebuild скриптов является abeni

## А кто такой root и как им стать?

Говоря языком Windows, root - это администратор. Но в отличие от Windows, root - абсолютный хозяин системы. Он может сделать все, даже удалить системные файлы, или произвести низкоуровневое форматирование жесткого диска. По идее, root всегда знает что делает, поэтому такая власть вполне оправдана. Рутром можно стать несколькими способами:

первоначально войти систему под этим аккаунтом.

через su. Для этого в консоли введите su

```
[user@gentoo user]$su
```

Password:

```
[root@gentoo user]#
```

Пользователь user должен быть в группе wheel. При вводе пароля он отображаться не будет.

через sudo. Это дополнительная программа, которую можно установить командой:

```
$ emerge sudo
```

Ее преимущества перед обычным su заключаются в том, что можно протолировать все выполненные команды, а значит понять, кто и как именно сделал что-то неправильно. :) Кроме того, можно разрешить определенным пользователям выполнять только определенные команды с правами root, что существенно повышает безопасность системы. И, наконец, для использования sudo не нужно знать пароль root - достаточно помнить свой. ;) Все эти и другие настройки производятся редактированием файла конфигурации sudo - /etc/sudoers.

Пользоваться ей очень просто:

```
$ sudo команда
```

(например, "sudo ls -l"). Если же вам нужно выполнить не одну команду, а просто временно стать root, используйте следующий синтаксис:

```
$ sudo -s
```

Для того, чтобы перестать быть root после этого, достаточно ввести команду

```
$ exit
```

Секреты консоли

## **Основные команды**

Можно на 100% гарантировать, что количество команд в вашей Linux системе больше пары тысяч. Но практически все линуксоиды используют из этого количества не более 100 команд, хотя и это, пожалуй, завышенное число. Человеку, который линукс видел только на скринах, будет полезно узнать, как же использовать эту консоль.

### **Команда man**

Linux - отлично документированная система. Подробную информацию о программе, команде или основным конфигурационным файлам можно получить при помощи команды `man`. Например `man whereis` выдаст вам подробную справку по использованию команды `whereis`, а команда `man fstab` выдаст синтаксис конфигурационного файла `/etc/fstab`.

Если вам не удалось найти нужную информацию, можно заглянуть в `/usr/share/doc` (в Gentoo Linux так же находится в `/usr/doc`). Там хранится документация программ.

Примечание: Для более детальной информации обратитесь в соответствующий раздел Страницы MAN

### **Команда cd**

Без этой команды навигация по файловой системе будет невозможна. Смысл её простой - сменить текущий каталог. А синтаксис такой: `cd /home/vasya/katalog` Можно использовать как абсолютные, так и относительные пути к папкам.

Абсолютный путь - это путь указанный от корневого каталога(`/`).

Относительный путь - это путь указанный относительно текущего каталога.

Например, если вы находитесь в каталоге `/home/vasya`, то для перехода

в /home/vasya/katalog вам нужно ввести только `cd katalog`. Необходимо отметить, что в Linux нет такого понятия, как "диски". Все разделы жестких дисков, CD-приводы, сетевые диски и т.д. отображаются в виде каталогов файловой системы (обычно в /mnt).

Чтобы узнать в каком каталоге вы сейчас находитесь используется команда `pwd`.

## Команда ls

Используется для получения списка файлов и каталогов. Например, `ls` выведет нам содержимое текущего каталога. Для вывода информации о каталоге, в котором вы сейчас не находитесь, используется `ls /etc`. Наверняка вас не удовлетворят только имена файлов. Для увеличения количества выводимой информации используйте опцию `"-l"` `ls -l`. Теперь `ls` выведет вам много полезной информации, такой как права, время последней модификации, размер и прочее. `ls -la` с опцией `"-a"` покажет скрытые файлы (by morgoth)

## Команда cp

Предназначена для копирования файлов и каталогов. Например, скопируем файл `/home/vasya/katalog/linux.txt` в каталог `/home/vasya`. Делается это так: `cp /home/vasya/katalog/linux.txt /home/vasya`. А вдруг мы хотим чтобы файл `linux.txt` получил имя `gentoo.txt`? Тогда делаем так: `cp /home/vasya/katalog/linux.txt /home/vasya/gentoo.txt`. По умолчанию `cp` не копирует каталоги. Чтож, заставим его это делать. Скопируем `/home/vasya/katalog` в каталог `/tmp` так: `cp -R /home/vasya/katalog /tmp`. Ключ `-R` указывает на рекурсивное копирование. Как и в случае с файлами, можно поменять конечное имя каталога: `cp -R /home/vasya/katalog /tmp/новое_имя`. У `cp` есть еще один полезный ключ: `-p`. Он предназначен для сохранения всех параметров файла (владельца, прав доступа, битов `suid`, `sgid`). Рекомендуем всегда указывать ключ `-p`, особенно если вы частенько сидите под суперпользователем (чего делать категорически не рекомендуется).

## Команда rm

Предназначена для удаления файлов. Например, удалим файл `/home/vasya/katalog/linux.txt`: `rm /home/vasya/katalog/linux.txt` или если каталог `/home/vasya/katalog` текущий, тогда можно так: `rm linux.txt`. Будьте осторожны, удаленный файл нельзя восстановить, как в Dos или Windows.

Для удаления пустого каталога `/home/vasya/testdir` введите `rmdir /home/vasya/testdir`. Не бойтесь указывать абсолютный путь к каталогу - будет удален только последний элемент пути. Для удаления этого же каталога рекурсивно (вместе с подкаталогами и файлами) введите `rm -r /home/vasya/testdir`.

## Команда df

Предназначена для просмотра оставшегося места на диске `df -h` Ключ `-h` позволяет увидеть вывод в удобном для человека формате (например 1K 234M 2G)

## А как мне быть если путь или имя очень длинные?

Длинные пути к файлам и каталогам - довольно частое явление. И бороться с этим достаточно легко. Все современные оболочки (aka shell) обладают возможностью автодополнения. Что это значит? Например, есть файл `cd /home/vasya/katalog/vtoroi_katalog/oooooooooooooooochendlinoeimyakataloga`. Ясно, что вбивать имя полностью - это сомнительное удовольствие. ;) Поэтому поручим часть работы shell'у. Мы пишем только часть имени, а остальное будет дополнять шелл. Для дополнения необходимо нажать Tab. Например: `cd /h<TAB>/v<TAB>/k<TAB>/v<TAB>/o<TAB>`

Примечание: в оболочке Zsh достаточно написать

`cd /h/v/k/v/o` и нажать "Tab". Оболочка сама развернет сокращение в полный путь. Но тут могут возникнуть проблемы, в том случае, если в каталоге `/home` есть 2 подкаталога `/home/vasya` и `/home/vika`. Шелл не сможет понять куда же вы хотите попасть. Чтобы помочь ему нужно указать еще одну букву в пути, то есть `/home/va<TAB>`, чтобы выбор был однозначным. Если вы не помните части пути, то можете два раза нажать Tab и вы увидите список возможных вариантов. Например: `ls /home/<TAB><TAB>` выведет на экран содержимое директории `/home`.

## Как сделать чтобы программа запущенная из терминала работала после его закрытия?

Такая ситуация возникает довольно часто. Например вам необходимо обновить систему по ssh находясь далеко от своего компьютера, но удерживать связь постоянно вы не можете. Для этой цели можно использовать программу `nohup`. Например, так можно запустить `xmms`: `nohup xmms` Вызываемой программе можно передавать аргументы. Теперь при закрытии терминала `xmms` будет работать, работать и работать ;) или: `xmms &` но это только освободит терминал. После закрытия терминала приложение завершится.

Так же можно использовать программу `screen`.

Графическая оболочка X Window



## А как работать с буфером?

Работа с буфером обмена в системе X Window гораздо удобнее, чем в ОС Windows. Для копирования и вставки текста делаем так:

выделяем нужный текст мышкой

в нужном месте щелкаем на среднюю кнопку или колесико мышки, и текст вставляется там, где курсор.

Конечно, вы можете использовать и комбинации ОС Windows (Ctrl+C, Ctrl+V), однако, следует помнить, что комбинация Ctrl+C экстренно остановит выполнение команды в терминале, другими словами, попытавшись скопировать вывод программы комбинацией Ctrl+C, вы рискуете убить ее выполнение. Будьте внимательнее.

Возможно также использование комбинации Ctrl+Insert для копирования в буфер и Shift+Insert для вставки из него.

## Как заставить иксы стартовать при загрузке?

Для этого их необходимо добавить в уровень запуска default. Для этого необходимо выполнить: `rc-update add xfs default` и `rc-update add xdm default` Первой командой мы заставили автоматически стартовать сервер шрифтов, а вторая указывает на автозапуск менеджера входа в систему. Для того, чтобы вместо xdm запускался другой менеджер входа необходимо немного отредактировать `/etc/rc.conf`.  
Файл: `/etc/rc.conf`

```
..  
DISPLAYMANAGER="kdm"  
..
```

Теперь вместо xdm будет запускаться kdm. Файл: `/etc/rc.conf`

```
..  
DISPLAYMANAGER="gdm"  
..
```

Теперь вместо xdm будет запускаться gdm более красивый менеджер.

# Термины

Мы стараемся придерживаться терминов, уже определенных здесь, а сталкиваясь с новыми трудностями при переводе и редактировании — пополняем словарь.

Кроме того, мы стремимся, чтобы терминология в разных пакетах системы была единой независимо от их происхождения.

---

## **Вики (WiKi) (от гавайского: быстрый)—**

гипертекстовая среда (обычно Веб-сайт) для сбора и структуризации письменных сведений (в программировании: документации, советов, часто задаваемых вопросов(FAQ), мини-руководств(HOWTO)).

Характеризуется следующими признаками:

- Множество авторов. Некоторые вики могут править все посетители.
- Возможность многократно править текст посредством самой вики-среды (вебсайта), без применения особых приспособлений на стороне редактора.
- Проявление изменений сразу после их внесения.
- Разделение информации на страницы, где у каждой своё название.
- Особый язык разметки, позволяющий легко и быстро размечать в тексте структурные элементы, форматирование, гиперссылки и т. п.
- Учёт изменений (учёт версий) текста и возможность отката к ранней версии.

