

Сергей Волох



Ubuntu Linux с нуля



- Установка и настройка Ubuntu Linux
- Графический режим и командная строка
- Установка драйверов оборудования
- Работа с периферийными устройствами
- Установка программ и решение возникших проблем
- Настройка подключения к Интернету
- Поиск и устранение неисправностей работы сети
- Работа со службами
- Устранение сбоев системы



Сергей Волох

Ubuntu Linux с нуля

Санкт-Петербург
«БХВ-Петербург»
2018

УДК 004.451
ББК 32.973.26-018.2
В68

Волох С. В.

В68 Ubuntu Linux с нуля. — СПб.: БХВ-Петербург, 2018. — 400 с.: ил.
ISBN 978-5-9775-3953-1

С позиции пользователя рассмотрена работа в популярной операционной системе Ubuntu Linux, в том числе установка и настройка системы, драйверов для работы с принтерами, сканерами и другим периферийным оборудованием. Описан процесс установки программ и настройка подключения к Интернету. Даны советы по использованию графического режима и командной строки. Описаны системные службы Ubuntu Linux, освещены вопросы по пользовательским правам доступа. Приведена информация по поиску неисправностей работы сети и устранению сбоев системы. Материал книги актуален и применим для работы с другими Linux-подобными операционными системами.

Для широкого круга пользователей Linux

УДК 004.451
ББК 32.973.26-018.2

Группа подготовки издания:

Руководитель проекта	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Екатерина Капалыгина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Дизайн обложки	<i>Марины Дамбиевой</i>

Подписано в печать 31.05.18.
Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 32,25.
Тираж 1000 экз. Заказ № 6778.
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.
Отпечатано с готового оригинал-макета
ООО "Принт-М", 142300, М.О., г. Чехов, ул. Полиграфистов, д. 1

ISBN 978-5-9775-3953-1

© ООО "БХВ", 2018
© Оформление. ООО "БХВ-Петербург", 2018

Оглавление

Предисловие	11
Для кого эта книга	12
Как устроена книга	12
Как читать эту книгу.....	13
От автора	13
Благодарности.....	14
Введение	15
История появления ОС Linux	16
GNU и Linux	17
Появление различных дистрибутивов.....	18
Дистрибутивы Linux	19
Red Hat Enterprise Linux	20
Fedora Linux.....	20
Debian.....	20
Ubuntu	20
Linux Mint	21
Slackware.....	21
Логотип Linux	21
Область применения Linux	22
Идеология свободного ПО.....	23
Безопасен ли открытый исходный код?.....	24
Критика и очевидные достоинства ОС Linux.....	25
ЧАСТЬ I. ЗНАКОМЬТЕСЬ: UBUNTU.....	27
Глава 1. Подготовка к установке ОС Ubuntu Linux.....	29
ОС Ubuntu Linux	29
Версии дистрибутивов ОС Ubuntu	29
Версии LTS и не-LTS.....	30
Варианты установки ОС Ubuntu Linux	30
Порядок установки операционных систем	31
Как получить дистрибутив ОС Ubuntu?.....	31
Запись образа диска с ОС Ubuntu.....	33

Создание загрузочного флеш-накопителя	34
Файловые системы ОС Ubuntu Linux	36
Файловые системы ext2, ext3 и ext4	36
Файловая система Btrfs	37
Файловая система XFS	37
Файловые системы ReiserFS и Reiser4	37
Файловая система JFS	38
Выводы	38
Глава 2. Теоретический минимум о файловой системе в ОС Ubuntu Linux	39
Разделы жесткого диска	39
Первичные, расширенные и логические разделы	40
Точки монтирования	40
Точки монтирования в ОС Ubuntu	41
Очередность создания разделов	42
Объем раздела /swap	43
Объем корневого раздела	43
Объем раздела /home	43
Подготовительные действия перед установкой	44
Выводы	44
Глава 3. Создание разделов для установки ОС Ubuntu Linux	45
Выбор типа носителя для загрузки операционной системы в меню BIOS	45
Особенности установки на платах с UEFI	46
Варианты запуска и установки ОС Ubuntu	46
Запуск ОС Ubuntu с загрузочного флеш-накопителя	47
Запуск ОС Ubuntu с DVD-привода	48
Для чего нужен Live-режим?	49
Запуск программы GParted	49
Возможности программы GParted	50
Разметка жесткого диска в программе GParted	51
Выводы	57
Глава 4. Установка ОС Ubuntu Linux	58
Запуск операционной системы в Live-режиме	58
Установка операционной системы из Live-режима	59
Выводы	66
Глава 5. Первый запуск и первоначальная настройка ОС Ubuntu	67
Запуск ОС Ubuntu	67
Настройка подключения к Интернету в Ubuntu	70
Подключение к сети по Wi-Fi	70
Настройка подключений в NetworkManager	71
Если не определилась сетевая карта	72
Создание подключения Ethernet	73
Подключение через DSL	74
Особый случай подключения через DSL	76
Подключение с помощью USB-модема	82
Возможные проблемы с созданием подключений	85
Выводы	87

Глава 6. Установка обновлений и локализация операционной системы.....	88
Установка обновлений	88
Установка пакетов локализации.....	91
Выводы	93
Глава 7. Обзор интерфейса ОС Ubuntu.....	94
Графическая среда рабочего стола.....	94
Рабочий стол в Unity.....	94
Панель меню	95
Панель запуска	97
Рабочее место.....	98
Главное меню	100
Меню HUD	101
Выводы	102
Глава 8. Настройка операционной системы и решение проблем.....	103
Параметры системы.....	103
Настройка раскладки клавиатуры	104
Смена системного времени.....	105
Настройка спящего режима	107
Пустой рабочий стол	108
Изменение порядка загрузки операционной системы	109
Выводы	111
ЧАСТЬ II. UBUNTU ДЛЯ ПОЛЬЗОВАТЕЛЯ.....	113
Глава 9. Структура файловой системы	115
Файловый менеджер Nautilus	115
Имена устройств	117
Структура файловой системы.....	119
Общее описание структуры каталогов.....	120
Корневая файловая система.....	123
Домашний каталог пользователя.....	123
Каталог /bin/	124
Каталог /boot/	124
Каталог /cdrom/	124
Каталог /dev/.....	124
Каталог /etc/	125
Каталог /home/.....	125
Каталоги /lib/ и /lib64/.....	126
Каталог /lost+found/	126
Каталог /media/	126
Каталог /mnt/	127
Каталог /opt/	127
Каталог /proc/	127
Каталог /root/	127
Каталог /run/	128
Каталог /sbin/	128
Каталог /snap/	128

Каталог /srv/	128
Каталог /sys/	128
Каталог /tmp/	129
Каталог /usr/	129
Каталог /var/	129
Выводы	129
Глава 10. Типы файлов в ОС Ubuntu	130
О файлах и каталогах	130
Содержимое и расширение файлов	131
Текстовые файлы	132
Исполняемые файлы	134
Прочие файлы	135
Изменение ассоциации файлов	135
Выводы	137
Глава 11. Администратор и суперпользователь root	138
Привилегированный пользователь root	138
Обычная учетная запись и запись администратора	139
Привилегии администратора	141
Административные задачи в текстовом режиме	143
Выводы	143
Глава 12. Знакомство с Консолью и Терминалом	144
Что такое Терминал и Консоль	144
Запуск Консоли	145
Запуск Терминала	146
Синтаксис команд в Терминале	148
Команда <i>echo</i>	148
Просмотр версии ядра	149
Установка текущего каталога	149
Просмотр текущего каталога	150
Просмотр содержимого каталога	151
Просмотр справки по командам	152
Подсказки Терминала	154
История введенных ранее команд	154
Очистка окна Терминала	155
Остановка выполнения команды	155
Выход из Терминала	156
Выводы	156
Глава 13. Полезные команды Терминала	157
О пользе работы в Терминале	157
Команда <i>whoami</i> , или "Кто я?"	157
Команда <i>who</i>	158
Создание файлов	159
Создание каталогов	162
Удаление каталогов	163
Удаление каталогов и файлов	164
Опасность выполнения команды <i>rm -rf/</i>	165

Копирование файлов и каталогов.....	166
Перемещение файлов и каталогов.....	171
Вывод содержимого файла.....	173
Вывод содержимого файла в обратной последовательности.....	175
Команды <i>more</i> и <i>less</i> для просмотра содержимого файлов.....	177
Команда поиска файлов.....	181
Отображение каталога программы.....	183
Поиск текстового шаблона в файлах.....	183
Управление остановкой системы.....	191
Выводы.....	193
Глава 14. Работа с привилегиями root.....	194
О привилегиях root.....	194
Команда <i>sudo</i> вместо прав root.....	195
Что использовать: <i>sudo</i> или <i>su</i> ?.....	195
Команда <i>sudo</i>	196
Запуск графических программ от имени администратора.....	200
Настройка <i>sudo</i> и прав доступа на выполнение команд.....	202
Время действия введенного пароля.....	205
Выводы.....	206
Глава 15. Установка драйверов в ОС Ubuntu.....	207
Драйверы в ОС Ubuntu.....	207
Установка сетевой карты Realtek.....	208
Установка драйвера беспроводной сети Broadcom.....	212
Установка драйвера видеокарты из репозитория.....	216
Общий метод установки видеодрайвера AMD/ATI Radeon.....	217
Установка аудиодрайвера.....	219
Установка ALSA из исходных текстов.....	221
Выводы.....	225
Глава 16. Установка принтера в ОС Ubuntu.....	226
Об установке принтеров, сканеров и МФУ.....	226
Где взять необходимые драйверы?.....	226
Устройства Brother.....	227
Устройства Canon.....	227
Устройства Epson.....	227
Устройства HP.....	227
Устройства Samsung.....	227
Устройства Xerox.....	228
Установка службы CUPS.....	228
Установка принтера Epson L800 средствами ОС.....	229
Настройка общего доступа.....	233
Установка МФУ Epson Stylus CX5900 с помощью драйверов.....	235
Выводы.....	240
Глава 17. Установка и удаление программ в ОС Ubuntu.....	241
Особенности установки и удаления программ.....	241
Зависимости пакетов в ОС Ubuntu Linux.....	242
Установка приложений с помощью Менеджера приложений Ubuntu.....	243

Установка приложений из deb-пакетов.....	246
Установка deb-пакетов с помощью Терминала.....	248
Установка приложений из Терминала	251
Установка приложений из исходных текстов.....	254
Репозитории в ОС Ubuntu	258
Удаление программ	262
Выводы	264
Глава 18. Установка дополнительного программного обеспечения	265
Дополнительное ПО в Ubuntu	265
Дополнительные медиамодули.....	265
Мультимедийные проигрыватели	267
Web-браузеры	269
BitTorrent-клиент	269
Файловый менеджер.....	270
Редактирование изображений.....	270
Запись CD/DVD-дисков	271
Выводы	271
Глава 19. Запуск Windows-приложений в ОС Linux	272
Wine для запуска приложений ОС Windows	272
Установка Wine.....	272
Установка и запуск Windows-приложений.....	275
Выводы	278
Глава 20. Работа с архивами в ОС Ubuntu.....	279
Об архивах в ОС Ubuntu.....	279
Менеджер архивов.....	279
Несжатые и сжатые архивы	281
Архиватор tar.....	282
Архиватор gzip	286
Архиватор bzip2	288
Прочие архиваторы	288
Неизвестный тип файла.....	290
Выводы	290
ЧАСТЬ III. UBUNTU ДЛЯ ОПЫТНОГО ПОЛЬЗОВАТЕЛЯ	291
Глава 21. Группы пользователей и права доступа.....	293
Пользовательские группы.....	293
Права доступа в Linux	293
Буквенное представление прав доступа к файлу	295
Буквенное представление прав доступа к каталогам.....	297
Цифровое представление прав доступа	300
Символьные биты SUID, SGID и sticky	301
Права доступа по умолчанию	304
Смена прав	305
Смена владельца файла	308
Смена атрибутов файла	308

Просмотр атрибутов	310
Выводы	311
Глава 22. Жесткие и символические ссылки	312
Ссылки в ОС Linux	312
Типы ссылок.....	313
Создание ссылок	314
Работа с символическими ссылками.....	316
Работа с жесткими ссылками.....	318
Выводы	319
Глава 23. Монтирование файловых систем	320
Точка монтирования.....	320
Монтирование устройств в ОС Ubuntu	320
Способы постоянных именований	321
Именование посредством UUID	323
Именование посредством имен устройств	324
Именование посредством меток устройств	324
Именование посредством Hardware ID	326
Универсальный метод получения информации о блочных устройствах	327
Автоматическое монтирование	327
Ручное монтирование и размонтирование устройства	331
Выводы	332
Глава 24. Уровни выполнения и цели в ОС Ubuntu	333
Инициализация системы	333
Уровни выполнения.....	333
Современная система уровня запуска.....	335
Выводы	338
Глава 25. Службы в ОС Ubuntu.....	339
Службы, сервисы и демоны	339
Службы в ОС Ubuntu	339
Управление службами в ОС Ubuntu	340
Автозагрузка служб и отключение служб	345
Управление питанием.....	346
Управление питанием ACPI.....	347
Ведение логов в systemd.....	348
Выводы	349
Глава 26. Управление процессами в ОС Ubuntu	350
Процессы в ОС Ubuntu	350
Системный монитор процессов <i>top</i>	350
Получение списка процессов.....	354
Завершение процесса командой <i>kill</i>	356
Завершение процесса командой <i>killall</i>	358
Управление процессами в Консоли	360
Прочие инструменты по управлению процессами	361
Выводы	362

Глава 27. Проверка работоспособности сети в ОС Ubuntu	363
Сеть в ОС Ubuntu	363
Определение сетевого оборудования	363
Проверка соединения с Интернетом	365
Трассировка маршрута	367
Выводы	369
Глава 28. Подключение к удаленному рабочему столу	370
Удаленный рабочий стол	370
Подключение к рабочему столу Windows	370
Подключение с помощью TeamViewer	374
Выводы	376
Глава 29. Восстановление загрузчика GRUB	377
Сбой в работе загрузчика	377
Восстановление загрузчика GRUB	377
Простой способ восстановления загрузчика GRUB	380
Восстановление GRUB в консоли загрузчика	382
Выводы	384
Глава 30. Увеличение разрешения ОС Ubuntu в VirtualBox	385
ОС Ubuntu в виртуальной машине	385
Виртуальная машина Oracle VM VirtualBox	385
Низкое разрешение экрана в виртуальной машине	386
Установка дополнений	386
Выводы	390
Заключение	391
Приложение. Описание электронного архива	393
Предметный указатель	395

Предисловие

Вы держите в руках книгу о популярной операционной системе Ubuntu Linux. Это не очередная книга по описанию возможностей операционной системы Ubuntu и не очередной скучный справочник. Настоящая книга является практическим руководством к действию, которого так часто не хватает желающим познать удивительный мир ОС Ubuntu Linux.