Для создания вики-среды необходимо особое ПО — движок вики. Это частный вид системы управления сайтом, довольно простой в своём устройстве и функциональности, ибо почти все действия по структуризации и обработке сведений делаются пользователями вручную. Главная особенность -- посетители сайта могут сами редактировать содержимое этих страниц и создавать новые. Wiki сделаны так, чтобы редактировать страницы было просто; для редактирования ничего кроме программы веб-браузера не нужно.

---

## **ALSA, или Advanced Linux Sound Architecture —**

продвинутая Звуковая Архитектура Linux, обеспечивает аудио- и MIDI- (Musical Instrument Digital Interface — Цифровой Интерфейс Музыкальных Инструментов) функциональность в операционной системе Linux. ALSA является основной звуковой подсистемой в ядрах 2.6, заменившей собой OSS (Open Sound System - Открытая Звуковая Система), которая использовалась в ядрах 2.4

## **X сервер (или Иксы) -**

обычного пользователя могут испугать мысли о необходимости печатать в командной строке. Почему бы ему не прокладывать свой путь на территории свободы выбора Gentoo (и Linux вообще), просто «водя мышкой»? Ну, конечно, вы сможете это сделать :-). В Linux предлагается широкое разнообразие пользовательских интерфейсов и сред, которые можно установить поверх установленной у вас системы.

Это один из самых больших сюрпризов, с которым сталкиваются новички: графический интерфейс пользователя — это всего лишь прикладная программа, выполняемая в системе. Он не является частью ядра Linux или каких-либо других компонентов системы. Это — мощный инструмент, позволяющий полностью задействовать графические возможности вашего компьютера.

Так как стандартизация очень важна, для организации прорисовки и перемещения окон по экрану, взаимодействия пользователя с мышью и клавиатурой, а также других важных базовых аспектов, был создан стандарт под названием X Window System, сокращённо X11 или просто X. Он используется в Unix, Linux и других Unix-подобных операционных системах во всем мире.

Приложение, которое дает пользователям Linux возможность запускать графические интерфейсы пользователя и использует стандарт X11, называется Xorg-X11, ответвление проекта XFree86. Проектом XFree86 было решено использовать лицензию, возможно, несовместимую с лицензией GPL, поэтому рекомендуется использовать Xorg. Пакет XFree86 больше не поддерживается в официальном дереве портежей.

Проект X.org

Проектом X.org создана и поддерживается бесплатно распространяемая реализация системы X11 с открытым исходным кодом. Это открытая инфраструктура рабочей станции, основанная на X11.

Xorg обеспечивает интерфейс между устройствами вашего компьютера и нужными вам графическими программами. Кроме того, Xorg является полностью сетевым, то есть вы можете запускать программу на одной системе, а отображать ее — на другой.

---

## **ebuild-**

сборочный файл (вероятно, авторское сокращение от emerge build file)

---

## **Оверлей -**

набор ebuild'ов, которые вы хотите добавить к официальной базе данных (дереву) Portage

---

Термин      Перевод

2D acceleration      ускорение двумерной (не двухмерной!) графики

activation      включение; подключение

architecture-specific      архитектурно-зависимый (о режиме или параметре);  
посвященный определенной архитектуре (о руководстве)

acceleration      ускорение

account      учетная запись

advanced      расширенный (напр., настройка);  
углубленный (напр., обучение)

attribute      атрибут (например, файла);  
свойство, параметр (иногда)

authentication      опознание

authorization      допуск

authorized      допущенный; санкционированный

autocompletion      автодополнение (слов в строке)

baselayout    ??? (\*\*)

basics          основы, основные действия

binary          двоичный; бинарный

binary [executable]    исполняемый файл

bind-mounting      привязка??? точки монтирования (создание проекции существующей точки подключения в другое место ФС) (\*\*) (см. комментарий к mount)

bindings      привязки (\*\*)

board of trustees    совет попечителей

bonding (, channel trunking)      объединение ([магистральных] сетевых каналов)

boot runlevel          загрузочный уровень исполнения

bootable (disk)      загрузочный (диск)

bootloader    [начальный] загрузчик; загрузчик ОС

bootsplash    загрузочная заставка

bootsplash-enabled kernel          ядро с [включенной] загрузочной заставкой

boot (bootstrap) partition    загрузочный раздел

bootstrapping      [начальная] загрузка [ОС]; [само]генерация (сборка) системы (в Gentoo; здесь используется нестандартное значение термина)

branch (portage)    ветвь (дерева портежей)

bridge (, network)    [сетевой] мост

broadcast [address]      широковещательный адрес

browse to    открыть [веб-страницу]

browser      браузер

bug    запрос (требующий реагирования); ошибка (разновидность запроса)

bug report    отчет об ошибке или запрос [форма запроса]

bugtracking system      система распределения запросов

(внимание! не багов, не ошибок — потому что не только их; и не отслеживания, управления или контроля — потому что распределения — это корректный перевод одного из американизмов)

cache    кэш

card    плата (деталь компьютера);

карта [PCMCIA, флэш-памяти]

cascading (stacked) profile      каскадный профиль

CD    компакт-диск

change request      запрос на изменение (разновидность запроса)

chapter      глава [руководства]

child [tag]    вложенный [тег] (о XML)

choice      выбор, вариант выбора; альтернатива; пункт меню

(документация по Gentoo помешана на идеологической теме свободы выбора)

пользователя, русский перевод стоит разнообразить)

choices       мн. ч. от choice;

возможность (свобода) выбора

chroot       [команда] chroot;

изменение корневого каталога

chrooted environment   среда с измененным корневым каталогом

color depth   цветность (разрядность цвета видеоплаты)

compose      составлять;

верстать (о разметке XML)

configuration file   конфигурационный файл; файл настроек

configuring; configuration   настройка [системы, чего-либо];

конфигурация [ядра]

content       текст, содержимое;

разделы (о разметке GuideXML)

daemon       демон

(программа слежения за системными событиями; тж. служба)

declare       объявлять (переменную)

default [by -]   по умолчанию

define       задавать, указывать, определять (последний вариант используйте с осторожностью, чтобы не путать определение в смысле установки значения с определением в смысле обнаружения)

desktop (environment)

desktop (computer)

              графическая среда (о KDE/Gnome и т.п.); рабочая станция (настольный компьютер, в отличие от сервера);

desktop documentation resources   документация по графической среде Gentoo;

документация по использованию в качестве рабочей станции

dependency   зависимость

dependency [package]   зависимый пакет

developer   разработчик

device file   файл устройства

device manager   диспетчер устройств

device node   специальный файл [устройства]

dial-in access

dial-up access   [входящий] коммутируемый доступ

directory   каталог

display manager   диспетчер отображения (не экрана!)

documentation listing   перечень документации

documentation repository   хранилище документации

documentation resources документация (имеется в виду спектр доступной документации)

down [interface] отключен

download загрузка (из интернета); загрузить

drive привод

driver драйвер, мн.ч. драйверы

dump программа [резервного копирования] dump;

дамп [памяти]

ebuild сборочный файл; пакет; файл ebuild

программа ebuild

(при переводе транслитерация не допускается из-за исключительной неблагозвучности для русского читателя!) Примечание: e-build — вероятно, авторское сокращение от emerge build file

effort [to centralize] работа [по централизации] (о цели/назначении Gentoo Handbook)

emerge [команда] emerge; средство установки/удаления программ

Примечание: автор Gentoo хотел, чтобы программы «являлись» как по волшебству (изначально система называлась Енох в честь загадочного библейского персонажа); для этого придумал команду «emerge хуз», что буквально переводится как «хуз, явись!». Остальные термины системы установки подогнаны под нее

environment variables переменные среды

experience [, improvement] впечатление (от Gentoo);

[повышение] удобства работы с Gentoo

FAQ,

frequently asked questions FAQ, распространенные вопросы

feature возможность [программы], функция;

особенность

feature request запрос на добавление функции (разновидность запроса)

fetch извлечь (ebuild/portage tree)

firewall межсетевой экран

firmware прошивка; встроенное ПО

floppy [disk] дискета

framebuffer кадровый буфер (используется для промежуточного хранения исходных данных для программного или аппаратного формирования видеокadra или полукadra для вывода на экран монитора)

front-end фасад (часть программы, взаимодействующая с пользователем)

functionality возможности, функции

generic универсальный (о цели компиляции);

общий, обычный (generic printer — обычный принтер)

Gentoo [система] Gentoo (женский или общий род)

Gentoo Documentation Project проект документации Gentoo

Gentoo Handbook настольная книга Gentoo (не руководство)

groupware [сервер, программа, пакет] обеспечения взаимодействия (рабочих групп)

guide руководство (требуется термин, отличный от Handbook);

hard link жесткая ссылка

hardened [Gentoo project] укрепленная система, проект укрепленного Gentoo

hardware аппаратура; устройства

hardware [device] устройство

hardware requirements требования к оборудованию

herd группа [сопровождения пакета]  
(не стадо и не толпа!)

highlighted выделенный (об элементе или о тексте)

host узел, компьютер

howto описание [выполнения чего-то]  
«Как [сделать то-то]» — в заголовке

hub (сетевой) концентратор

image [kernel, disk ...] образ

image [graphics] изображение

init script; initscript сценарий инициализации  
(\*\*)сценарий запуска??? тж. runlevel script

initialization инициализация [переменной];  
подготовка [раздела]

initrd (initial root disk) начальный корневой диск (initrd);  
диск initrd

initrd image образ диска initrd

injecting [into Portage] внедрение [ПО в дерево Portage]

installation установка

installation CD ... установочный [компакт-]диск;  
«...» — тип диска (минимальный, универсальный и т.д.)

installer программа установки

instructions указания; инструкции

internet интернет (с маленькой буквы, м.р., склоняется — интернета, интернетом)

invalid недопустимый

IP address IP-адрес (сетевой адрес протокола интернета в виде последовательности чисел)

issue см. bug

job задание (пакетное; на печать и т.п.)

jumper      переключатель  
 kernel      ядро  
 keymap      раскладка [клавиатуры]  
 laptop      ноутбук, портативный компьютер  
 latest      новейший  
 lightweight      упрощенный (о синтаксисе)  
 linker      компоновщик  
 linking [static]      компоновка  
 linking [dynamic]      динамическое связывание  
 LiveCD      Живой [установочный] [компакт-]диск  
 (установочный, если с него можно установить систему)  
 locale      регион;  
 [набор] региональных настроек  
 (само слово locale означает «местность»)  
 log      журнал [регистрации событий]  
 logger      средства журналирования  
 login [user's]      учетная запись  
 login [process]      вход в систему  
 login [as]      войти в систему [как / под именем]  
 loopback connector      [разъем-] заглушка  
 machine      машина; компьютер  
 maintainer      куратор (пакета и т.п.)  
 man page      страница справки  
 manager      диспетчер (о программе)  
 mark [up]      верстка, оформление, разметка (текста документа XML)  
 mask [package]      замаскировать [пакет] (не «заблокировать»)  
 MBR, master boot record      главная загрузочная запись  
 merge [with; together]  
 merge [software with system]      объединять [с];  
 устанавливать [ПО]  
 modeless      немодальный  
 mount      монтировать; смонтировать [дисковый раздел] (включить в состав файловой системы)

Вопрос к обсуждению: хочется в будущем уйти от этого термина, например, заменив его на подключение/отключение или присоединение/отсоединение.