Сейчас на полках книжных магазинов представлено великое множество книг схожей тематики, читатель теряется в выборе качественной литературы и покупает первое попавшееся издание, которое зачастую не оправдывает его ожиданий.

В данной книге ОС Ubuntu Linux рассмотрена с самого начала: с момента установки до восстановления ее работоспособности в случае программного сбоя. Подробно описывается весь жизненный цикл работы с этой ОС, т. к. это и нужно читателю, впервые столкнувшемуся с данной операционной системой. Весь материал книги выстроен в строгой логической последовательности, ровно так, как это нужно читателю для полноценного изучения операционной системы Ubuntu Linux. Жизненный цикл подразумевает период от загрузки дистрибутива с сайта разработчика операционной системы, ее установки, настройки, добавления драйверов и программного обеспечения до восстановления ОС в случае ее краха, который нередко возникает в процессе работы. Читателю не нужно приобретать отдельную книгу для того, чтобы понять тонкости работы с операционной системой и повысить свой уровень знаний после поверхностного знакомства с рассматриваемой операционной системой.

Сведения, которые вы почерпнете после прочтения данной книги, не только пригодятся вам в повседневной работе с данной операционной системой, но и будут базисом вашей деятельности как IT-специалиста в данной отрасли. Кроме того, полученные вами знания в результате прочтения данной книги можно применить не только в ОС Ubuntu Linux, но и в других дистрибутивах на базе ядра Linux. Это означает, что прочитав данную книгу, вы с легкостью можете "мигрировать" на любой другой понравившийся вам дистрибутив Linux, и вам не потребуется покупать еще одну книгу, чтобы разобраться в нем.

Для кого эта книга

В первую очередь, книга предназначена для тех, кто хочет быстро и полноценно изучить рассматриваемую операционную систему, т. е. для вас. Вы же не просто так заглянули в эту книгу?

Для новичка в мире Linux данная книга окажется как нельзя кстати. Новичку не потребуется искать дополнительную информацию на возникающие вопросы в процессе знакомства с Ubuntu Linux. Все что нужно начинающему пользователю, есть в этой книге.

Книга будет полезна и тем, кто уже хоть немного знаком с операционной системой Ubuntu Linux либо с другой ОС на базе Linux. Пользователю, знакомому с рассматриваемой операционной системой, будет полезна информация о работе с Терминалом, установке драйверов, а также будет интересно узнать о структуре файловой системы и правах учетных записей.

Если же вы опытный пользователь и хотите повысить свои навыки работы в ОС Ubuntu Linux, то с помощью этой книги научитесь контролировать права доступа, узнаете о символьных битах, научитесь управлять процессами и службами ОС Ubuntu Linux и т. п.

Вас ждет приятное и увлекательное знакомство с операционной системой Ubuntu Linux.

Как устроена книга

В книге рассматривается настольная версия операционной системы Ubuntu, которая принадлежит к семейству операционных систем Linux.

Изначально Linux разрабатывалась для профессионалов, о чем и говорит применение этой ОС в качестве серверной операционной системы. Однако с появлением различных дистрибутивов ситуация стала меняться в корне, и операционная система Linux начала завоевывать сердца пользователей. Одной из таких операционных систем, в силу своей простоты и надежности, явилась рассматриваемая в этой книге ОС Ubuntu Linux.

Книга состоит из трех частей. Материал каждой части нацелен на определенный уровень подготовки и знаний. Так, *часть I* предназначена для тех, кто еще только приступает к знакомству с операционными системами и хочет начать его именно с ОС Ubuntu Linux. На этом уровне подготовки у читателя еще нет необходимых знаний и сведений об изучаемой операционной системе, поэтому в книге есть все, что может понадобиться начинающему пользователю.

Часть II предназначена не только для тех, кто прочитал и изучил предыдущий материал, но и для тех, кто уже знаком с любым дистрибутивом на базе операционной системы Linux (не только ОС Ubuntu Linux). Обычный пользователь в этой части найдет много интересного и познавательного.

И наконец, *часть III* предназначена как и для первых двух категорий пользователей, так и для продвинутого пользователя, который желает более детально погрузиться в изучение мира операционной системы Ubuntu Linux.

Как читать эту книгу

Как и любую другую, эту книгу рекомендуется читать с первых страниц, т. к. изучение материала в ней построено в строгой логической последовательности. Например, прежде чем приступить к установке операционной системы читатель должен узнать о разделах жесткого диска, о первичных и расширенных логических дисках и их отличиях. Однако если вы уже немного знакомы с ОС Ubuntu Linux и желаете повысить свой уровень знаний, то можете сразу приступить к чтению *части II "Ubuntu для пользователя"*.

Читателям, уже знакомым с операционной системой Ubuntu Linux и желающим познать внутренний мир ОС Linux, будет полезна *часть III "Ubuntu для опытного пользователя"*. Здесь вы найдете много полезной для себя информации об особенностях работы ОС, знание которых необходимо для лучшего понимания раскрываемой в книге тематики, для дальнейшего самосовершенствования в качестве специалиста в данной отрасли.

От автора

На полках книжных магазинов представлено великое множество литературы по рассматриваемой в этой книге теме. Однако причиной написания "еще одной книги по линуксу" послужил тот фактор, что большинство современных книг не имеют четкого и последовательного изложения материала. Новички просто теряются в выборе книг и берут первую попавшуюся, а потом просто забрасывают изучение новой операционной системы. Поэтому я решил написать такое руководство, которое не только без труда поможет вам освоить азы операционной системы Ubuntu Linux, но и обеспечит базис знаний, который послужит мощным фундаментом для более глубокого изучения этой операционной системы.

Авторов книг часто критикуют за излишнюю многословность, а иногда за слишком скупое изложение материала. Сложно найти золотую середину, когда изучаемый материал будет одновременно интересен и полезен. Поэтому для сохранения баланса интереса и пользы я приложил немало усилий, чтобы донести до читателя именно ту частичку знаний, которая без труда отложится в долговременной памяти, а не забудется после первого прочтения.

Если у вас возникнут какие-либо вопросы или пожелания, буду рад общению с вами на моем сайте **volokh.info**.

Благодарности

В начале книг авторы любят кого-нибудь благодарить. Некоторые даже благодарят своих домашних питомцев. С одной стороны, это кажется забавным, а с другой, когда обдумываешь каждое предложение, такие литературные приемы зачастую помогают связать слова. В такие моменты понимаешь "цену" речевых оборотов. Поэтому не судите строго за "неосторожные слова".

В первую очередь конечно же хочется поблагодарить своих родителей за то, что они подарили мне жизнь. Если бы не они, то ничего бы не было: ни книги, ни тем более меня. Также хочется поблагодарить всех моих близких за то, что они с пониманием относились к тому, что я днями пропадал за компьютером, когда писал эту книгу. Самое ценное, что есть в этой жизни — это время: время, проведенное с близкими. Время — это то, чего нам всегда не хватает и не будет хватать. Надеюсь, что я смогу отблагодарить родных за те дни, которые я провел в виртуальной реальности, готовя материал для книги, и в дальнейшем буду больше времени уделять своим близким.

Отдельную благодарность хочу выразить вам, моим читателям, за то, что приобрели данную книгу, а не скачали ее из Интернета. Именно для вас она и создавалась.

Особую благодарность хочу выразить издательству "БХВ-Петербург" за то, что оно взялось за издательство книги, а ее сотрудники проделали огромную работу, чтобы эта книга вышла свет. Спасибо всем тем людям, которые были задействованы при подготовке к выпуску этой книги. Такие люди всегда остаются за кадром, но их работа крайне важна и зачастую остается неоцененной.

Всех и не перечислишь, поэтому спасибо всем тем, кто верил в меня и переживал за меня. Хочется также выразить благодарность моим школьным учителям, которые многое для меня сделали и передали свой жизненный опыт. Жизненный опыт хоть и передается устно, но зачастую приобретается, и в основном нелегким путем. И, наверное, жизненный опыт делает нас такими, какие мы сейчас есть.

Надеюсь, данная книга окажется для вас полезной. Приятного чтения.

Введение

Ubuntu Linux — это операционная система (ОС), основанная на базе ядра ОС Debian GNU, которая в свою очередь базируется на ядре Linux.

ПРИМЕЧАНИЕ

Очень часто к названию дистрибутива добавляется слово Linux, которое является неким указателем на принадлежность к семейству операционных систем. Этот "указатель" может быть как после названия дистрибутива (Ubuntu Linux), так и перед ним (Linux Mint).

Операционная система Ubuntu Linux является свободным программным обеспечением и распространяется бесплатно: любой пользователь, может свободно скачать данную операционную систему, установить ее и пользоваться ее абсолютно бесплатно.

На данный момент Ubuntu Linux является самым популярным дистрибутивом не только среди рядовых пользователей, но и среди IT-специалистов. Такой интерес с ОС объясняется тем, что она нацелена на простоту и удобство в использовании. Кроме того, Ubuntu Linux не уступает другим дистрибутивам в плане безопасности самой операционной системы.

Благодаря огромному числу разработчиков (сообществу) этой операционной системы Ubuntu Linux имеет поддержку большинства популярных языков мира, что обеспечивает ее доступность для разных языковых групп.

Популярность Ubuntu Linux обуславливается и тем, что ОС не требовательна к ресурсам компьютера: она будет прекрасно запускаться и работать даже на самом старом компьютере, на котором уже никогда не запустятся другие операционные системы.

Среди IT-специалистов данная операционная система популярна также из-за ее длительной поддержки, которая предполагает выпуск обновлений для ОС в течение 5 лет с момента выпуска версии LTS.

Эти и многие другие факторы являются несомненным достоинством данной операционной системы и весомым аргументом при выборе ее в качестве базовой ОС для вашего компьютера. Ubuntu Linux своей простотой и мощностью заслужила ваше вни-

мание и достойна изучения. Однако необходимо узнать историю появления самого ядра Linux, т. к. ядро Linux — это основа операционной системы Ubuntu Linux.

История появления ОС Linux

Linux (ли́нукс) — семейство свободно распространяемых операционных систем, основанных на базе ядра ОС UNIX.

Термин "семейство ОС" подразумевает наличие различных версий (дистрибутивов) этой операционной системы.

Первый релиз (выпуск) ядра состоялся 5 октября 1991 года, когда студент Хельсинского университета Линус Торвальдс начал работать над написанием собственной ОС, которая не имеет ограничений по сравнению с ОС Minix, написанной Эндрю Таненбаумом.

ОС Minix (ми́никс, от слова mini-UNIX) имела ряд ограничений на ее использование, например, применение Minix разрешалось только в образовательных целях. Это означало, что любое функционирование этой операционной системы в коммерческих целях было противозаконным. В связи с этим Линус Торвальдс начал разработку ядра Linux на базе ОС Minix, которая в свою очередь является UNIX-подобной операционной системой. Немного запутано, но такова история появления Linux.

Само название Linux происходит от объединения имени его основателя Линуса и названия ОС UNIX. Однако сам Линус Торвальдс изначально дал имя ядру Freax — гибриду английских слов *free* (бесплатный, свободный) и *freak* (чудной) с окончанием X. Хозяин FTP-сервера (Ари Лемке), на котором Линус выложил в общий доступ исходный код ядра, переименовал его в Linux. Впоследствии такое название закрепилось за этой ОС.