Термин произошел от процесса установки в накопители больших ЭВМ пакетов дисков (которые действительно надо было монтировать [закреплять], иначе они могли физически улететь, покалечив кого-нибудь), и потерял смысл.

mountpoint      точка монтирования [дискового раздела] (см. комментарий к mount)



multilib      многорежимный профиль (с одновременной поддержкой нескольких наборов библиотек для различных режимов среды пользователя, например, 32- и 64-битного)

naming      именованное

nano    nano (текстовый редактор)

netmask      маска [подсети]

networking    подключение к сети;

сетевое подключение (в контексте настройки узла)

NIS    NIS (network information service — служба сетевой информации)

open documentation license      лицензия открытой документации

option      параметр [командной строки]; директива

вариант [выбора];

возможность [пользователя];

(никогда не опция!)

optional      необязательный [шаг, компонент];

дополнительный;

факультативный

output, [program]    сообщения [программы]

override      подмена, подменить;

переопределение, переопределять

overwrite      перезаписать

package      пакет [программного обеспечения]

package maintenance system      система управления (!) пакетами

packaging system    система сборки (!)

paragraph    абзац (не параграф!)

part    часть [руководства]

(общая структура: часть-глава-раздел-абзац;

раздел намеренно для совместимости подчинен главе в нарушение логики)

partition      [дисковый] раздел; раздел устройства;

(не раздел на устройстве! что-либо записываем тоже «в» раздел!)

partitioning    разбиение

patch    сущ. заплатка, исправление, файл исправлений

patched      исправленный

PC    компьютер (персональный компьютер, если есть противопоставление миниЭВМ)

ping    команда ping;

проверка отклика [командой ping]

pipe    канал (механизм межпроцессной передачи данных)

pool группа; пространство (ресурсов, памяти)

portage [package] портеж;

пакет (системы Portage)

Portage [system] [система] Portage (женский или общий род, без перевода);  
Примечание: для «явления» программ (см. примечание к emerge) автор взял за основу систему port, и предположительно использовал игру слов:

port-portage сохраняется узнаваемость

port-age — «век портов», «зрелый port»

portage — «кипа, связка»

portage tree дерево портежей

portage snapshot снимок дерева портежей

prebuilt [package] готовый [двоичный] пакет

precompiled [package] см. prebuilt [package]

printer-friendly version версия для печати

private key секретный ключ [шифрования]

privilege полномочия

проху прокси-сервер («сервер-посредник» для доступа в интернет)

public key открытый ключ [шифрования]

purpose назначение (в заголовке раздела документа)

quick installation guide краткое руководство по установке

raise [privilege] превышать полномочия

reboot перезагрузка; перезагрузить

reference [manual] справочное руководство

reference [platform] эталонная платформа

release notes замечания к выпуску

release, [software] выпуск [ПО]

release [version] окончательная (стабильная) версия

relevance соответствие (напр., критерию);  
отношение (к чему-либо)

remote удаленный [компьютер];  
дистанционный

rendered document представление документа (о XML)

repository хранилище

rescue [disk] аварийное восстановление;  
диск аварийного восстановления

return [an error] выдавать [ошибку]

root [user] [учетная запись] root; администратор

router (сетевой) маршрутизатор

runlevel      уровень исполнения  
 runlevel script      сценарий инициализации (тж. initscript) (\*\*)  
 script    сценарий  
 section [, guide]      раздел [руководства]  
 security      безопасность; защита  
 security advisory      предупреждение об уязвимости (защиты системы)  
 serial    последовательный  
 service      служба (тж. демон)  
 setting (of variable)    значение; установка значения переменной  
 signature    признак; идентификатор; подпись  
 skel [profile]    скелетный профиль  
 skin    образ; внешний вид? [вариант] оформления? костюм?  
 socket [network]    сокет  
 software      программа; программное обеспечение  
 software management tool      система управления программным обеспечением  
 source code    исходные коды (\*\*) как лучше: коды или код?  
 source mirror      зеркало-источник [Gentoo]  
 (хотя это и зеркало исходных кодов, на нем есть и двоичные образы)  
 specific      определенный, конкретный (очень редко специфичный!)  
 stage    стадия (в Gentoo — генерации операционной системы, вообще — часто установки или загрузки системы)  
 stage file (archive)    файл [архив] стадии; stage-файл [stage-архив]  
 (при упоминании сохранять английское написание файла, чтобы его находили)  
 start [device]      запуск [устройства]  
 statement    выражение  
 stop [device]    остановка [устройства]  
 striping [raid]      чередование (RAID 0, RAID 5)  
 subarch[itecture]    разновидность архитектуры  
 summary    аннотация (краткое описание документа)  
 superuser    администратор (ср. пользователь)  
 supplicant [WPA]    запросчик  
 swap    подкачка  
 switch, [command line]    параметр [командной строки]  
 switch, [network]    (сетевой) коммутатор  
 symlink      символьная ссылка  
 (тж. в других ОС — softlink, alias, shortcut, shadow: псевдоним, ярлык, тень)  
 syntax      синтаксис (выражения);  
 формат, структура (текстового конфигурационного файла)

system information системные параметры (в руководстве по настройке), сведения о системе (при ответе на запрос пользователя)

system, [computer] [компьютерная] система

tag (xml/html) тег (в XML или HTML; при указании на метки вместе с содержимым — элемент); метка (проч.)

tarball архив (набор файлов, помещенный в один файл-контейнер)

target [of compilation] цель [компиляции]

time zone часовой пояс

tips & tricks полезные советы (не применяются выражения: «советы и хитрости», «маленькие хитрости»)

tool средство; утилита;

(иногда, по смыслу) служба

toolchain (\*\*) средства компиляции?

traffic трафик;

движение, поток (употребляется реже)

tweak ухищрение (сущ.)

подстроить (гл.)

unmount размонтировать (см. комментарий к mount)

up [interface] включен

upstream [developers] разработчики самого пакета (входящего в состав Gentoo)

USE flags USE-флаги (признаки использования функций, указываемые в переменной USE)

user учетная запись (о реквизитах пользователя);

пользователь (о человеке)

user-space режим пользователя; [адресное] пространство пользователя

userland режим пользователя (см. user-space) (\*\*)

variable переменная; значение переменной (будьте внимательны!)

verbose подробный; расширенный [вывод, листинг]

подробные сообщения

wallpaper обои (на графическом рабочем столе)

web site веб-сайт

window manager диспетчер окон

wireless (сущ.) беспроводное подключение; беспроводная сеть; беспроводное устройство (по контексту)

wrapper обертка? упаковщик? (\*\*)

Запрещенные термины, или КАК НЕ НАДО

Употребление этих терминов — лучший способ «облегчить» работу редактора: при

этом задержка публикации вашего материала (из-за того, что правка документа будет вызывать непреодолимую грусть) — обеспечена ;-) Неверный перевод  
Правильно

брандмауэр, firewall      межсетевой экран

карта(видео/сетевая/...)    плата

(а карта — игральная [card] или географическая [map])

гид    руководство

(гид — это экскурсовод)

десктоп      настольный (прил.)

[графическая] среда (сущ.)

директория    каталог

(по-русски директория — справочник или орган власти)

ебилд      сборочный файл [ebuild]

(догадываетесь, почему?)

локаль      регион;

региональные настройки

менеджер    диспетчер

опция,

опциональный      возможность; функция; параметр

патч    заплатка, исправление

переменные окружения    переменные среды

(речь идет о среде исполнения)

продвинутый      расширенный, углубленный

релиз      выпуск (о программе)

издание (о книге)

репозиторий документации      хранилище документации

сигнатура    признак, идентификатор, подпись

система уведомления о ошибках и изменениях;

система управления изменениями

система отчета об ошибках      система распределения запросов

скрипт      сценарий

суперпользователь

сверхпользователь      администратор;

[корневая] учетная запись root (root administrative account)

Споры о разности понятий в разных ОС — отставить (я не виноват, что именно в Linux упрощенная модель допуска)

сценарий управления сервисом      сценарий инициализации

(расширение смысла разумно, но это изменение есть смысл вносить в оригинальный текст, иначе будет путаница)

фреймбуфер      кадровый буфер  
(термин устоялся, когда Линус еще не родился)

Текст документа распространяется на условиях лицензии Creative Commons — Attribution / Share Alike.

Этот неофициальный сайт создал и поддерживает Алексей Чумаков. © 2005—2006 проект «русский генту».

Материалы проекта Gentoo © 2001—2006 Gentoo Foundation, Inc.

## MAN (справки по командам)

### ***Emerge***

#### **Имя:**

emerge - интерфейс командной строки к системе портежей (Portage system)

#### **Синтаксис:**

emerge [опции] [действие] [ebuild | файл\_tbz2 | класс | зависимость]...

emerge [опции] [действие] файл\_ebuild

emerge sync | rsync

emerge --help -h [system | config | sync]

emerge --version

#### **Описание:**

***emerge*** - это полный интерфейс командной строки к системе портежей. Используется, в основном, для установки пакетов, причем, emerge может автоматически обрабатывать любые зависимости, заданные в требуемом пакете. Утилита emerge может также обновлять дерево портежей, предоставляя доступ к новым и обновленным пакетам. Утилита emerge также обеспечивает бесппроблемное обновление версий установленных пакетов. Позволяет работать с пакетами как в виде исходных кодов, так и в двоичном виде, также может использоваться для создания двоичных пакетов с целью дальнейшего распространения. Пакеты EBUILD, файлы TBZ2, классы и зависимости

Основное назначение утилиты `emerge` - установка пакетов. Устанавливаемые пакеты можно задать одним из четырех основных способов: как пакет `ebuild`, как файл `tbz2`, как класс или как зависимость.