Работая над ядром Linux, Торвальдс хотел сделать его доступным для коммерческого использования, однако вначале своей работы над операционной системой он признавал тот факт, что его ОС непереносима на другие архитектуры компьютеров и, возможно, никогда не будет поддерживать их.

Об этом и о самом факте разработки операционной системы Линус сообщил 25 августа 1991 года в группе новостей **comp.os.minix**.

Письмо Линуса Торвальдса на сайте comp.os.minix

From: torvalds@klaava.Helsinki.Fi (Linus Benedict Torvalds)

To: Newsgroups: comp.os.inix

Subject: Чего вам больше всего не хватает в minix?

Summary: небольшой опрос для моей операционной системы Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.Fi>

Date: 25 августа 1991 г., 20:57:08 GMT

Organization: University of Helsinki

Привет всем пользователям minix!

Я пишу (бесплатную) операционную систему (это просто хобби, ничего большого и профессионального вроде gnu) для AT 386(486). Я вожусь с этим с апреля, и она, похоже, скоро будет готова. Напишите мне, кому что нравится/не нравится в minix, поскольку моя ОС на нее похожа (кроме всего прочего, у нее — по практическим соображениям — то же физическое размещение файловой системы).

Пока что я перенес в нее bash (1.08) и gcc (1.40), и все вроде работает. Значит, в ближайшие месяцы у меня получится уже что-то работающее, и мне бы хотелось знать, какие функции нужны большинству. Все заявки принимаются, но выполнение не гарантируется :-)

Линус (torvalds@kruuna.helsinki.fi)

PS. Она свободна от кода minix и включает мультизадачную файловую систему. Она НЕ переносима (используется переключение задач 386 и пр.) и, возможно, никогда не будет поддерживать ничего, кроме AT-винчестеров, потому что у меня больше ничего нет :-(

Выкладывая в общий доступ свою ОС, Линус и не предполагал, что его разработкой заинтересуется множество людей по всему миру.

К разработке Linux присоединилось огромное количество добровольных разработчиков, и постепенно эта система стала пригодной для практического использования.

GNU и Linux

Немаловажную роль в развитии Linux сыграл проект GNU, основанный Ричардом Столлманом.

Изначально ОС UNIX была несвободным программным обеспечением (ПО), разработанным в подразделении Bell Labs компании AT&T. В 1983 году Ричард Столлман объявил о начале работы над проектом свободной операционной системы UNIX, которую он назвал GNU Unix (Gnu's Not Unix).

Сообщение Ричарда Столлмана

Четверг, 27 сентября 1983 г., 12:35:59 EST

Свободный Unix!

После Дня Благодарения я начинаю писать Unix-совместимую программную систему GNU (Gnu's Not Unix), которую буду предоставлять свободно(!) всем, кто может ее использовать. Нужна помощь в виде времени, денег, программ и оборудования.

GNU будет содержать ядро плюс все утилиты, необходимые для того, чтобы писать и запускать программы на Си: редактор, оболочку, компилятор Си, линкер, ассемблер и еще несколько вещей. После этого будут добавлены программа форматирования текста, YACC, игра Empire, электронная таблица и сотни других вещей. Мы надеемся включить все, что обычно поставляется с Unix-системами, и все, что еще может оказаться полезным, в том числе онлайн-ую и печатную документацию.

GNU будет способна запускать программы Unix, но не будет идентична Unix. Мы будем вносить в систему улучшения, основываясь на нашем опыте работы с другими операционными системами...

Аббревиатура GNU расшифровывается как "GNU — это не UNIX" (GNU is Not UNIX). Чуть позже Ричард Столлман написал свой знаменитый Манифест GNU, который впоследствии стал основой лицензии GPL. С этого момента аббревиатура GNU стала расшифровываться как *General Public License* (Основная общественная лицензия).

С программным обеспечением под такой лицензией разрешается делать все что угодно: копировать, дорабатывать, продавать и т. п., правда, при условии того, что данное ПО также будет распространяться под лицензией GPL. Иными словами, ПО под лицензией GPL делает программное обеспечение свободным и гарантирует, что оно останется свободным и любой желающий может заглянуть в исходный код программы, находящейся под этой лицензией.

К 1990 году система GNU была почти закончена, и для нее было написано множество утилит, однако, как пишет Столлман, не хватало только ядра. И словно по удачному стечению обстоятельств внезапно появляется Линус Торвальдс и представляет миру свое ядро. А ведь это именно то, чего так не хватало Ричарду Столлману.

Факт, что Линус Торвальдс со своей разработкой оказался в нужное время в нужном месте, определил будущее ОС Linux.

Возможно, без ядра Linux проект GNU UNIX мог бы остаться невостребованным, не покажи Линус Торвальдс свою разработку миру разработчиков.

К слову сказать, правильной было бы называть GNU/Linux, а не просто Linux, и Ричард Столлман вправе поправлять всех нас, когда мы называем Linux Linux-ом, а не GNU/Linux. Ведь его заслуга в становлении ОС Linux весьма весома.

Появление различных дистрибутивов

После публикации исходных текстов ОС Linux стремительно начала набирать популярность, и ядро, написанное под платформу x86, было портировано (заточено) и под платформу x64.

Со временем, по мере развития ОС Linux, перед разработчиками встала задача получить компьютер со стабильно работающей операционной системой и с определенным набором утилит. Так начинают появляться первые дистрибутивы.

Дистрибутив (англ. *distribute* — распространять) — это форма распространения определенного программного обеспечения.

Каждый дистрибутив нацелен на определенный круг пользователей, от новичка до профессионалов.

Когда перед разработчиками встала задача получить компьютер с постоянно работающей ОС, то они начали создавать собственные наборы дискет с ядром и опреде-

ленным набором утилит. Само ядро и набор утилит можно было загрузить на жесткий диск компьютера и уже запускать ОС прямо с жесткого диска, а не с дискеты.

Именно такие наборы дискет стали прототипами современных дистрибутивов ОС Linux.

Тем не менее, несмотря на появившуюся возможность скопировать файлы ОС на жесткий диск компьютера, этого было недостаточно. Пользоваться таким дистрибутивом мог только очень профессиональный пользователь, т. к. для запуска требовалась самостоятельная компиляция исходных текстов программ в понятный для компьютера набор инструкций.

Так появился дистрибутив *Slackware*, созданный Патриком Фолькердингом в 1993 году. Данный дистрибутив является старейшим дистрибутивом из тех, которые сейчас активно развиваются.

На данный момент количество дистрибутивов Linux огромно. И это все благодаря открытости и свободе распространения исходных текстов под лицензией GNU GPL. Перечислять все не имеет смысла, но стоит кратко познакомиться с основными популярными дистрибутивами ОС Linux.

Дистрибутивы Linux

Прежде чем называть какие-то конкретные дистрибутивы, приведем классификацию наиболее популярных систем управления пакетами в дистрибутивах Linux (табл. B1).

Таблица B1. Классификация дистрибутивов по пакетным форматам

Пакетный формат	Примеры популярных дистрибутивов
DEB-based	Debian, Ubuntu, Mint, Knoppix
RPM-based	Red Hat, Fedora, SUSE, Novell
Pacman-based	Arch Linux, Chakra, Manjaro
Source-based	Slackware, Gentoo

В силу того что появлялось множество дистрибутивов, разработчики воплощали в них свои идеи и вносили улучшения. Так появилось деление на пакетные форматы, приведенные выше в таблице.

Каждый из форматов по-своему хорош и нацелен на решение тех или иных задач. Говорить о том, что тот или иной формат плох или лучше, чем другой, было бы неправильным. Нужно понимать, что каждый дистрибутив адресован той или иной аудитории, а это может быть как начинающий пользователь, так и программист.

Red Hat Enterprise Linux

Классическим дистрибутивом ОС Linux является дистрибутив **Red Hat Enterprise Linux**, выпускаемый компанией Red Hat (красная шляпа).

Red Hat Enterprise Linux распространяется по годовой подписке и нацелен на корпоративный сектор. Главной особенностью этого дистрибутива является наличие платной поддержки на протяжении 10 лет. К примеру, для того чтобы получить бинарные (уже скомпилированные) пакеты обновления для операционной системы, нужно заплатить некоторую денежную сумму.

Если ранее, до 2002 года, Red Hat Linux был дистрибутивом общего назначения, то теперь (с мая 2002 года) Red Hat Enterprise Linux актуален только для коммерческих организаций. Устанавливать его на домашний компьютер не имеет смысла, если даже не глупо.

Fedora Linux

Fedora (Федора) — дистрибутив ОС Linux, выпускаемый компанией Red Hat.

Продукт является неким подобием "полигона" для тестирования новых технологий, которые планируется включить в дистрибутив Red Hat Enterprise Linux.

Дистрибутив Fedora распространяется бесплатно, и его поддержка осуществляется только благодаря сообществу в Интернете. Fedora будет интересна тем, кто любит устанавливать и тестировать новое программное обеспечение. Разумеется, что в качестве рабочего сервера такой дистрибутив не годится.

Debian

Debian (Дэбиан) — некоммерческий дистрибутив ОС Linux, предназначенный для работы в качестве как серверной, так и настольной операционной системы.

Большую популярность дистрибутив завоевал у профессионалов в области безопасности информационных систем благодаря быстрому реагированию на найденные ошибки и угрозы.

К программному обеспечению, поставляемому с дистрибутивом Debian, предъявляются жесткие требования, что в свою очередь гарантирует стабильность выпускаемых версий дистрибутива.

Дистрибутив Debian стал ответвлением целой линейки Debian-подобных дистрибутивов: Ubuntu, Knoppix, Astra-Linux и т. д.

Debian рекомендуется в первую очередь специалистам по безопасности, а также опытным пользователям.

Ubuntu

Ubuntu (Убунту) — операционная система, основанная на проекте Debian. Главным спонсором и разработчиком является компания Canonical.

Ubuntu — очень популярная ОС не только среди рядовых пользователей, но также среди специалистов, которые устанавливают ее в качестве серверной операционной системы.

Мнение о том, что Ubuntu не годится на роль серверной ОС, ошибочно, и это доказано не один раз. При правильной и грамотной установке Ubuntu можно получить стабильный сервер, который выдержит любые нагрузки и атаки. Поэтому использовать Ubuntu в качестве серверной ОС не только можно, но и нужно.

Помимо всего выше названного, Ubuntu неплохо справляется с обязанностями домашней ОС благодаря своему дружественному пользовательскому интерфейсу.

Именно об этой операционной системе пойдет речь в данной книге.

Linux Mint

Linux Mint (англ. *mint* — мята) — дистрибутив, основанный на ОС Ubuntu.

Linux Mint практически ничем не отличается от ОС Ubuntu, кроме включенного в состав дистрибутива списка пакетов.

Основная философия разработки Linux Mint — простота и удобство, которые предполагают готовность работы ОС сразу же после ее установки. Это означает, что пользователю не потребуется дополнительно скачивать наиболее популярное программное обеспечение.

Все это, несомненно, является большим достоинством для начинающих пользователей, чтобы склониться в пользу выбора миграции на Linux с ОС Windows.

Slackware

Slackware Linux — один из самых старейших дистрибутивов на ядре Linux, выпущенный Патриком Фолькердингом 16 июля 1993 года.

Slackware по праву называют самым "юниксовым", однако и самым сложным среди всех современных дистрибутивов Linux.

Установка дистрибутива Slackware Linux несколько сложна ввиду того, что весь процесс установки выполняется в текстовом режиме. По этой причине без определенного багажа знаний этот дистрибутив многим может оказаться не под силу, но тем, кто хочет погрузиться в более детальное изучение мира Linux, этот дистрибутив будет как нельзя кстати.

Логотип Linux

Официальным логотипом и символом ОС Linux является пингвин **Tux** (произносится как Такс). Слово Tux придумал Джеймс Хьюз, соединив два слова: "(T)orvalds (U)ni(X)".

Почему в качестве талисмана был выбран именно пингвин, Линус Торвальдс объясняет тем, что ему очень нравятся пингвины. Примечательно, что жена Линуса

Торвальдса, Туве Торвальдс, в книге "Just for Fun" рассказывает о том, что однажды в австралийском зоопарке Линуса клюнул пингвин.

Область применения Linux

Спектр применения ОС Linux довольно обширен, начиная с домашних станций и заканчивая системами военного назначения.

Ни для кого не секрет, что Linux используется в качестве ОС для серверов и прекрасно справляется с возложенной задачей. Нередки случаи, когда хорошо настроенные серверы на базе ОС Linux не требовали перезагрузки в течение многих лет. Windows-пользователю это может показаться фантастикой, но факт остается фактом. Linux не нуждается в перезагрузке, как того часто требует ОС Windows после незначительной конфигурации системы. В ОС Linux порою достаточно перезагрузить демон (в ОС Windows — служба), и операционная система примет новую конфигурацию без перезагрузки самой ОС. Весьма замечательно, не правда ли?

ОС Linux также нашла свое применение и у профессионалов. В своей деятельности ее часто используют программисты при разработке Web-сайтов.