## **ebuild**

`ebuild` необходимо задавать, как минимум, в виде допустимого имени в каталоге пакетов портежей (valid Portage package directory name) без указания версии или категории, например, `portage` или `python`. Можно добавить также категорию и номер версии, например, `sys-apps/portage` или `python-2.2.1-r2`. Утилита `emerge` игнорирует завершающую косую черту, так что, можно использовать автоматическое завершение для имён файлов. В качестве значения `ebuild` можно также задавать реальное имя файла, например, `/usr/portage/app-admin/python/python-2.2.1-r2.ebuild`.

## **Файл\_tbz2**

Файл `_tbz2` должен быть корректным файлом `.tbz2`, созданным с помощью команды

`ebuild <пакет>-<версия>.ebuild` пакет или

`emerge --buildpkg [категория/]<пакет>`, или

`quickpkg /var/db/pkg/<категория>/<пакет>`.

## **Класс**

Классы пакетов - удобный способ обозначить большую группу пакетов. В настоящее время поддерживается два класса: `system` и `world`. Класс `system` содержит набор пакетов, считающихся необходимыми для правильной работы системы. Класс `world` содержит все пакеты в системе плюс все пакеты, перечисленные в файле `/var/cache/edb/world` или `/var/lib/portage/world` ( $\Rightarrow$  `portage-2.0.51`). Подробнее см. далее в разделе Файлы

Учтите, что эти команды обычно используются вместе с `--update`.

## **Зависимость**

Зависимость описывает ограничения на пакет, который будет установлен. Например, `>=dev-lang/python-2.2.1-r2` что соответствует последней доступной версии языка Python, большей или равной `2.2.1-r2`. Аналогично, `<dev-lang/python-2.0` соответствует последней доступной версии языка Python до `2.0`. Учтите, что во многих командных интерпретаторах потребуется замаскировать символы вроде `'<'` и

'=' ; чтобы избежать проблем с маскировкой, берите зависимость в одиночные или двойные кавычки.

## ***Действия***

### **Нет действия**

Если действие не указано, в систему добавляются указанные пакеты и удовлетворяются все их зависимости. В качестве аргументов можно указывать пакеты ebuild, файлы tbz2, классы или зависимости. Учтите, что для установки файла tbz2 необходимо указать опцию --usrpkg. Пакеты добавляются в конце файла world, так что они учитываются при последующих обновлениях.

### **--clean**

#### **-c**

Очищает систему, удаляя пакеты, не влияющие на функциональность системы. В качестве аргументов можно указывать пакеты build, классы или зависимости. Например,

`emerge clean binutils` удаляет старые версии binutils;

`emerge clean net-www/mozilla-0.9.9-r2` удаляет указанную версию Mozilla.

Использовать эту команду, в общем случае, безопасно. Учтите, что пакеты в единственном экземпляре (unslotted packages) команда clean не удаляет.

### **--depclean**

Выявляет все пакеты, установленные в системе без видимой причины. Утилита emerge генерирует список пакетов, которые должны быть установлены, на основе списка системных пакетов и файла world. Затем она сравнивает этот список со списком фактически установленных пакетов; найденные различия выдаются как ненужные пакеты, которые предлагается удалить. Предупреждение: Удаление некоторых пакетов может привести к тому, что пакеты, связанные с удаленными, перестанут работать и начнут сообщать о недостающих библиотеках. Снова установите с помощью emerge пакет, который перестал работать, и эта проблема будет решена. Учтите, что изменения флагов USE могут существенно повлиять на результат, выдаваемый при выполнении действия depclean.

### **--help**



## **-h**

Выдает справочную информацию по утилите emerge. Добавление одного из прочих перечисленных дополнительных аргументов приводит к выдаче более детальной справочной информации по соответствующему действию. Встроенная справочная документация утилиты emerge обновляется чаще, чем эта страница справочного руководства; обратитесь к ней при возникновении проблем, которые не поможет решить эта страница справочного руководства.

## **--info**

Выдает информацию для включения в отчеты об ошибках. Эта информация помогает разработчикам при решении проблем, о которых вы сообщаете. Включайте, пожалуйста, эту информацию при посылке отчета об ошибке. Расширенную информацию можно получить при указании опции `--verbose`.

При "впрыске" пакета для него вставляется "заглушка" так, что система портежей считает его установленным. Это удобно если необходимо, например, установить двоичную версию XFree86 для нетривиальной видеокарты или просто добавить собственные пакеты. Необходимо указать категорию и конкретную версию вставляемого пакета. Например, `emerge inject sys-kernel/gentoo-sources-2.4.19`

## **--prune**

### **-P**

Предупреждение: Это действие может удалить важные пакеты! Удаляются все, кроме последних, версий соответствующих пакетов. Если для пакета есть несколько слотов, остается только последняя версия в каждом слоте. При этом не происходит проверки зависимостей, поэтому могут быть удалены пакеты, необходимые для нормальной работы системы. Используйте действие `clean`, если не уверены абсолютно в том, что делаете. В качестве аргументов можно указывать пакеты `ebuild`, классы или зависимости -- примеры см. в описании действия `clean`.

## **--regen**

Вызывает проверку и обновление кеша зависимостей для всех пакетов `ebuild` в дереве портежей. Кеш используется для ускорения поиска и построения деревьев зависимостей. Эта команда не рекомендуется для пользователей `rsync`, поскольку `rsync` обновляет кеш, используя кеши на сервере. Если вы не знаете, чем отличается "пользователь `rsync`" от других пользователей, значит, вы - "пользователь `rsync`" :). Пользователи `rsync` должны просто выполнить команду `emerge sync` для регенерации кеша.

## **--search**

### **-s**

Ищет соответствия переданной строке в дереве портежей. Искомая строка представляет собой регулярное выражение. Например, команда

`emerge search "^kde"` ищет любой пакет, имя которого начинается с "kde";

`emerge search "gcc$"` ищет любой пакет, имя которого заканчивается на "gcc";

`emerge search "office"` ищет любой пакет, имя которого содержит слово "office".

Если поиск необходимо выполнять и в описании пакета, используйте опцию `--searchdesc` или `-S`.

### **--sync**

Иницирует обновление дерева портежей с одного из зеркал `rsync.gentoo.org`. Учтите, что любые изменения, выполненные в дереве портежей, при этом будут стерты. За исключением специальных случаев, для обновления используется `rsync`. Чтобы узнать, как избежать удаления изменения, см. описание `PORTDIR_OVERLAY` на странице справочного руководства `make.conf(5)`.

### **--unmerge**

#### **-C**

Предупреждение: При этом могут быть удалены важные пакеты! Удаляются все соответствующие пакеты. Проверка зависимостей не выполняется, поэтому могут быть удалены пакеты, необходимые для нормальной работы системы. В качестве аргументов можно задавать пакеты `ebuild`, классы или зависимости - примеры см. выше в описании `clean`.

### **--ask**

#### **-a**

### **--buildpkg**

#### **-b**

Говорит `emerge` о необходимости построить двоичные пакеты для всех соответствующих пакетов `ebuild`, помимо их обычного добавления в дерево. Полезно для тех, кто занимается поддержкой пакетов или при администрировании нескольких систем Gentoo Linux (строим один раз, устанавливаем пакеты `tbz2` с помощью `emerge` везде). Пакет будет создан в каталоге `${PKGDIR}/All`. В качестве альтернативы для уже включенных в дерево пакетов можно использовать опцию

quickpkg, которая создает пакеты tbz2 из "живой" файловой системы.

### **--buildpkgonly**

**-B**

Создает двоичные пакеты для всех обработанных пакетов ebuild, не включая их фактически в дерево пакетов. Проблема только в том, что все зависимости времени построения уже должны быть включены в систему с помощью emerge.

### **--changelog**

**-l**

Используйте эту опцию вместе с действием --pretend. В результате, будут показаны записи ChangeLog для всех пакетов, которые будут обновляться.

### **--columns**

При использовании совместно с --pretend выдает имена пакетов, новую и старую версию выравненными по столбцам для простоты вырезания и копирования.

### **--debug**

**-d**

Требуется от emerge работать в отладочном режиме. В этом режиме среда построения bash будет работать с опцией -x, выдавая детальную отладочную информацию в стандартный выходной поток. Опция --debug отлично подходит для поиска синтаксических ошибок bash.

### **--deep**

**-D**

При использовании совместно с опцией --update, этот флаг требует от emerge учитывать все дерево зависимостей пакетов, а не просто проверять непосредственные зависимости пакетов. Это, например, позволяет учесть обновления в библиотеках, не перечисленных явно в зависимостях пакета.

### **--emptytree**

**-e**

Создает видимость того, что дерево установленных пакетов содержит только glibc; эту опцию стоит использовать совместно с --pretend. Она позволяет разработчикам получить полное представление о дереве всех зависимостей пакета и перестраивать деревья в целом с использованием новейших версий библиотек.

**--fetchonly**

**-f**

Вместо создания(установки) пакета, просто выбирает(скачивает, загружает) (fetches) в ../distfiles все пакеты (главный пакет и все его зависимости).

**--fetch-all-uri**

**-F**

**--getbinpkg**

**-g**

**--getbinpkg**

**-G**

**--help**

**-h**

**--newuse**

**-N**

используя совместно с опцией --update, позволяет проверить(и переустановить) все пакеты, флаги которых на данный момент были изменены в /etc/make.conf

**--noconfmem**

Требует от портежей не учитывать записи слияния, показывающий, что файл конфигурации в каталоге CONFIG\_PROTECT уже был включен в дерево. Система портежей обычно будет включать такие файлы только раз, чтобы не заставлять пользователя возиться с одной и той же конфигурацией несколько раз. Этот флаг приведет к включению файла конфигурации каждый раз.

**--nodeps**

## **-o**

Включает указанные пакеты, не включая зависимости. Учтите, что процесс построения может закончиться неудачно, если зависимости окажутся не удовлетворенными.