Почему именно Linux, а не Windows? ОС Linux очень хорошо выполняет функции сервера и не требовательна к ресурсам, в отличие от Windows. На деле причин в пользу выбора Linux более чем одна, и можно долго их перечислять.

Наверняка у многих из вас дома имеется устройство, называемое маршрутизатором. Никогда не задумывались о том, что за ОС управляет работой этого устройства? Нет? Тогда для вас станет новостью, что многие маршрутизаторы управляются ОС Linux. Разумеется, не все маршрутизаторы, потому что выбор ОС для подобных устройств зависит от предпочтений самого производителя, но зачастую выбор делается в пользу свободных операционных систем (GNU/Linux и UNIX-подобных ОС). И это не случайно, потому что их можно модернизировать, не нарушая лицензии. И все это абсолютно без каких-либо лицензионных отчислений.

Практически в каждом магазине или супермаркете стоят банкоматы и различные платежные терминалы. Так вот на некоторых из них также установлена ОС Linux, не на всех, но тем не менее такое устройство, как банкомат или терминал, способно управляться ОС Linux.

Да что там говорить про банкоматы, в военной и космической промышленности используется операционная система Linux, специально "заточенная" на выполнение специфических задач.

Каждый день мы используем устройства на базе ОС Linux и не замечаем этого. Простой пример — ваш смартфон. Он имеет в своем составе ОС Android (не в обиду будет сказано приверженцам устройств другого компьютерного гиганта), которая основывается на ядре ОС Linux.

ОС Linux постепенно входит в нашу повседневную жизнь, и мы этого не замечаем. Наверно потому, что производители устройств под управлением ОС Linux часто забывают упомянуть о факте, что это устройство работает-то на базе Linux. Если

бы при загрузке или выключении устройства появлялся логотип пингвина Тух, то пользователи бы задумались: "Не Linux ли это?"

Идеология свободного ПО

Движение за свободное программное обеспечение возникло благодаря Ричарду Столлману, который в 1983 году сформулировал идею о необходимости использования свободного ПО (СПО).

Собственно идея СПО базируется на предоставлении полного права (свободы) на использование программного обеспечения под любой свободной лицензией. Это означает, что, владея копией такого ПО, вы имеете неограниченные права на установку, запуск, свободное использование, распространение, а также на любые изменения и усовершенствования данного ПО. Такую программу вы имеете право даже продавать или раздавать ее совершенно бесплатно.

Таким образом, программа является свободной, если соблюдается 4 принципа:

- ◆ свобода использования программы, любым способом и в любых целях (свобода 0);
- ◆ свобода изменять программу под свои цели и нужды (свобода 1);
- ◆ свобода распространения копий программ, бесплатно или за деньги (свобода 2);
- ◆ свобода распространять измененные версии программ (свобода 3).

Провозгласив эти 4 принципа, Ричард Столлман понял, что этого будет недостаточно, и необходимо какое-то документальное оформление данной концепции. Таким образом, появилась лицензия GNU General Public License (Основная общественная лицензия GNU), сокращенно GNU GPL или просто GPL.

В этой лицензии Столлман оговорил одно условие: ни один пользователь, внесший в программу какие-либо изменения, не имеет право ее распространять, если он не соблюдает всех принципов свободного ПО.

Дело в том, что до появления лицензии GNU GPL Ричард Столлман занимался разработкой текстового редактора Emacs, который базировался на исходном тексте Джеймса Гослинга. В один прекрасный момент для Гослинга и далеко не прекрасный для Столлмана Джеймс Гослинг продал право распространения редактора Emacs компании UniPress. Компания UniPress, в свою очередь, потребовала от Ричарда Столлмана прекратить распространение его версии Emacs.

Чтобы подчеркнуть отличие лицензии GNU GPL от других лицензий (copyright), которые ограничивают свободу, ввели новый термин — **copyleft** (копилефт).

Copyleft предполагает сохранение свободы распространения программы, даже если в программе был использован небольшой фрагмент исходного кода, находящийся под лицензией GNU GPL. Таким образом, программа, использующая код GNU GPL, должна быть выпущена под этой же лицензией.

Программа, использующая в своем составе какую-то часть свободного кода, становится свободной программой.

Стоит иметь в виду, что понятия "свободный" и "бесплатный" — совершенно разные вещи. Бесплатный (freeware) не имеет отношения к свободному программному обеспечению.

Безопасен ли открытый исходный код?

В Интернете довольно много споров на тему: безопасен ли открытый исходный код или нет?

Мнения по этому вопросу разделились. Одни считают, что открытый исходный код безопасен, а другие утверждают о вреде открытых исходных текстов, потому что в таком коде легче найти ошибки и использовать их во вред. К единому мнению эти два лагеря так и не пришли и, возможно, не придут.

Вначале, когда проекты с открытым исходным кодом разрабатывались программистами-добровольцами, такой исходный код не считался надежным. В нем часто присутствовали порой очевидные ошибки и уязвимости, т. к. поначалу был важен результат, а о качестве кода заботились в последнюю очередь.

Сейчас же ситуация изменилась, и над открытыми проектами работают программисты, труд которых оплачивается, и эта деятельность является основной, а не выступает в качестве хобби.

Те, кто придерживается мнения о вреде открытости исходных текстов, считают, что в программе с закрытым исходным реже находят ошибки и уязвимости, чем с открытым. Однако это утверждение ошибочно. Да, в закрытых программах труднее найти уязвимость, потому что их исходный текст закрыт. Придется искать ошибки "вслепую", а этот процесс отнимает много времени. Но даже в таких программах находят ошибки и порой не раз. Исходные тексты знать совсем не обязательно, ведь при поиске уязвимости взломщик посылает программе ошибочные и неожиданные для программы данные и наблюдает за тем, как она отреагирует на это. Если программист допустил ошибку, то программа в лучшем случае зависнет и перестанет отвечать. В худшем случае программа в ответ на действия взломщика выдаст те данные, которые могут нанести вред организации или предприятию.

Вспомним хотя бы яркий пример закрытой программы Adobe Flash Player. За всю историю существования этой программы в ней было найдено столько уязвимостей, что сразу и не вспомнишь их количество. Ошибки и уязвимости всплывали даже сразу после очередного обновления и оставались незакрытыми продолжительное время.

Приверженцы пользы открытых исходников придерживаются мнения, что открытость исходных текстов программ позволяет быстрее находить все возможные ошибки и также оперативно их устранять. И это мнение имеет право на жизнь.

Ради справедливости вспомним, сколько раз в программе Sendmail (кроссплатформенный почтовый сервер) находились уязвимости и как эти уязвимости оперативно устранялись открытым сообществом. Администраторы тех систем, которые подвергались взлому через найденные в Sendmail уязвимости, признаются, что знание исходных текстов помогло им самостоятельно закрывать бреши в системе.

Критика и очевидные достоинства ОС Linux

Так же как и споры о пользе и недостатках открытых исходных текстов, не угасают споры о преимуществах и недостатках ОС Linux.

Начнем с недостатков, а затем рассмотрим преимущества ОС Linux перед другими системами.

Итак, сложность в освоении ОС Linux останавливает многих начинающих пользователей. Действительно, работа с терминалом (аналог командной строки в ОС Windows) отбивает желание начинающих пользователей изучать ОС Linux. Разумеется, что в некоторых дистрибутивах прибегать к работе с терминалом придется лишь в редких случаях, но все же придется.

Вторым недостатком является отсутствие, точнее, малое количество игр, написанных под эту операционную систему. Несомненно, большая часть игр разработана только для ОС Windows, и неизвестно, когда разработчики игр массово начнут создавать продукты для Linux-геймеров. Да, игры для Linux есть, но их очень мало.

Несомненным минусом является низкий спрос на компьютеры с предустановленной ОС Linux. И по этой причине поставщики не желают терпеть убытки, и устанавливают ту ОС, компьютеры с которой будут быстрее раскупаться.

Хочется отметить также и тот факт, что к начинающим "линуксоидам" в русскоязычном сегменте Интернета относятся немного недружелюбно. Когда новичок обращается за помощью в решении какого-либо вопроса, то зачастую на него обрушивается такой шквал негатива, что отбивает у него всяческое желание не что-то спрашивать, но и изучать эту ОС.

В одной из книг по ОС Linux было примерно такое высказывание: "Современное поколение пользователей не желает пользоваться терминалом в ОС Linux, они хотят только кликать мышкой".

На первый взгляд, усложнять себе жизнь глупо. Если ОС предоставляет инструмент, который будет выполнять нужные задачи быстрее, чем простой ввод команд в терминале, то почему бы не воспользоваться этим инструментом? И почему использование таких программных инструментов является неправильным, и за это нужно недолго любить современное поколение пользователей?

Однако стоит согласиться с тем, что профессионал *должен* владеть навыками работы в Терминале, а простому пользователю это совсем необязательно, и не нужно его за это ненавидеть или обрушивать на него свой гнев.

Наверно, из-за такого негативного отношения к начинающим "линуксоидам" со стороны русскоязычного сообщества ОС Linux еще не скоро появится на домашних компьютерах наших соотечественников.

Да, есть вполне адекватные советы новичкам от более опытных "линуксоидов", но таких крайне мало. Просто каждый из нас забывает, с чего он сам начинал.

Итак, назовем главные достоинства ОС Linux.

Первым плюсом будет *свободность* и *открытость* ОС Linux. Именно два этих качества позволили ОС Linux стать такой, какой мы ее видим сейчас.

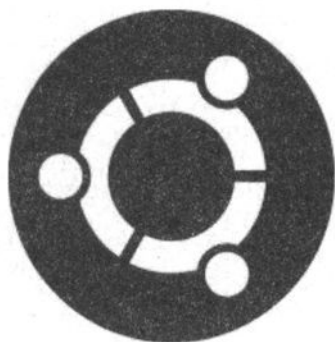
Вторым плюсом стало то, что ОС Linux является *безопасной* операционной системой. Не зря именно ОС Linux устанавливают в качестве серверной операционной системы.

После безопасности следует *быстродействие*, которое присуще не всем операционным системам. Согласитесь, никому из нас не понравится работать на слишком медленной системе, а ОС Linux шустро работает даже на слабом компьютерном железе.

Если первые версии ОС Linux предлагали сложный процесс установки, то уже современные дистрибутивы имеют довольно простую установку в графическом режиме. ОС сама все сделает за вас, а вам только останется ответить на несколько простых вопросов, и — вуаля! — у вас на компьютере имеется работающая ОС Linux.

И наверно, главным достоинством будет наличие большого списка поддерживаемого оборудования. Для установки какого-либо драйвера порой достаточно выбрать драйвер, предлагаемый системой по умолчанию, и ваша звуковая карта или видеокарта заработает должным образом.

О преимуществах можно говорить довольно долго, но не будем заходить слишком далеко и поднимать столько горячую тему для обсуждения, а лучше перейдем непосредственно к изучению ОС Linux.



ЧАСТЬ I

Знакомьтесь: Ubuntu

Глава 1.	Подготовка к установке ОС Ubuntu Linux
Глава 2.	Теоретический минимум о файловой системе в ОС Ubuntu Linux
Глава 3.	Создание разделов для установки ОС Ubuntu Linux
Глава 4.	Установка ОС Ubuntu Linux
Глава 5.	Первый запуск и первоначальная настройка ОС Ubuntu
Глава 6.	Установка обновлений и локализация операционной системы
Глава 7.	Обзор интерфейса ОС Ubuntu
Глава 8.	Настройка операционной системы и решение проблем

ГЛАВА 1



Подготовка к установке ОС Ubuntu Linux

ОС Ubuntu Linux

Ubuntu Linux — это операционная система, основанная на дистрибутиве Debian Linux и спонсируемая компанией Canonical Ltd.

Название "Ubuntu" пришло к нам из южно-африканских языков банту и означает человечность и взаимопомощь друг другу.

Новые версии ОС Ubuntu выходят каждые 6 месяцев. Первым публичным выпуском ОС Ubuntu был релиз Ubuntu 4.10 Preview. По заверению разработчиков ОС Ubuntu всегда будет доступна бесплатно.

Требования для ОС Ubuntu минимальны:

- ◆ процессор с тактовой частотой 700 МГц;
- ◆ 1024 Мбайт оперативной памяти;
- ◆ VGA-совместимая графическая карта, поддерживающая разрешение 1024×768 точек;
- ◆ 10 Гбайт свободного дискового пространства.

Эти требования касаются только самой операционной системы, но не стоит забывать, что помимо ОС на вашем компьютере будут работать программы различной сложности, которые задействуют часть ресурсов компьютера.

Версии дистрибутивов ОС Ubuntu

На данный момент для загрузки существуют две основные версии дистрибутивов ОС Ubuntu, которые представлены в 32- и 64-битном исполнении:

- ◆ Ubuntu Desktop;
- ◆ Ubuntu Server.

В книге мы будем использовать 64-битную версию ОС, т. к. 32-битные системы уходят в прошлое и разработчики постепенно отказываются от их поддержки.