## **--noreplace**

### **-n**

Пропускает указанные в командной строке пакеты, если они уже были установлены. Без этой опции любые указанные в командной строке пакеты, пакеты ebuild или зависимости будут приводить к повторному включению соответствующего пакета системой Portage, даже если он уже установлен. Учтите, что зависимости система Portage по умолчанию повторно не включает.

## **--nospinner**

Отключает для сеанса "крутилку" (spinner). "Крутилка" активируется, если команда подана с терминала (TTY). Этот флаг отключает ее, независимо от устройства.

## **--oneshot**

Включает пакеты как обычно, но не добавляет в профиль world с целью дальнейшего обновления.

## **--onlydeps**

### **-o**

Включает (или делает вид, что включает) зависимости указанных пакетов, а не сами пакеты.

## **--pretend**

### **-p**

Вместо реального включения просто показывает, что должно устанавливаться, если не указать --pretend. Использование опции --pretend настоятельно рекомендуется перед установкой любого незнакомого пакета. В выдаваемом результате:

N = новый

U = обновление

R = замена

B = блокируется уже установленным пакетом.

### **--quiet**

**-q**

Результаты могут отличаться, но, в общем случае, объем выдаваемых системой портежей результатов сокращается или выдача уплотняется.

### **--resume**

Возобновляет последнее действие по включению пакетов. Учтите, пожалуйста, что возобновить можно только действие, закончившееся с ошибкой из-за сбоя. Если делать нечего, будет просто выдано сообщение об успешном выполнении.

### **--searchdesc**

**-S**

Сопоставляет строку поиска с полем описания и именем пакета. Учтите, что сопоставление с регулярным выражением ведется и в описании пакета.

### **--skipfirst**

Это действие допустимо только совместно с --resume. Оно удаляет первый пакет в списке кандидатов на возобновление установки, так, чтобы включение могло продолжиться даже при наличии неисправимой или несущественной ошибки. Эту опцию надо использовать только в случаях, когда пропуск пакета не нарушает зависимости.

### **--tree**

**-t**

### **--update**

**-u**

Обновляет пакеты до самой новой из доступных версий. Учтите, что --update еще не обладает всеми необходимыми возможностями. Он не будет автоматически обновлять зависимости пакетов в файле world, если их самих там еще нет.

### **--upgradeonly**

#### **-U**

Обновляет пакеты, исключая обновления, приводящие к понижению версии устанавливаемого пакета. Слоты (SLOTs) учитываются на простейшем уровне.

### **--usepkg**

#### **-k**

Требует от emerge использовать двоичные пакеты (из каталога \$PKGDIR), если они есть, избегая этим, по возможности, продолжительных компиляций. Эта опция полезна для установок с компакт-дисков; можно выполнить `export PKGDIR=/mnt/cdrom/packages`, а затем использовать эту опцию, чтобы emerge "вытягивала" пакеты с CD для удовлетворения зависимостей.

### **--usepkgonly**

#### **-K**

Работает так же, как и `--usepkg`, но включает только двоичные пакеты. Все необходимые двоичные пакеты должны быть доступны на момент проверки зависимостей или emerge просто прекратит работу.

### **--verbose**

#### **-v**

Требует от emerge работать в многословном (verbose) режиме. В настоящее время этот флаг приводит к тому, что emerge выдает сообщения об ошибках GNU info, если они будут.

### **--version**

#### **-V**

Выдает номер версии emerge. Эту опцию нельзя сочетать с другими; имя и формат действия при этом не важны.

## **Выдаваемые результаты**

При использовании emerge с флагами `pretend/verbose`, результаты на первый взгляд могут показаться сложными для понимания. Этот раздел объясняет, что означает вся эта "стенографическая" информация.

[blocks B ]

[blocks B ] app-text/dos2unix (from pkg app-text/hd2u-0.8.0)

Пакет Dos2unix блокирует включение в дерево пакета hd2u. Блокирование выявляется, когда два пакета перекрываются по файлам или могут вызвать те или иные повреждения в системе. Однако обычно включать одновременно блокирующие друг друга пакеты не нужно, поскольку обычно они обеспечивают одинаковые возможности.

[ebuild N ]

[ebuild N ] app-games/qstat-25c

Пакет Qstat - новый в вашей системе и будет включаться в дерево с помощью emerge впервые.

[ebuild R ]

[ebuild R ] sys-apps/sed-4.0.5

Пакет sed 4.0.5 уже был включен, но если выполнить команду еще раз, система портежей повторно включит этот пакет (в данном случае, sed).

[ebuild F ]

[ebuild F ] media-video/realplayer-8-r6

Пакет realplayer требует, чтобы вы загрузили исходники вручную. При попытке включить пакет в дерево, если исходные тексты пакета не найдены, система портежей остановится и выдаст инструкции о том, как загрузить недостающие файлы.

[ebuild U ]

[ebuild U ] net-fs/samba-2.2.8\_pre1 [2.2.7a]

Пакет Samba 2.2.7a уже был включен и может быть обновлен до версии 2.2.8\_pre1.

[ebuild UD]

[ebuild UD] media-libs/libgd-1.8.4 [2.0.11]

Пакет Libgd 2.0.11 уже был включен, но если выполнить команду, система портежей автоматически понизит (Downgrade) версию пакета до 1.8.4. Это может произойти, если более новая версия пакета была замаскирована, поскольку оказалась



поврежденной или создает угрозу защите системы, а исправление еще не выпущено.

Еще это может произойти потому, что пакет, который вы пытаетесь включить с помощью `emerge` требует для успешного включения более старую версию пакета. В данном случае, библиотека `libgd 2.x` несовместима с `libgd 1.x`. Это означает, что пакеты, созданные с использованием `libgd 1.x`, не скомпилируются с версией 2.x, и ее надо понизить, прежде чем удастся эти пакеты успешно установить.

```
[ebuild U-]
```

```
[ebuild U-] x11-base/xfree-4.3.0 [4.2.99.902]
```

Дефис (-) представляет отсутствие информации о слоте (SLOT information) пакета Xfree. Это происходит, когда предыдущая версия была просто "впрыснута" (подробнее см. описание действия `inject`, или оказалась настолько старой, что соответствующий слот не существует. В любом случае, более новая версия Xfree доступна для обновления.

```
[ebuild U ]
```

```
[ebuild U ] net-analyzer/nmap-3.15_beta2 [3.15_beta1] -gtk -gnome
```

Значения `-gtk` и `-gnome` отражают статус USE-переменных при включении пакета `nmap`. Здесь мы видим, что `nmap` может использовать USE-переменные `gtk` и `gnome`, но в текущих установках переменные `gtk` и `gnome` отключены. Поэтому дополнительная поддержка `gtk` и `gnome` не будет включена в `nmap` при его установке.

#### Примечание

Статус USE выдается только если используются опции `--pretend` и `--verbose`.

## ПРИМЕЧАНИЯ

Практически всегда надо перед любой попыткой установки или обновления выполнять ее же, но с опцией `--pretend`. Это позволяет увидеть, что будет сделано, а также выявить блокирующие пакеты, которые надо будет предварительно удалить. Это тем более важно для классов `system` и `world`, которые могут вызвать обновление большого количества пакетов, если дерево портежей было достаточно активно.

Обычно также имеет смысл использовать опцию `--update`, которая позволяет игнорировать уже полностью обновленные пакеты, но обновлять при этом все остальные. При установке пакета с неустановленными зависимостями, если зависимости эти не указаны явно в списке параметров, они не будут добавляться в файл `world`. Если необходимо внести зависимости в файл `world` для дальнейшего

обновления, не забудьте явно указать их в качестве параметров emerge.

В командной строке можно указывать USE-переменные, переопределяющие значения, указанные в стандартных местах, что позволяет избежать использования ряда нежелательных зависимостей. Флаги USE, заданные в командной строке, НЕ запоминаются. Например, команда

```
USE="-x -gnome" emerge mc
```

установит пакет mc с указанными установками USE.

Если команда emerge --update system или

emerge --update world не сработает с выдачей сообщений об ошибках, причина может быть в том, что пакет ebuild использует новую возможность, которой нет в этой версии утилиты emerge. Можно сначала выполнить

emerge --update portage для обновления до последней версии, которая должна поддерживать любые необходимые новые возможности.

## ЗАМАСКИРОВАННЫЕ ПАКЕТЫ

Примечание: Будьте осторожны при использовании пакетов, находящихся в стадии разработки. Проблемы и ошибки, возникающие в результате неправильного использования замаскированных пакетов, приводят к напрасной трате времени разработчиков Gentoo. Подумайте заранее, сможете ли справиться с проблемами, которые могут возникнуть.

Маски в системе портежей обеспечивают три основные функции: они позволяют задать период тестирования, в течение которого пакеты могут использоваться на машинах; они предотвращают использование пакета, который начал сбоить, и позволяют маскировать существующие пакеты, сломанные или угрожающие защите. Маскирование можно делать двумя способами: с помощью файла package.mask и с помощью ключевых слов. Как демаскировать пакет при использовании этих способов описано ниже. Учтите также, что если утилите emerge передается пакет ebuild, то все виды маскировки игнорируются и emerge будет пытаться включить пакет в систему.

package.mask

Файл package.mask, в основном, блокирует использование пакетов, вызывающих проблемы или известных, как приводящих к проблемам на некоторых машинах. Он находится в каталоге /etc/portage. Пакеты можно демаскировать в файле путем добавления символа комментария (#) в начале строки, маскирующей пакет.

## Ключевые слова

Ключевые слова (KEYWORDS) также используются для маскировки пакетов, находящихся на стадии тестирования. Есть определяющие архитектуру ключевые слова для каждого пакета, позволяющие системе портежей определить, с какими системами совместим пакет. Пакеты, компилирующиеся на некоторой архитектуре, но не считающиеся "стабильными", маскируются путем указания тильды (~) перед названием архитектуры. Утилита emerge проверяет значение переменной среды ACCEPT\_KEYWORDS, прежде чем разрешить или запретить включение пакета, замаскированного с помощью КЛЮЧЕВЫХ СЛОВ.

Чтобы проинформировать утилиту emerge, что необходимо построить эти "тестовые" версии пакетов, можно установить переменной ACCEPT\_KEYWORDS значение "~arch", где arch - одно из значений: x86, ppc, sparc, mips, alpha, arm, hppa. Например, команда ACCEPT\_KEYWORDS="~x86" emerge xfree приведет к тому, что emerge будет учитывать замаскированные ключевыми словами версии XFree в качестве потенциальных кандидатов на обновление.