ОС Ubuntu Desktop предназначена для домашнего пользования, а ОС Ubuntu Server для серверного применения и поставляется без графического интерфейса. Для изучения ОС Ubuntu Linux нам нужна "десктопная" версия Ubuntu, т. е. Ubuntu Desktop.

Существуют также версии ОС Ubuntu с долгосрочной поддержкой, например Ubuntu 16.04 LTS. Поговорим о них более подробно.

Версии LTS и не-LTS

В чем отличие LTS-версии, от не-LTS?

Аббревиатура LTS расшифровывается как Long Term Support — релиз с долгосрочной поддержкой. Действительно, LTS-версия имеет долгосрочную поддержку, т. е. на протяжении 5 лет вы будете получать все нужные обновления для вашей ОС — разработчики Ubuntu будут выпускать их, а вы можете быть уверены, что эти обновления никоим образом не затронут стабильность работы вашей версии операционной системы.

Обычные версии ОС Ubuntu, т. е. не LTS-версии, содержат все новшества, которые могут быть включены в последующие релизы (выпуски) LTS-версий. Логично предположить, что такие версии предназначены в основном для тех, кто интересуется новинками и кому интересно экспериментировать с системой.

У дистрибутивов Ubuntu нумерация версий состоит из пары чисел, разделенных точкой, которые означают дату выпуска и месяц. Так, 10.04 означает, что версия дистрибутива вышла в апреле 2010 года, а версия 16.04 — в апреле 2016 года.

Версия LTS подходит не только для изучения дистрибутива Ubuntu Linux, но и для дальнейшего его использования в качестве основной рабочей системы, т. к. LTS-версии отличаются своей стабильностью и долгосрочной поддержкой. Именно LTS-версию мы и будем рассматривать в книге.

ПРИМЕЧАНИЕ

На момент написания этих строк была выпущена версия Ubuntu 17.04. Использовать ее в качестве настольной операционной системы не рекомендуется, т. к. эта версия ОС является нестабильной и предназначена в основном для тестирования новшеств.

Варианты установки ОС Ubuntu Linux

Существует три основных варианта установки операционной системы Ubuntu Linux на компьютер:

- ◆ Ubuntu как единственная ОС на компьютере;
- ◆ Ubuntu как дополнительная ОС на компьютере;
- ◆ ОС Ubuntu на виртуальной машине.

Первый вариант предполагает установку ОС Ubuntu Linux в качестве основной и единственной операционной системы на компьютере. В этом случае ОС Ubuntu

устанавливается на чистый жесткий диск и размечает весь его объем для монопольного использования.

При выборе второго варианта установки ОС Ubuntu устанавливается в качестве дополнительной операционной системы. Допустим, на компьютере уже имеется ОС Windows, которая используется в качестве основной, а ОС Ubuntu ставится в качестве дополнительной системы. В этом случае пространство жесткого диска компьютера размечается для использования двух операционных систем и уже используется в соответствии с выделенным для каждой из них объемом.

И наконец, третий вариант предполагает установку ОС Ubuntu на виртуальной машине, например, Oracle VM VirtualBox. Установка производится в так называемом "контейнере", т. е. под "присмотром" программы Oracle VM VirtualBox. На жестком диске компьютера создается специальный файл, который имитирует жесткий диск, и установка операционной системы практически ничем не отличается от обычной установки ОС.

Как правило, на компьютере пользователя уже установлена одна из версий ОС Windows, поэтому мы рассмотрим установку ОС Ubuntu Linux в качестве дополнительной системы к Windows 7. Данный вариант является наиболее сложным и наиболее интересным из всех вариантов установки, т. к. требует внимательной и правильной разметки жесткого диска с целью не допустить потери уже имеющейся информации на нем.

Порядок установки операционных систем

При установке операционных систем Ubuntu и Windows на жесткий диск следует соблюдать правильную очередность этого процесса.

ОС Windows агрессивно относится к операционным системам, уже установленным на компьютере. Так, если сначала установить ОС Ubuntu в качестве основной системы, а затем попытаться "добавить" ОС Windows, то программа установки ОС Windows "затрет" загрузчик Ubuntu Linux, и ОС Ubuntu перестанет запускаться. Избежать этого позволит обратный процесс — сначала необходимо установить ОС Windows, а уже затем ОС Ubuntu.

ОС Ubuntu дружелюбно относится к операционным системам, "живущим по соседству", и позволяет правильно сконфигурировать порядок загрузки операционных систем без нарушения их нормальной работы.

И еще один совет: будьте внимательны при установке ОС на компьютер, на жестком диске которого уже имеются важные данные.

Как получить дистрибутив ОС Ubuntu?

Получить дистрибутив с ОС довольно легко. Для этого перейдите на русскоязычный сайт <http://ubuntu.ru/> и вверху страницы выберите ссылку **Скачать Ubuntu**. Вашему вниманию будут представлены дистрибутивы двух версий: 32- и 64-бит-

ные. Ранее мы уже определились с разрядностью ОС и остановили свой выбор на 64-битной версии. Ее можно скачать по прямой ссылке или через любой torrent-клиент (рис. 1.1).

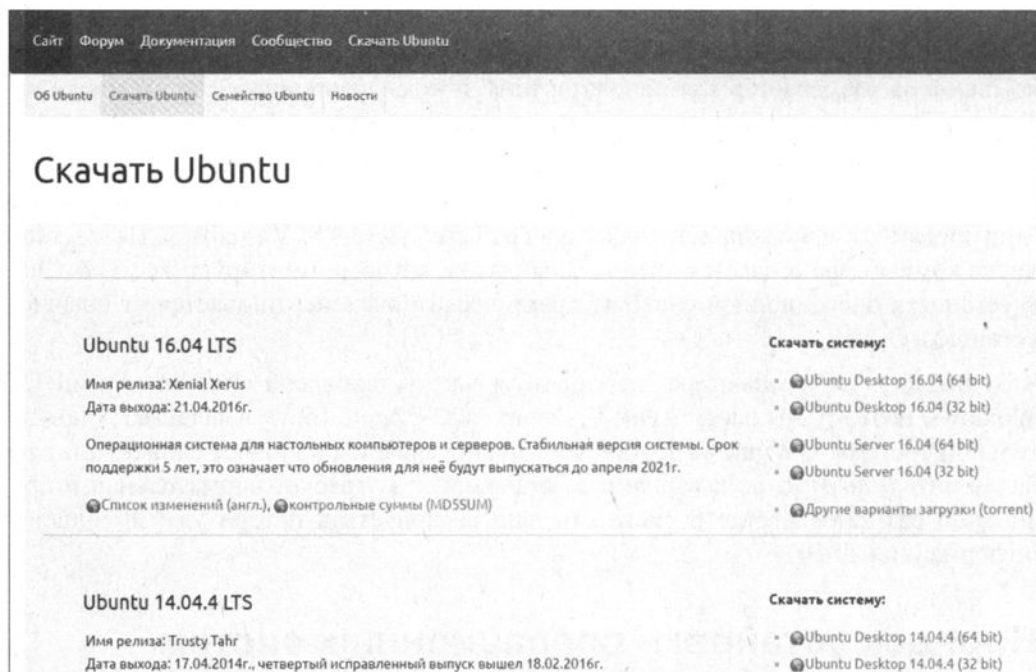


Рис. 1.1. Варианты загрузки ОС Ubuntu

На этой же странице вы можете проверить MD5-суммы образов дистрибутивов. Это необязательно, но бывает полезным в тех случаях, когда вы хотите убедиться, что загруженный дистрибутив является именно тем, который предоставили разработчики ОС Ubuntu, и он не подвергался изменению со стороны взломщиков. Нередки случаи, когда злоумышленникам удавалось подменить оригинальный образ дистрибутива на модифицированный, содержащий вредоносный код. MD5-суммы не являются панацеей, потому что если злоумышленник смог получить полный контроль над сайтом, то ему не составит труда изменить и значение MD5-суммы на сайте. Однако это удастся не всегда, и MD5-суммы бывают полезны.

Для прямой загрузки ISO-образа операционной системы щелкните по ссылке **Ubuntu Desktop 16.04 (64 bit)**, и файл образа дистрибутива будет загружен посредством вашего браузера.

ПРИМЕЧАНИЕ

Предпочтительнее загружать дистрибутив с помощью torrent-клиента. Для этого надо щелкнуть по ссылке **Другие варианты загрузки (torrent)** и уже затем выбрать нужную версию для загрузки. Затем следует выбрать вариант **ubuntu-16.04-desktop-amd64.iso.torrent** и, наконец, открыть загруженный torrent-файл в своем torrent-клиенте.

Напомним, что установка ОС Ubuntu будет производиться на компьютер с уже установленной операционной системой Windows.

Нам остается только дождаться окончания загрузки дистрибутива Ubuntu (рис. 1.2).



Рис. 1.2. Загрузка ОС Ubuntu Linux с помощью torrent-клиента

Когда ISO-образ дистрибутива будет загружен, запишем его на DVD-диск и приступим к установке.

Запись образа диска с ОС Ubuntu

Файл дистрибутива имеет расширение `iso`. Такой файл является своего рода архивом и предназначен для создания образа загрузочного диска.

Для того чтобы создать загрузочный диск, необходимо записать загруженный образ дистрибутива на DVD-диск.

ПРИМЕЧАНИЕ

Для создания загрузочного образа диска не следует записывать ISO-образ целиком на диск. Для корректного процесса создания загрузочного диска нужно воспользоваться соответствующей опцией в вашей программе для записи дисков.

Вкратце поясним суть процесса создания загрузочного DVD-диска.

При создании загрузочного диска посредством таких программ, как Nero, InfraRecorder и т. п., в соответствующем меню программы следует выбрать пункт меню записи образа на DVD-диск. В InfraRecorder — **Действия | Записать образ**, а в Nero — **Образ, проект, копир. | Образ диска или сохр. проект**. Затем нужно выбрать ISO-образ, и программа для записи дисков сама все сделает за нас.

СОВЕТ

Не стоит пытаться разархивировать ISO-образ и разархивированные файлы записывать на диск. Ничего из этого не получится, и вы только испортите DVD-диск. Пользуйтесь специализированными программами.

Если на вашем компьютере отсутствует DVD-привод, на помощь приходит установка операционной системы с USB-накопителя. Для этого необходимо создать загрузочный флеш-накопитель (или, попросту говоря, флешку) из ISO-образа.

Создание загрузочного флеш-накопителя

Для создания загрузочной флешки существует множество программ. Мы воспользуемся программой UNetbootin, которая позволяет создать загрузочные флешки для различных Linux-дистрибутивов.

Загрузить программу можно с сайта <http://unetbootin.github.io>. Существуют версии как для Linux-дистрибутивов, так и для ОС Windows. Выберите версию для ОС Windows, т. к. именно эта операционная система на данный момент установлена на нашем компьютере.

Для создания загрузочной флешки нам понадобится USB-накопитель объемом не менее 8 Гбайт. Флешку нужно предварительно отформатировать под файловую систему FAT32, если она была у вас с другой файловой системой. После этого запустите программу UNetbootin и установите переключатель **Образ диска** (рис. 1.3).

Нажмите на кнопку с тремя точками и выберите дистрибутив с ОС Ubuntu.

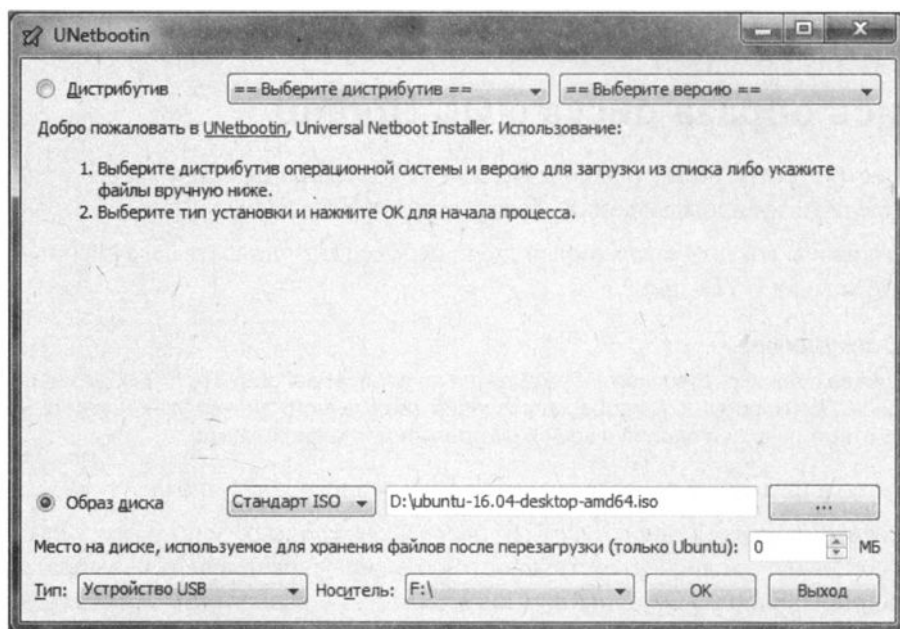


Рис. 1.3. Программа UNetbootin

Программа UNetbootin автоматически определит подключенный USB-накопитель. Если этого не произошло, вручную укажите тип носителя и сам носитель.