Предупреждение: Не устанавливайте этой переменной значение, отличающееся от соответствующего вашей архитектуре.

## Как сообщать об ошибках

Пожалуйста, сообщайте о любых обнаруженных ошибках через [bugs.gentoo.org](http://bugs.gentoo.org) веб-сайт

Пожалуйста, включайте в отчет об ошибке результат, который выдает утилита emerge.

## Ссылки

`emerge --help`, `ebuild(1)`, `ebuild(5)`, `make.conf(5)`

Ряд вспомогательных приложений находится в каталоге `/usr/lib/portage/bin`. Пакет `app-admin/gentoolkit` содержит полезные сценарии, например, `qpkg` (средство запросов к пакетам).

## Файлы

`/var/cache/edb/world` (`/var/lib/portage/world` в версиях `>=portage-2.0.51`)

Содержит список всех указанных пользователем пакетов. Можно без опаски редактировать этот файл, добавляя пакеты, которые необходимо учитывать при обновлении класса world, и удаляя те, которые обновлять при этом не нужно.

`/etc/make.conf`

Содержит переменные для процесса построения пакетов, переопределяя значения, заданные в `make.globals`. Именно этот файл надо редактировать вместо перечисленных далее.

`/etc/make.globals`

Содержит стандартные переменные для процесса построения. Не редактируйте этот файл.

`/etc/portage/mirrorsCode: # cat /etc/portage/mirrors`

`local http://192.168.0.10 http://192.168.0.1:8080`

`http://distfiles.gentoo.org http://distro.ibiblio.org/pub/linux/distributions/gentoo`

Содержит список зеркал, на которых будет производится поиск пакетов, запрошенных `emerge`. Стоит отметить, что данный список имеет приоритет выше, чем `GENTOO_MIRRORS` в файле `/etc/make.conf`, следовательно, данный список будет обрабатываться первым, причем зеркала, прописанные под словом `local`, будут обрабатываться самыми первыми.

`/etc/portage/package.keywordsCode: # cat /etc/portage/package.keywords`

`x11-misc/xcompmgr ~x86`

`x11-misc/transset ~x86`

`app-portage/getdelta ~x86`

`app-portage/deltup ~x86`

`dev-util/bdelta ~x86`

`net-im/jabberd ~x86`

`net-wireless/wifiscanner ~x86`

`net-wireless/wlassistant ~x86`

Позволяет установить `ACCESS_KEYWORDS` для конкретных пакетов. Данный файл обрабатывается позже чем `/etc/make.conf`, следовательно его параметры будут актуальнее...

`/etc/portage/package.useCode: # cat /etc/portage/package.use`

x11-wm/fluxbox kde

app-misc/mc -X

Позволяет установить USE флаги для конкретных пакетов. Данный файл обрабатывается позже чем /etc/make.conf, следовательно его параметры будут актуальнее...

/etc/make.profile/package

Содержит список пакетов, используемых для базовой системы. Классы system и world учитывают содержимое этого файла. Не редактируйте его.

/etc/make.profile/make.defaults

Содержит специфические для профиля переменные для процесса построения пакетов. Не редактируйте этот файл.

/etc/make.profile/use.defaults

Содержит список пакетов, которые, в случае установки, приводят к установке по умолчанию соответствующего флага использования (use flag). Не редактируйте этот файл.

/etc/make.profile/virtuals

Содержит список стандартных пакетов, используемых для разрешения виртуальных зависимостей. Не редактируйте этот файл.

/var/cache/edb/virtuals

Содержит список пакетов, используемых для разрешения виртуальных зависимостей. В случае, когда поиск виртуальных зависимостей заканчивается неудачно, можно переупорядочить записи соответствующих виртуальных зависимостей в этом файле. Первый пакет, указанный после virtual, используется для сопоставления с virtual.

/usr/portage/profiles/use.desc

Содержит основной список флагов USE с описаниями их функций. Не редактируйте этот файл.

## Примеры конфигурационных файлов

**Автор:** Евгений Батогов а.к.а JohnBat26 (e-mail: [johnbat26@gmail.com](mailto:johnbat26@gmail.com))

**С версии:** 1.0

**Обновлено:** в 1.3 (31.10.07)

### **Мой компьютер:**

#### **Ноутбук DELL Inspiron 9400:**

- Intel Core 2 Duo 2 Ghz.
- RAM: 2 gb DDR2.
- VIDEO: Nvidia Geforce 7900 GS 256 Mb DDR3
- HDD: 120 Gb.
- Max. Res. 1920X1200 pixels
- FS: Only XFS !

#### **Программное обеспечение:**

- Gentoo Linux.
- Kernel: 2.6.23.
- GCC: 4.2.2.
- Glibc: 2.6.1
- X.org: 1.4.0-r2
- KDE 3.5.8.
- ...

### **1. Мой make.conf**

```
# These settings were set by the catalyst build script that automatically built this stage
# Please consult /etc/make.conf.example for a more detailed example
CFLAGS="-O3 -march=native -mtune=native -msse3 -mfpmath=sse -pipe -fomit-frame-pointer"
CHOST="x86_64-pc-linux-gnu"
CXXFLAGS="${CFLAGS}"
MAKEOPTS="-j3"
USE="* X a52 aac aalib acl acpi aim alsa amarok amuled apm arts asf async
    audiofile automount avi bash-completion bcmath berkdb bitmap-fonts
    bluetooth bzip2 c++ cairo caps cardbus cdda cddb cdinstall cdparanoia cdr
    cdrom cdsound clamav clamd connectionstatus contactnotes cpio
    cpudetection cpulimit cracklib crypt cscope ctype cups curl curlwrappers
```

dba dbus dell depth32 dhcp directfb divx divx4linux djvu dri dv dvb dvd  
 dvdr dvdread dvi dvr3 effects emovix encode exif expat extensions fbcon  
 fbsplash ffmpeg fftw firefox flac flash foomaticdb fortran ftp gd gdb  
 gdbm gif gimp gimpprint ginac glitz gmail gphoto2 gpm graphviz gtk gtk2  
 hal hbcu hddtemp history iconv icq ieee1394 imagemagick imap imlib inkjar  
 interbase ipv6 irc jabber java javascript jbig jpeg jpeg2k kde kdecards  
 kdenlive kdenlive kdenlive kdenlive kdenlive kdenlive kdenlive kdenlive  
 kernel\_linux kipi lame lcmsjpeg ldap ldapsam lha libcaca libclamav libg++  
 libnotify libvisual libwww live lm\_sensors logitech-mouse logrotate magic  
 matroska memlimit mikmod mime mmx mng modplug mozbranding mozdevelop mp3  
 mp4 mp4live mpeg mpeg2 mplayer musepack musicbrains mysql ncurses nfs nls  
 nokia6600 nomotif nptl nptlonly nsplugin ntfs nvidia obex octave office  
 ofx ogg oggvorbis openexr opengl openssl overlays pam pcntl pcre pdf  
 pdflib perl player pmount png posix povray pulseaudio python qt qt-static  
 qt3 qt3support qt4 query-browser quicktime rar rc5 rdesktop readline real  
 realmedia rss ruby samba sasl scanner scanner sdl sensord session  
 sharedext sharedmem simplex xml slang slp smtp sms smtp sndfile soap sockets  
 sound sounds spamassassin speex spell speex sql sqlite3 sse sse2 ssl  
 stats subversion svg symlink sysfs tcl tcltk tcpd theora threads tiff  
 truetype udev unicode unzip usb utf8 vcd videos vim-pager vim-with-x  
 visualization vncviewer vorbis widescreen wifi wireshark wmf wmp xforms  
 xfs xine xinerama xml xml2 xorg xpm xprint xskatcards xsl xv xvid yahoo  
 zip zlib"

ACCEPT\_KEYWORDS="~amd64 ~x86 amd64 x86"

AUTOCLEAN="yes"

PORTAGE\_NICENESS=39

FEATURES="ccache candy userfetch buildpkg -sandbox"

CCACHE\_DIR="/var/tmp/ccache/"

CCACHE\_SIZE="4G"

CC="gcc"

CXX="g++"

#PORTDIR\_OVERLAY="/usr/local/overlays/xeffects /usr/local/layman/xeffects-experimental"

GENTOO\_MIRRORS="ftp://ftp.ussg.iu.edu/pub/linux/gentoo ftp://ftp.ucsb.edu/pub/mirrors/linux/gentoo/  
 http://ftp.ucsb.edu/pub/mirrors/linux/gentoo/ http://gentoo.chem.wisc.edu/gentoo/ ftp://gentoo.mirrors.pair.com/  
 http://gentoo.mirrors.tds.net/gentoo/ ftp://gentoo.mirrors.tds.net/gentoo/ http://gentoo.netnitco.net/  
 ftp://gentoo.netnitco.net/pub/mirrors/gentoo/source/ http://mirror.espri.arizona.edu/gentoo/"