Для начала процесса создания загрузочной флешки нажмите кнопку **ОК**. Начнется процесс распаковки файлов на USB-накопитель (рис. 1.4).

Процесс подготовки загрузочной флешки занимает около 5 минут. По окончании нажмите кнопку **Выход** (рис. 1.5).

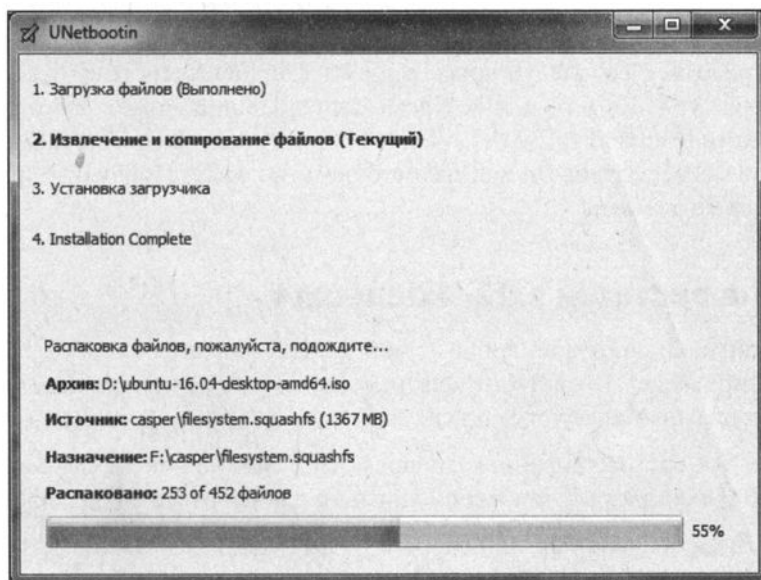


Рис. 1.4. Начат процесс копирования файлов на USB-накопитель

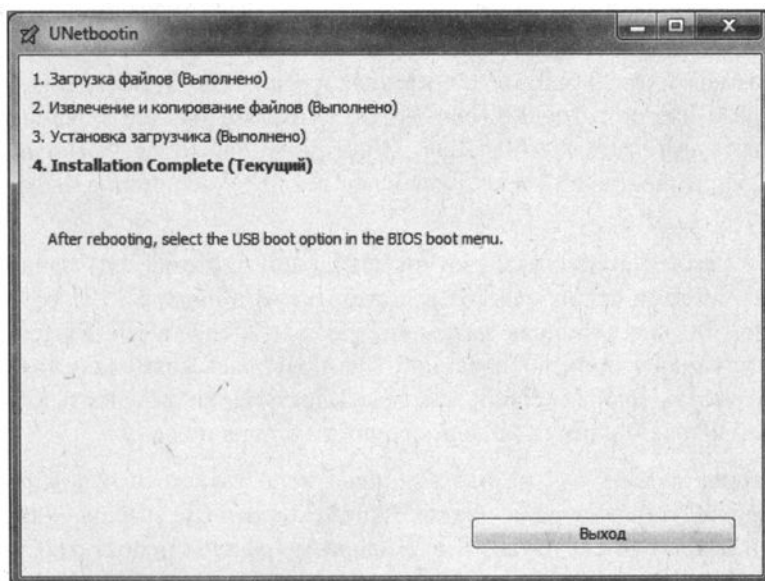


Рис. 1.5. Программа UNetbootin сообщила о завершении процесса создания загрузочной флешки

Файловые системы ОС Ubuntu Linux

Прежде чем приступить непосредственно к процессу установки ОС Ubuntu на компьютер, следует познакомиться с файловыми системами (ФС), которые используются в ОС на базе Linux.

Задача выбора файловой системы встает в начале установки ОС Ubuntu. Конечно, сменить ФС можно и после установки операционной системы, но логичнее и правильнее все это сделать в процессе установки.

ОС Ubuntu работает со следующими типами файловых систем: ext2, ext4, ext4, Btrfs, XFS, ReiserFS, Reiser4 и JFS. Кроме этого, Ubuntu может работать с файловыми системами FAT16, FAT32, VFAT и NTFS, которые предназначены для работы в ОС Windows. Рассматривать мы их не будем, но кратко обсудим "родные" для Ubuntu файловые системы.

Файловые системы ext2, ext3 и ext4

Основной файловой системой для ОС Ubuntu Linux является ext, которая постепенно модернизировалась и претерпевала изменения, но даже прежние версии этой файловой системы используются до сих пор.

Extended File System (расширенная файловая система) — поколение файловых систем ext2, ext3 и ext4, разработанных специально для работы с ОС на основе Linux.

Изначально была создана первая версия ФС, получившая название ext (extfs). Выпущена она была в апреле 1992 года. Целью ее создания было стремление преодолеть ограничения файловой системы Minix File System.

В разработанной первой версии файловой системы были сняты некоторые ограничения, заложенные в ФС Minix File System. Так, появилась возможность работы с файлами размером до 2 Гбайт, в то время как в Minix File System было ограничение размера раздела жесткого диска в 64 Мбайт. Также в ФС ext появилась возможность создавать файлы, максимальная длина названия которых могла составлять теперь уже 255 символов, в то время как в Minix File System длина была ограничена 30 символами.

Со временем Реми Кардом (разработчик файловой системы ext) начал понимать, что файловая система ext нуждается в доработке, и в январе 1993 года появилась вторая версия ФС, получившая название Second Extended File System, т. е. ext2. В ней была улучшена производительность, и появилась возможность присваивать различные атрибуты файлам, такие как права доступа, возможность устанавливать владельца файла, возможность задавать группу доступа и т. д.

Однако и вторая версия ФС не была лишена недостатков, и в ноябре 2001 года в свет вышла следующая версия — ext3 (Third extended file system — третья версия расширенной файловой системы). В ext3 были устранены недостатки предыдущей версии, такие как отсутствие поддержки раздельного доступа к файлам, отсутствие временных меток модификации файлов. Главным отличием новой версии ФС стала

возможность журналирования файловой системы. Это функция позволила восстанавливать работу ОС в случае сбоя в работе компьютера.

Помимо журналирования, появилась возможность работы с файлами до 1 Тбайт и размером блочных данных до 16 Тбайт.

ФС ext3 долгое время была стандартом файловой системы в ОС на базе Linux до выхода четвертой версии расширенной файловой системы — ext4 (Fourth extended file system), первая экспериментальная версия которой была выпущена в октябре 2006 года и через два года стала стабильной файловой системой.

В ext4 существенно увеличен размер файлов по сравнению с предшествующей ФС. Так, ограничение размера файла стало равным 16 тебибайтам, а максимальный размер объема жесткого диска — 1 эксбибайт. Также был существенно улучшен механизм фрагментации файлов.

Файловая система ext4 предлагается по умолчанию в ОС Ubuntu, начиная с версии 9.10.

Именно с этой файловой системой мы и будем работать. Остальные файловые системы используются не так часто и в основном для специфических задач. По этой причине мы опишем их очень кратко.

Файловая система Btrfs

Btrfs (B-tree FS, Better FS или Butter FS) — файловая система для ОС на базе Linux, разработанная компанией Oracle Corporation в 2007 году для более эффективного использования дискового пространства.

Она будет полезна тем, чья работа по большей части связана с файлами малого размера, и тем, кому необходимо максимально задействовать ресурсы жесткого диска.

Файловая система XFS

XFS — журналируемая файловая система, разработанная компанией Silicon Graphics в 1994 году. Целью создания XFS стала возможность работы с файлами большого объема, более 2 Тбайт.

Система отличается высокой производительностью за счет выполнения некоторых операций на лету, но и не лишена недостатков, например, невозможно изменить размер имеющейся файловой системы, а также существует большой риск потери данных в случае сбоя питания компьютера.

XFS пользуется популярностью у тех, кто занимается видеомонтажом на компьютерах с большой вычислительной мощностью.

Файловые системы ReiserFS и Reiser4

Файловая система ReiserFS, созданная компанией Namesys, предназначена для работы с файлами небольшого размера. Следующим поколением (четвертым) этой файловой системы в 2004 году стала файловая система Reiser4.

ReiserFS позволяет снизить фрагментацию файлов за счет упаковки небольших файлов в один блок.

Reiser4 была призвана устранить недостатки своего предшественника.

Сейчас разработка ReiserFS приостановлена, но ею продолжают пользоваться те, кому часто приходится работать с маленькими файлами.

Файловая система JFS

JFS (Journaled File System) — журналируемая файловая система, разработанная компанией IBM в 1990 году.

JFS предназначена для обеспечения высокой производительности на высоконагруженных серверах, при этом она позволяет вести журнал изменений метаданных, обеспечивая целостность всей структуры файловой системы. В случае сбоя питания сохраняются устаревшие копии файлов, что позволяет восстановить работу, но при этом будут восстановлены прежние версии файлов, которые были до сбоя питания.

На этом закончим краткий обзор типов файловых систем и перейдем к установке ОС Ubuntu Linux.

Выводы

Мы познакомились с версиями дистрибутивов ОС Ubuntu, выбрали нужную нам 64-битную версию и скачали ее образ. Сделали загрузочный дистрибутив для последующей установки ОС Ubuntu 16.04 LTS по соседству с ОС Windows.

Мы узнали о файловых системах, которые используются в ОС Ubuntu, и выбрали предлагаемую по умолчанию файловую систему ext4.

ГЛАВА 2



Теоретический минимум о файловой системе в ОС Ubuntu Linux

Разделы жесткого диска

Раздел жесткого диска — это часть долговременной памяти, выделенной для удобства работы с информацией, находящейся на компьютере.

Так, один раздел жесткого диска обычно выделяется для хранения файлов самой ОС, а другой для хранения личной информации.

Разделы мы можем увидеть в любой операционной системе. Ярким примером является наличие дисков C: и D: в ОС Windows (ОС Ubuntu: /, /home, соответственно) — рис. 2.1.

Эти разделы по своей сути являются логическими, т. к. получены логическим разбиением (разделением) пространства одного физического накопителя.

Преимуществ использования разделов множество. Перечислим только основные:

- ♦ возможность установить несколько операционных систем на одном физическом диске;
- ♦ возможность разделять информацию по типу — файлы ОС, личные данные и т. п.;
- ♦ возможность использования разных файловых систем на различных разделах.

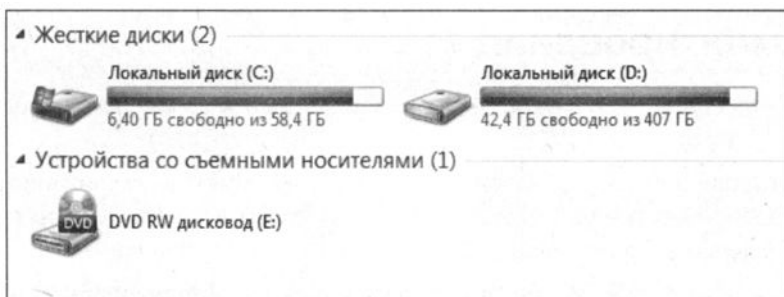


Рис. 2.1. Разделы жесткого диска в ОС Windows

Логических разделов может быть несколько. Их количество необязательно ограничивается двумя.

Первичные, расширенные и логические разделы

Вся информация о размещении разделов на жестком диске хранится в так называемой *таблице разделов* (partition table), которая является частью *главной загрузочной записи* (master boot record, MBR), содержащей код и данные, необходимые для загрузки ОС.

Разделы могут быть *первичными* либо *расширенными*.

В первом секторе первичного раздела находится *загрузочный сектор*, который отвечает за загрузку ОС с этого раздела. В этом разделе и находится главная загрузочная запись MBR. Первичный раздел должен обязательно присутствовать на жестком диске.

MBR содержит информацию о том, какой именно раздел должен быть использован при загрузке ОС. В MBR под таблицу разделов выделено всего 64 байта, каждая запись которой (для одного раздела) занимает 16 байт.

Таким образом, всего на жестком диске может быть создано не более четырех разделов. На момент разработки структуры MBR этого было достаточно.

Когда первичных разделов стало недостаточно, был введен *расширенный раздел* (EBR, Extended Boot Record), в котором можно создавать несколько логических разделов.

По правилам, расширенный раздел должен быть только один. По этой причине, на жестком диске можно создать лишь три первичных раздела и один расширенный.

Идея использования MBR постепенно отходит на второй план, и производители компьютеров уже переходят на технологию EFI (Extensible Firmware Interface — расширяемый микропрограммный интерфейс) или UEFI (Unified EFI) вместо привычной BIOS, т. к. там вместо MBR используется GPT (GUID Partition Table). В связи с этим снимаются ограничения на количество разделов.

Однако понимать принцип разбиения пространства жесткого диска все же необходимо, т. к. в книге мы будем рассматривать именно этот вариант деления разделов.

Точки монтирования

Точка монтирования — это каталог, присоединенный к дереву каталогов корневой файловой системы.

Точка монтирования предназначена для присоединения (монтирования) разделов диска к файловой системе в ОС Linux. Для того чтобы легче было понять, рассмотрим это определение на примере файловой системы ОС Windows.

В ОС Windows корневым каталогом является диск C:. Для хранения личных документов в большинстве случаев создается раздел диска D:. В ОС Linux это будет / и

/home соответственно. Таким образом, на практике получается, что в ОС Windows корневой каталог на каждом из разделов диска будет свой, в то время как в ОС Linux корневой каталог только один — /. В ОС Linux совсем неважно, сколько имеется разделов диска, корневой каталог будет всегда один, а точек монтирования может быть несколько.

Допустим, что на компьютере с установленной ОС Windows имеются три диска: C:, D: и E:. Первый раздел содержит файлы операционной системы и установленные драйверы. На втором разделе хранятся документы пользователя. Диск E: — это оптический привод. Теперь в качестве наглядности отобразим это в сравнительной таблице (табл. 2.1).

Таблица 2.1. Сравнительная таблица разделов с ОС Windows и точек монтирования в ОС Linux

Наименование диска в ОС Windows	Точка монтирования в ОС Linux
Диск C:	/
Диск D:	/home
Диск E:	/media

Таким образом, понятно, что ОС Linux не создает новый диск, а просто монтирует его к корневой файловой системе, которая отображается как прямой слеш / и является *основной точкой монтирования*.

Точки монтирования в ОС Ubuntu

Наверняка вы спросите: какие точки монтирования можно создать при установке ОС Ubuntu? Все зависит от целей, которые вы преследуете при установке операционной системы.

Для настольной ОС достаточно создать три раздела (точки монтирования):

- ♦ / — корневой каталог файловой системы;
- ♦ /swap — файл подкачки;
- ♦ /home — раздел для хранения личных документов.

Этих трех разделов часто бывает достаточно для вполне работоспособной домашней операционной системы на базе Linux. Однако ОС Linux позволяет также создать дополнительные разделы (точки монтирования).

- ♦ /boot — содержит начальный загрузчик и заголовки ядра ОС. Является аналогом зарезервированного диска в операционной системе Windows 7, на котором содержатся основные файлы загрузчика ОС Windows, необходимые для восстановления запуска системы и т. п.
- ♦ /dev — содержит файлы драйверов и устройств. В ОС Windows нет строгого деления на каталоги драйверов, поэтому драйверы устройств могут находиться в разных каталогах.

- ◆ /opt — содержит дополнительно программное обеспечение. Аналогом в ОС Windows является каталог C:\Program Files.
- ◆ /srv — каталог, содержащий файлы системных сервисов. В ОС Windows нет строго определенного каталога, из которого должны запускаться системные сервисы.
- ◆ /tmp — каталог для хранения временных файлов. В ОС Windows аналогом является каталог C:\Temp.
- ◆ /usr — каталог установленных пользователем приложений. В ОС Windows для этих целей служит каталог C:\Program Files.
- ◆ /usr/local — каталог данных, доступный всем пользователям. В ОС Windows в качестве примера можно привести каталог C:\Users\Public.
- ◆ /var — каталог изменяемых файлов, таких как очереди печати, запущенные программы, сервисы и т. п. В ОС Windows нет строго определенного каталога для этих целей.

Так выглядит список точек монтирования, которые можно создать на стадии установки ОС на основе Linux. Они позволяют гибко настроить ОС и свести к минимуму риск потери данных в случае сбоя, а также облегчают процесс сопровождения операционной системы.

Мы не будем создавать все эти точки монтирования, т. к. в нашем случае ОС Ubuntu будет устанавливаться в качестве дополнительной системы. Достаточно создать корневой каталог / и файл подкачки /swap.

Каталог /home создавать необязательно, т. к. мы сможем воспользоваться монтированием диска D:, на котором хранятся наши документы. Однако если вы планируете постепенно перейти на ОС Ubuntu и сделать ее основной операционной системой, то можно создать каталог /home. В любом случае, это можно сделать и после установки ОС Ubuntu.

Очередность создания разделов

Изначально, когда компьютеры были не такие быстрые и мощные как сейчас, каждый программист «бился» за быстродействие системы и экономил процессорное время. Со временем эта необходимость отпала, и программы стали создавать без прежней экономии памяти и процессорного времени.

С тех пор при установке операционной системы появилось негласное правило создания разделов строго по своей очередности. Вначале создавался раздел /swap таким образом, чтобы он находился ближе к шпинделю жесткого диска, затем создавался корневой раздел / и уже потом все остальные разделы, такие как /home и т. п.

Поскольку мы будем устанавливать ОС Ubuntu в качестве дополнительной системы, то данное правило очередности соблюсти не удастся, т. к. ОС Windows уже заняла требуемое место на жестком диске. Однако расстраиваться совсем не обяза-

тельно — на современном компьютере мы практически не заметим разницу в скорости обращения к жесткому диску.

Объем раздела /swap

В Интернете можно встретить различные доводы в пользу выбора того или иного размера раздела /swap. Некоторые советуют выбирать размер в зависимости от объема оперативной памяти, установленной на компьютере. Так, если на вашем компьютере установлены 2 Гбайт оперативной памяти, то размер файла подкачки, т. е. раздела /swap, должен быть не менее 2 Гбайт. На машинах, где оперативная память больше или равна 4 Гбайт, рекомендуют создавать файл подкачки, равный половине объема оперативной памяти, либо вовсе его не создавать за ненадобностью. Универсального ответа нет и, вероятно, не будет. Однако стоит помнить, что в любом случае, каким бы не был файл подкачки, оперативная память работает намного быстрее всяческих файлов подкачки.

По своему опыту автор рекомендует создавать файл подкачки равный чуть больше объема оперативной памяти: если размера файла подкачки будет не хватать, его легко можно изменить в любой удобный момент времени.

Объем корневого раздела

Только что установленная ОС Ubuntu занимает примерно 4–6 Гбайт в зависимости от выбранных опций установки.

Для нормальной работы ОС требуется не менее 10–15 Гбайт свободного пространства, которое понадобится для установки различных программ. Если вы планируете использовать сложные программы, которые требуют значительный объем дискового пространства, то нужно исходить из этих критериев.

На первое время, 60 Гбайт дискового пространства будет вполне достаточно.

Объем раздела /home

Обычно разделу /home отдают все оставшееся пространство жесткого диска. Поскольку мы будем устанавливать ОС Ubuntu в качестве дополнительной операционной системы, то этот раздел создавать не обязательно. Все личные файлы и документы можно хранить на томе с файловой системой NTFS, т. е. на диске D:, который можно будет монтировать при работе с ОС Ubuntu.

ПРИМЕЧАНИЕ

Запомните, что ОС Ubuntu создаст каталог /home в корневом каталоге, и /home будет принадлежать корневому каталогу /.

Подготовительные действия перед установкой

Финальным этапом подготовки к установке ОС Ubuntu станет *дефрагментация диска*, пространство которого мы хотим задействовать для установки операционной системы.

Так как на нашем компьютере уже имеется установленная ОС Windows, то для установки ОС Ubuntu нам потребуется дополнительно свободное дисковое пространство. Где его взять? Свободное дисковое пространство на диске C: или D: можно задействовать для нужд ОС Ubuntu, но перед этим нам нужно произвести дефрагментацию того раздела диска, свободное место которого мы будем "отнимать" для дополнительной ОС, т. е. для ОС Ubuntu Linux.

Дефрагментация диска нужна для того, чтобы максимально близко друг к другу расположить фрагментированные файлы на жестком диске. Для этого в ОС Windows нажмите кнопку **Пуск**, а затем последовательно выберите **Все программы | Стандартные | Служебные** и запустите программу Дефрагментация диска.

В окне программы Дефрагментация диска выделите нужный диск, например D:, и нажмите кнопку **Дефрагментация диска**. Будет произведена дефрагментация диска, и на этом подготовительные действия будут окончены.

Выводы

Мы познакомились с основами файловой системы ОС Ubuntu и узнали, что представляют собой точки монтирования в этой операционной системе.

Мы познакомились с разделами, которые бывают первичными, расширенными и логическими. Мы узнали, что первичных разделов может быть три, а расширенный раздел только один. Эти знания нам понадобятся при создании разделов в процессе установки ОС Ubuntu.

ГЛАВА 3



Создание разделов для установки ОС Ubuntu Linux

Выбор типа носителя для загрузки операционной системы в меню BIOS

Нам предстоит настроить порядок загрузки компьютера в BIOS. Если вы уже имели дело с переустановкой ОС, то уже, возможно, сталкивались с изменением порядка загрузки в BIOS. Значит, этот материал вам будет знаком.

Для начала нам необходимо попасть в BIOS. Для этого при загрузке компьютера, т. е. до того момента, как начнется загрузка ОС, нужно успеть нажать соответствующую клавишу запуска BIOS. Обычно это клавиша , но иногда и <F2>, <F10>, <F12> и др. Все зависит от производителя материнской платы вашего компьютера. Если вы сомневаетесь, какая клавиша отвечает за меню загрузки BIOS, то нужно обратиться к документации вашего компьютера и выяснить это.

Итак, вы знаете, какую именно клавишу нажимать (в качестве примера это будет клавиша). Теперь при загрузке компьютера во время появления надписи "Press Del to enter Setup" вы должны успеть нажать клавишу , после чего вы попадете в основное меню BIOS.

ПРИМЕЧАНИЕ

Дать какие-то четкие инструкции по настройке загрузки в BIOS не удастся, т. к. данные настройки будут различаться в зависимости от модели и производителя материнской платы.

После того, как вы открыли меню BIOS, т. е. нажали соответствующую клавишу запуска, вам нужно в одном из меню найти пункт, который будет называться примерно так: **Boot device priority**. В этом меню BIOS вам нужно настроить приоритет загрузки с DVD-привода или USB-накопителя (в зависимости от выбранного вами носителя).

Назначьте первым в списке именно тот носитель, с которого вы будете загружать ОС Ubuntu для установки ее на компьютер. Для сохранения проделанных изменений нажмите клавишу <F10> и введите букву Y для выхода из меню BIOS с сохранением внесенных изменений.

В некоторых моделях материнских плат компьютер при запуске позволяет выбрать тип носителя, с которого должна быть произведена загрузка. В этом случае появляется надпись примерно следующего содержания: "Press F12 to select boot device". От вас требуется только нажать клавишу <F12> и выбрать тип носителя, с которого вы хотите загрузить компьютер.

Если же ваш компьютер не имеет такого меню выбора загрузочных устройств при включении, то вам все же придется менять приоритет загрузки именно в меню BIOS.

ПРИМЕЧАНИЕ

Перемещаться по меню BIOS можно с помощью клавиш-стрелок, а выбирать нужные пункты меню следует с помощью клавиши <Enter>. Для того чтобы вернуться в предыдущее меню, используется клавиша <Esc>.

Особенности установки на платах с UEFI

Как нам уже известно, на новых компьютерах на смену BIOS пришел UEFI.

UEFI (Unified Extensible Firmware Interface — унифицированный расширяемый интерфейс прошивки) — это специальный программный интерфейс между операционной системой и программами, которые управляют низкоуровневыми функциями оборудования.

Данный стандарт пришел на смену EFI, являясь его прямым потомком.

В поздних версиях меню BIOS есть режим поддержки UEFI, который называется **Legacy** либо **UEFI Disabled**.

MBR позволяет создавать не более трех первичных разделов и один расширенный при условии, что материнская плата не имеет поддержки UEFI. Платы, имеющие поддержку UEFI, позволяют работать с новой таблицей разделов GPT, которая обеспечивает создание уже не четырех, а множества разделов на диске. При этом они все будут основными, т. е. первичными. Замечательно, не так ли?

Таким образом, когда включен режим UEFI, используется GPT вместо устаревшей MBR. Эти особенности необходимо знать, если на вашем компьютере имеется поддержка UEFI. В остальном загрузка и установка ОС на компьютере с UEFI практически ничем не отличается от установки на компьютеры с поддержкой BIOS.

Варианты запуска и установки ОС Ubuntu

До момента появления CD/DVD-дисков основным способом загрузки являлись *дискеты*. Теперь, когда компьютеры стали мощнее, а операционные системы сложнее, стали возможными загрузка и установка операционной системы с оптического диска CD либо DVD. И вот не так давно стал доступен третий способ: загрузка с USB-носителя (загрузочной флешки).

Последние два способа используются по сей день, поэтому рассмотрим каждый из них в отдельности.

Запуск ОС Ubuntu с загрузочного флеш-накопителя

После задания приоритета загрузки с USB-накопителя подключите вашу загрузочную флешку к компьютеру и перезагрузите его. При загрузке появится вот такое окно загрузчика UNetbootin (рис. 3.1).