#http://mirrors.acm.cs.rpi.edu/gentoo/ ftp://ftp.ndlug.nd.edu/pub/gentoo/ http://open-  
 systems.ufl.edu/mirrors/gentoo #http://gentoo.llarian.net/ ftp://gentoo.llarian.net/pub/gentoo  
 #http://gentoo.binarycompass.org #http://mirror.datapipe.net/gentoo ftp://mirror.datapipe.net/gentoo  
 http://prometheus.cs.wmich.edu/gentoo #http://modzer0.cs.uaf.edu/public/gentoo/  
 #http://mirror.usu.edu/mirrors/gentoo/ ftp://mirror.usu.edu/mirrors/gentoo/ #ftp://lug.mtu.edu/gentoo  
 http://mirror.phy.olemiss.edu/mirror/gentoo http://mirror.mcs.anl.gov/pub/gentoo/  
 #ftp://mirror.mcs.anl.gov/pub/gentoo/ http://gentoo.mirrors.easynews.com/linux/gentoo/  
 #http://gentoo.cites.uiuc.edu/pub/gentoo/ ftp://gentoo.cites.uiuc.edu/pub/gentoo/

```
#ftp://ftp.www.edu/pub/mirrors/ftp.gentoo.org http://gentoo.localhost.net.ar/
#ftp://mirrors.localhost.net.ar/pub/mirrors/gentoo http://www.las.ic.unicamp.br/pub/gentoo/
#ftp://ftp.las.ic.unicamp.br/pub/gentoo/ http://gentoo.inode.at/ ftp://gentoo.inode.at/source/
#http://gd.tuwien.ac.at/opsys/linux/gentoo/ ftp://gd.tuwien.ac.at/opsys/linux/gentoo/
#http://ftp.belnet.be/mirror/rsync.gentoo.org/gentoo/ ftp://ftp.belnet.be/mirror/rsync.gentoo.org/gentoo/
#http://mirror.bih.net.ba/gentoo/ ftp://mirror.bih.net.ba/gentoo/ #ftp://ftp.sh.cvut.cz/MIRRORS/gentoo/gentoo
#http://gentoo.supp.name/ http://mirror.uni-c.dk/pub/gentoo/ http://ftp.linux.ee/pub/gentoo/distfiles/
#ftp://ftp.linux.ee/pub/gentoo/distfiles/ #http://trumpetti.atm.tut.fi/gentoo/ ftp://trumpetti.atm.tut.fi/gentoo/
#http://ftp.public.fix.fi/gentoo/ ftp://ftp.public.fix.fi/gentoo http://gentoo.modulix.net/gentoo/ #http://ftp.club-
internet.fr/pub/mirrors/gentoo ftp://gentoo.imj.fr/pub/gentoo/ #ftp://ftp.tu-clausthal.de/pub/linux/gentoo/
ftp://sunsite.informatik.rwth-aachen.de/pub/Linux/gentoo #http://linux.rz.ruhr-uni-bochum.de/download/gentoo-
mirror/ "
```

```
#FETCHCOMMAND="/usr/bin/getdelta.sh ${URI}"
SYNC="rsync://rsync.gentoo.org/gentoo-portage"
INPUT_DEVICES="keyboard mouse synaptics evdev"
VIDEO_CARDS="nv nvidia vesa"
LINGUAS="ru"
PORTDIR="/usr/portage"
ALSA_CARDS="hda-intel"
```

## 2. Файл *xorg.conf* для NVIDIA Geforce

### Section "Module"

```
Load      "ddc" # ddc probing of monitor
Load      "dbe" # Double buffer extension
SubSection "extmod"
    Option "omit xfree86-dga" # don't initialize the DGA extension
EndSubSection
Load      "record"
Load      "xtrap"
Load      "type1"
Load      "freetype"
Load      "synaptics"
Load      "glx"
Load      "wfb"
```

### EndSection

### Section "Files"

```
FontPath  "/usr/share/fonts/misc:unscaled"
FontPath  "/usr/share/fonts/Type1/"
FontPath  "/usr/share/fonts/TTF/"
FontPath  "/usr/share/fonts/corefonts"
FontPath  "/usr/share/fonts/freefonts"
FontPath  "/usr/share/fonts/sharefonts"
```



```

FontPath  "/usr/share/fonts/terminus"
FontPath  "/usr/share/fonts/ttf-bitstream-vera"
FontPath  "/usr/share/fonts/unifont"
FontPath  "/usr/share/fonts/100dpi:unscaled"
FontPath  "/usr/share/fonts/75dpi:unscaled"
FontPath  "/usr/share/fonts/artwiz"
FontPath  "/usr/share/fonts/cyrillic"
EndSection

Section "ServerFlags"
    Option  "AllowMouseOpenFail" "true"
    Option  "SuspendTime"        "10"
    Option  "OffTime"            "15"
EndSection

Section "InputDevice"
    Identifier "Keyboard1"
    Driver     "kbd"
    Option     "XkbRules"        "xorg"
    Option     "XkbModel"        "microsoftpro"
    Option     "XkbLayout"       "us,ru"
    Option     "XkbVariant"      ",winkeys"
    Option     "XkbOptions"       "grp:shift_toggle,grp_led:scroll"
    Option     "AutoRepeat"       "500 30"
EndSection

Section "InputDevice"
    Identifier "USBMouse"
    Driver     "mouse"
    Option     "Device"           "/dev/input/mice"
    Option     "Buttons"          "7"
    Option     "CorePointer"
    Option     "Protocol"         "auto"
    Option     "ZAxisMapping"     "4 5 6 7"
EndSection

Section "InputDevice"
    Identifier "Touchpad"
    Driver     "synaptics"
    Option     "SendCoreEvents"   "true"
    Option     "Device"           "/dev/input/mouse0"
    Option     "Protocol"         "auto-dev"
    Option     "LeftEdge"         "1700"

```

```

Option "RightEdge"      "5300"
Option "TopEdge"        "1700"
Option "BottomEdge"     "4200"
Option "FingerLow"      "25"
Option "FingerHigh"     "30"
Option "MaxTapTime"     "180"
Option "MaxTapMove"     "220"
Option "VertScrollDelta" "100"
Option "HorizScrollDelta" "100"
Option "MinSpeed"       "0.09"
Option "MaxSpeed"       "0.18"
Option "AccelFactor"    "0.15"
Option "EdgeMotionMinZ" "17"
Option "EdgeMotionMaxZ" "21"
Option "EdgeMotionMinSpeed" "30"
Option "EdgeMotionMaxSpeed" "35"
Option "LeftRightScrolling" "1"
Option "UpDownScrolling" "1"
Option "EmulateMidButtonTime" "75"
Option "ZAxisMapping"      "4 5"
Option "SHMConfig"         "on"

```

EndSection

#### Section "Monitor"

```

Identifier "DellLFP"
HorizSync  28.0 - 96.0
VertRefresh 43.0 - 60.0
Option     "DPMS"

```

EndSection

#### Section "Device"

```

# Some names might need to be changed hereafter:
Identifier "NVIDIA GeForce 7900GS"
Driver     "nvidia"
VendorName "nVidia Corporation"
BoardName  "GeForce 7900 GS"
BusID      "PCI:1:0:0"
#Option    "NoLogo" "1"
Option "UseDisplayDevice" "DFP"
#### 2D ACCELERATION
#Option "RenderAccel"      "false" # hardware Render acceleration

```

```
#Option "BackinStore"      "true" #[] prevent artifacts?
#Option "NoRenderExtension"  "true"
## some options
Option "SWcursor"          "false" #[]
Option "Render"             "true"
Option "Composite"          "true"
#### 3D ACCELERATION
Option "EnablePageFlip"     "yes" #[] Improves performance
Option "AGPFastWrite"       "yes" #[]
Option "AGPMode"            "4"   # Supports AGP 4x
# VideoRam 262144
# Insert Clocks lines here if appropriate
#Option "IgnoreDisplayDevices" "CRT, TV"
Option "TripleBuffer" "True"
Option "AddARGBGLXVisuals" "True"
```

EndSection

Section "Screen"

```
Identifier "Screen1"
Device "NVIDIA GeForce 7900GS"
Monitor "DellLFP"
DefaultDepth 24
SubSection "Display"
    Viewport 0 0
    Depth 8
    Modes "1920x1200" "1400x1050" "1280x1024" "1024x768" "800x600" "640x480"
EndSubSection
SubSection "Display"
    Viewport 0 0
    Depth 16
    Modes "1920x1200" "1400x1050" "1280x1024" "1024x768" "800x600" "640x480"
EndSubSection
SubSection "Display"
    Viewport 0 0
    Depth 24
    Modes "1920x1200" "1400x1050" "1280x1024" "1024x768" "800x600" "640x480"
EndSubSection
Option "AddARGBGLXVisuals" "true"
```

EndSection

Section "ServerLayout"

```
Identifier "Flat Panel Only"
```

```

Screen    "Screen 1"
InputDevice "Touchpad"    "AlwaysCore"
InputDevice "USBMouse"    "CorePointer"
InputDevice "Keyboard1"    "CoreKeyboard"
EndSection
Section "Extensions"
    Option    "Composite"    "enable"
EndSection

```

### 3. Файл *xorg.conf* для ATI radeon:

```

Section "ServerLayout"
    Identifier    "X.org Configured"
    Screen      0  "Screen0" 0 0
    InputDevice  "Synaptics" "CorePointer"
    InputDevice  "Mouse"     "SendCoreEvents"
    InputDevice  "Keyboard0" "CoreKeyboard"
    Option       "OffTime"   "3"      # 3 indicates number of min until monitor-off
EndSection

Section "ServerFlags"
    Option       "AllowMouseOpenFail"
EndSection

Section "dri"
# Access to OpenGL ICD is allowed for all users:
#   Mode 0666
# Access to OpenGL ICD is restricted to a specific user group:
    Group 27    # video
    Mode 0660
EndSection

Section "Files"
    FontPath    "/usr/share/fonts/misc:unscaled"
    FontPath    "/usr/share/fonts/Type1/"
    FontPath    "/usr/share/fonts/TTF/"
    FontPath    "/usr/share/fonts/corefonts"
    FontPath    "/usr/share/fonts/freefonts"
    FontPath    "/usr/share/fonts/sharefonts"
    FontPath    "/usr/share/fonts/terminus"
    FontPath    "/usr/share/fonts/ttf-bitstream-vera"

```

```
FontPath "/usr/share/fonts/unifont"
FontPath "/usr/share/fonts/100dpi:unscaled"
FontPath "/usr/share/fonts/75dpi:unscaled"
FontPath "/usr/share/fonts/artwiz"
FontPath "/usr/share/fonts/cyrillic"
EndSection
```

#### Section "Module"

```
Load "glx"
# This loads the miscellaneous extensions module, and disables
# initialisation of the XFree86-DGA extension within that module.
SubSection "extmod"
    Option "omit xfree86-dga" # don't initialise the DGA extension
EndSubSection
Load "dbe"
Load "dri"
Load "xtrap"
Load "record"
Load "freetype"
Load "type1"
```

EndSection

#### Section "InputDevice"

```
Identifier "Keyboard0"
Driver "kbd"

Option "XkbModel" "aspire5020"
Option "XkbLayout" "se"
#Option "XkbVariant" "nodeadkeys"
```

EndSection

#### Section "InputDevice"

```
Identifier "Synaptics"
Driver "synaptics"
Option "Device" "/dev/psaux"
Option "Protocol" "auto-dev"
Option "LeftEdge" "1700"
Option "RightEdge" "5300"
Option "TopEdge" "1700"
Option "BottomEdge" "4200"
Option "FingerLow" "25"
Option "FingerHigh" "30"
```

```

Option      "MaxTapTime"    "180"
Option      "MaxTapMove"    "220"
Option      "VertScrollDelta" "100"
Option      "MinSpeed"      "0.09"
Option      "MaxSpeed"      "0.18"
Option      "AccelFactor"    "0.0015"
Option      "SHMConfig"     "on"

```

```

# # Option      "Repeater"    "/dev/ps2mouse"

```

```

EndSection

```

```

Section "InputDevice"

```

```

Identifier  "Mouse"
Driver      "mouse"
Option      "Device"        "/dev/input/mice"
Option      "Protocol"       "imps/2"
Option      "ZAxisMapping"   "4 5"
Option      "Buttons"        "5"

```

```

EndSection

```

```

Section "Monitor"

```

```

Identifier  "Monitor0"
VendorName  "LPL"
ModelName   "0"
Option      "DPMS"

```

```

EndSection

```

```

# === ATI device section ===

```

```

Section "Device"

```

```

Identifier  "Card0"
Driver      "fglrx"
VendorName  "ATI Technologies Inc"
BoardName   "ATI Mobility X600"

```

```

# ### generic DRI settings ###

```

```

# === disable PnP Monitor ===

```

```

#Option      "NoDDC"

```

```

# === disable/enable XAA/DRI ===

```

```

Option "no_accel"      "no"
Option "no_dri"        "no"

```

```

# === misc DRI settings ===

```

```

Option "mtrr"          "off" # disable DRI mtrr mapper, driver has its own code for mtrr

```

```

# ### FireGL DDX driver module specific settings ###

```

```

# === Screen Management ===
Option "DesktopSetup"          "0x00000100"
Option "MonitorLayout"        "NONE,LVDS"
Option "IgnoreEDID"           "off"
Option "HSync2"                "unspecified"
Option "VRefresh2"            "unspecified"
Option "ScreenOverlap"        "0"

# === TV-out Management ===
Option "NoTV"                  "yes"
Option "TVStandard"            "NTSC-M"
Option "TVHSizeAdj"           "0"
Option "TVVSizeAdj"           "0"
Option "TVHPosAdj"            "0"
Option "TVVPosAdj"            "0"
Option "TVHStartAdj"          "0"
Option "TVColorAdj"           "0"
Option "GammaCorrectionI"      "0x00000000"
Option "GammaCorrectionII"     "0x00000000"

# === OpenGL specific profiles/settings ===
Option "Capabilities"          "0x00000000"

# === Video Overlay for the Xv extension ===
Option "VideoOverlay"          "on"

# === OpenGL Overlay ===
# Note: When OpenGL Overlay is enabled, Video Overlay
# will be disabled automatically
Option "OpenGLOverlay"         "off"

# === Center Mode (Laptops only) ===
Option "CenterMode"            "off"

# === Pseudo Color Visuals (8-bit visuals) ===
Option "PseudoColorVisuals"    "off"

# === QBS Management ===
Option "Stereo"                "off"
Option "StereoSyncEnable"      "1"

# === FSAA Management ===
Option "FSAAScale"             "1"
Option "FSAADisableGamma"      "no"
Option "FSAACustomizeMSPos"     "no"
Option "FSAAMSPosX0"           "0.000000"
Option "FSAAMSPosY0"           "0.000000"
Option "FSAAMSPosX1"           "0.000000"
Option "FSAAMSPosY1"           "0.000000"

```

```

Option "FSAAMSPosX2"      "0.000000"
Option "FSAAMSPosY2"      "0.000000"
Option "FSAAMSPosX3"      "0.000000"
Option "FSAAMSPosY3"      "0.000000"
Option "FSAAMSPosX4"      "0.000000"
Option "FSAAMSPosY4"      "0.000000"
Option "FSAAMSPosX5"      "0.000000"
Option "FSAAMSPosY5"      "0.000000"
# === Misc Options ===
Option "UseFastTLS"        "0"
Option "BlockSignalsOnLock" "on"
Option "UseInternalAGPGART" "no"
Option "ForceGenericCPU"   "no"
Option "DynamicClocks"      "on" # Use ATI Powerplay features
BusID "PCI:1:0:0"  # vendor=1002, device=3150
Screen 0
EndSection

#Section "Device"
### Available Driver options are:-
### Values: <i>: integer, <f>: float, <bool>: "True"/"False",
### <string>: "String", <freq>: "<f> Hz/kHz/MHz"
### [arg]: arg optional
#Option  "NoAccel"          # [<bool>]
#Option  "SWcursor"         # [<bool>]
#Option  "Dac6Bit"          # [<bool>]
#Option  "Dac8Bit"          # [<bool>]
#Option  "BusType"          # [<str>]
#Option  "CPPIOMode"        # [<bool>]
#Option  "CPusecTimeout"    # <i>
#Option  "AGPMode"          # <i>
#Option  "AGPFastWrite"     # [<bool>]
#Option  "AGPSize"          # <i>
#Option  "GARTSize"         # <i>
#Option  "RingSize"         # <i>
#Option  "BufferSize"       # <i>
#Option  "EnableDepthMoves" # [<bool>]
#Option  "EnablePageFlip"   # [<bool>]
#Option  "NoBackBuffer"     # [<bool>]
#Option  "DRIRemote"        # [<bool>]
#Option  "PanelOff"         # [<bool>]
#Option  "DDCMode"          # [<bool>]

```



```

#Option "MonitorLayout"      # [<str>]
#Option "IgnoreEDID"         # [<bool>]
#Option "UseFBDev"           # [<bool>]
#Option "VideoKey"           # [<i>]
#Option "MergedFB"           # [<bool>]
#Option "CRT2HSync"          # [<str>]
#Option "CRT2VRefresh"       # [<str>]
#Option "CRT2Position"       # [<str>]
#Option "MetaModes"          # [<str>]
#Option "MergedDPI"          # [<str>]
#Option "NoMergedXinerama"   # [<bool>]
#Option "MergedXineramaCRT2IsScreen0" # [<bool>]
#Option "DisplayPriority"     # [<str>]
#Option "PanelSize"          # [<str>]
#Option "ForceMinDotClock"   # [<freq>]
#Option "RenderAccel"        # [<bool>]
#Option "SubPixelOrder"      # [<str>]
#Option "ShowCache"          # [<bool>]
#Option "DynamicClocks"      # [<bool>]
#Option "VGAAccess"          # [<bool>]
#Option "LVDSProbePLL"       # [<bool>]
#Option "ReverseDDC"         # [<bool>]
#Option "BIOSHotkeys"        # [<bool>]

# Identifier "Card0"
# Driver "ati"
# VendorName "ATI Technologies Inc"
# BoardName "ATI Mobility X600"
# BusID "PCI:1:0:0"
#EndSection

Section "Screen"
    Identifier "Screen0"
    Device "Card0"
    Monitor "Monitor0"
    DefaultDepth 24

    SubSection "Display"
        Viewport 0 0
        Depth 16
#        Modes "1280x800"
    EndSubSection
    SubSection "Display"

```

```
Viewport 0 0
Depth 24
# Modes "1280x800"
EndSubSection
EndSection
```

#### **4. Файл [/etc/fstab](#)**

## Важные команды Gentoo Linux

1. **`emerge --ask --verbose`** (или **`emerge -av`**) - просмотр USE-флагов перед началом компиляции.
2. **`emerge имя_пакета`** – установка пакета.
3. **`emerge sync`** – обновление локального дерева портежей.
4. **`echo "category/some_package some_flags" >> /etc/portage/package.use --`** прописывание USE-флагов в `/etc/portage/package.use`, для этого выполните

## Полезные утилиты Gentoo Linux

- Пакет **gentoolkit** (для установки наберите: **emerge gentoolkit**) содержит:
  - а) **euse**, которая существенно упрощает управление флагами USE. Выполните **euse -i flag** для получения информации о флаге "flag". Так же просмотрите map-страницу euse (не волнуйтесь. она короткая :-)).
  - б) **eclean**. Она поможет удалить не нужные файлы. Эта утилита может работать как с исходниками, так и с бинарными пакетами. (просто вызовите её как **eclean-dist** или **eclean-pkg** соответственно). Прочитайте страницу map для получения полного списка опций.
- Утилита **genlop** (для установки наберите: **emerge genlop**) позволяет извлечь множество различной информации из этого файла. Например genlop -с покажет вам какой пакет собирается в данный момент и попытается предсказать сколько времени осталось до конца сборки.
- Утилита **ufed** (для установки наберите: **emerge ufed**) позволяет просматривать и устанавливать USE флаги в более дружелюбном интерфейсе.
- Скрипт **update-world** (ссылка: <http://www.gentoo.org/news/ru/gwn/20061204-newsletter.xml> Пункт 3. Полезные советы / Простейший путь обновления своей системы) позволяет оставлять обновляющуюся систему не боясь, что обновление прервется на каком-либо пакете. Данный скрипт просто продолжит обновление со следующего пакета.
- Оконные приложения:
  - **kuroo** (написана на **Qt**, установка: **emerge kuroo**);
  - **porthole** (написана на **GTK**, установка: **emerge porthole**);
  - **portagemaster** (написана на **Java**, установка: **emerge portagemaster**);предлагают доступ к **portage** в оконном режиме. Выбирайте на свой вкус! :)
- Общая Linux утилита **lshw – hardware list** (**emerge lshw**) показывает информацию о аппаратном обеспечении компьютера.
- Если необходимо заново сгенерировать файл world используйте команду **regenworld**.
- Если необходимо пересобрать модули ядра после компиляции нового ядра, то воспользуйтесь утилитой **module-rebuild** (**emerge module-rebuild**)



## Ресурсы с документацией

1. <http://gentoo.ru>
2. <http://gentoo.org>
3. <http://ru.gentoo-wiki.com>
4. <http://www.rugentoo.org>
5. <http://linuxforum.ru>

## Ресурсы с программным обеспечением

1. <http://packages.gentoo.org>
2. <http://gentoo-portage.com>

## Каналы IRC (freenode.net)

1. `#gentoo.`
2. `#gentoo-amd64`
3. `#gentoo-kde`
4. `#gentoo-ru.`
5. `#rugentoo.`
6. `#gentoo-doc-ru`