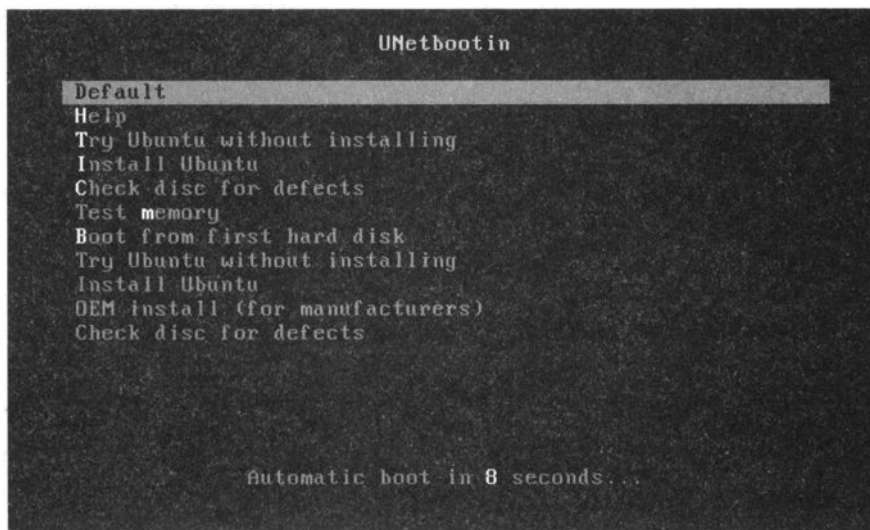


Рис. 3.1. Окно выбора меню загрузки в UNetbootin

По умолчанию выбран первый вариант **Default**, который будет запущен через несколько секунд. На данном этапе никаких дополнительных действий от нас не требуется, и нам нужно дождаться загрузки ОС Ubuntu в Live-режиме (рис. 3.2).

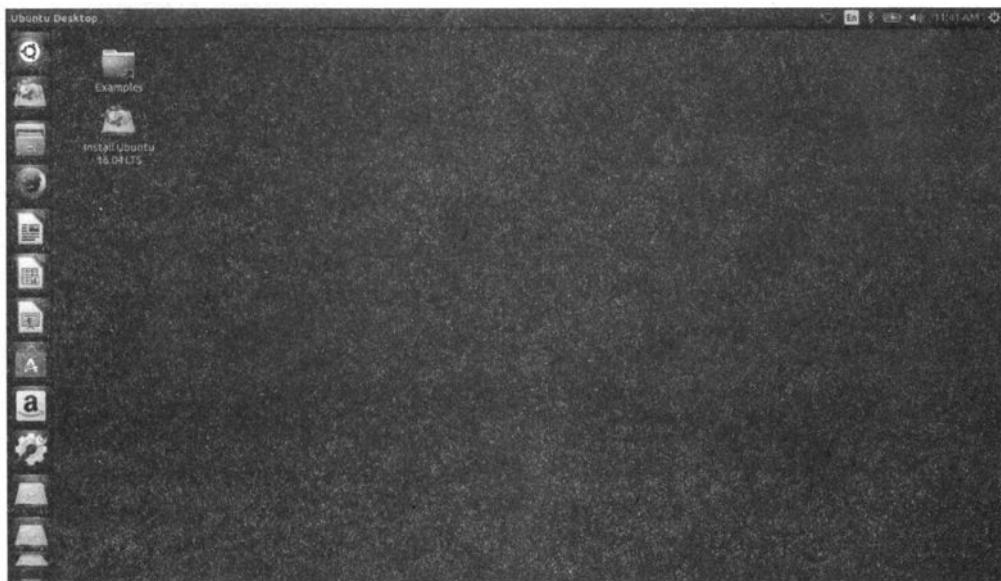


Рис. 3.2. ОС Ubuntu загрузилась в Live-режиме

Теперь ОС Ubuntu загружена в Live-режиме, который позволит нам сделать разметку диска для установки операционной системы. Но об этом чуть позже, а сейчас рассмотрим вариант загрузки с DVD-привода.

ПРИМЕЧАНИЕ

Live-режим предназначен для запуска ОС Ubuntu Linux без ее непосредственной установки.

Запуск ОС Ubuntu с DVD-привода

Вставьте в DVD-привод диск с записанным дистрибутивом ОС Ubuntu и перезагрузите компьютер.

Через несколько минут компьютер откроет приветственное окно установщика ОС Ubuntu (рис. 3.3).

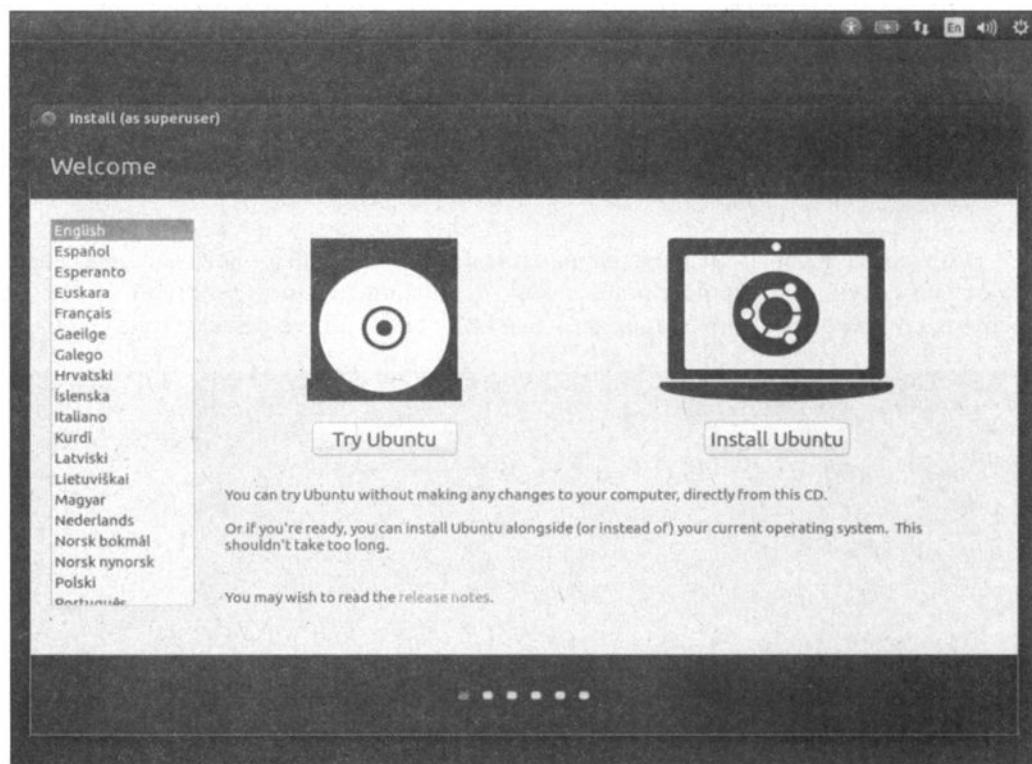


Рис. 3.3. Приветственное окно установщика Ubuntu

Так как нам нужно разметить жесткий диск компьютера под установку Ubuntu, то нужно запустить Live-режим. Для этого нажмите кнопку **Try Ubuntu**. Запустится Live-режим, который позволит нам воспользоваться операционной системой без ее установки на компьютер.

Для чего нужен Live-режим?

Наверняка у вас появился вопрос: зачем запускать ОС Ubuntu Linux без ее установки? Не лучше ли сразу ее установить на компьютер?

Запуск ОС в Live-режиме понадобится для предварительной разметки пространства жесткого диска для последующей установки ОС Ubuntu. Ведь не зря мы делали дефрагментацию раздела, который был отформатирован в файловой системе NTFS. Именно этот раздел, точнее свободное пространство этого раздела, мы отведем под нужды ОС Ubuntu Linux. И лучше всего в этом случае нам поможет именно Live-режим, в котором мы будем редактировать созданные разделы жесткого диска.

Запуск программы GParted

После запуска ОС Ubuntu в Live-режиме нам нужно освободить свободное место жесткого диска для установки Ubuntu Linux. Выделять свободное пространство мы будем с раздела D:, который был создан ОС Windows.

Запомните, а лучше запишите объем каждого раздела на вашем компьютере. Это позволит вам не запутаться в созданных разделах и правильно сделать разметку жесткого диска.

После того как вы запомнили или записали размеры разделов диска, в Live-режиме нажмите кнопку с логотипом ОС Ubuntu, которая называется *главным меню* (или Dash), и в строке поиска введите слово `gparted` (рис. 3.4).

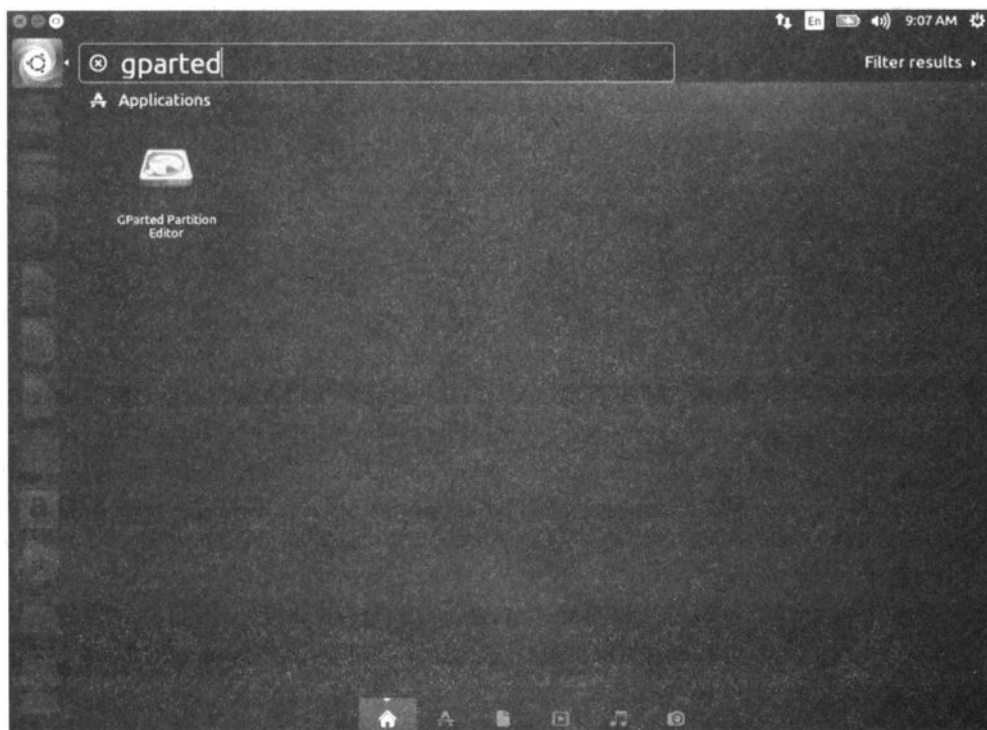


Рис. 3.4. Поиск программы GParted Partition Editor

В ответ на это главное меню мгновенно отобразит в качестве результата программу GParted Partition Editor (далее — просто GParted). Запустите ее. Она поможет нам разметить пространство жесткого диска.

Разметка жесткого диска с двумя разделами C: и D: в общем виде выглядит так, как показано на рис. 3.5.

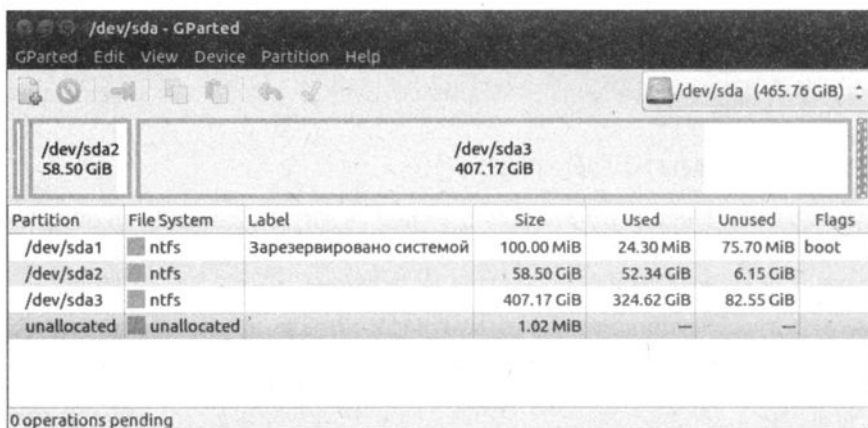


Рис. 3.5. Разметка жесткого диска в программе GParted

В текущей конфигурации жесткого диска с установленной ОС Windows мы можем наблюдать три первичных раздела:

- ◆ /dev/sda1 — раздел, автоматически создаваемый при установке ОС Windows 7 для восстановления работы ОС в случае сбоя. На этом разделе хранятся файлы загрузчика ОС Windows;
- ◆ /dev/sda2 — раздел для хранения файлов операционной системы и установленных программ, т. е. диск C:;
- ◆ /dev/sda3 — раздел для хранения личных файлов и документов, т. е. диск D:.

Об именовании разделов мы поговорим чуть позже, а теперь вкратце познакомимся с функциональными особенностями программы GParted.

Возможности программы GParted

Программа GParted Partition Editor позволяет выполнять любые операции с жестким диском: создавать и изменять разделы, задавать метки, форматировать раздел под ту или иную файловую систему и т. п.

Мы будем работать с такими возможностями программы, как:

- ◆ изменение размера раздела (команда контекстного меню **Resize/Move**);
- ◆ создание нового раздела (команда контекстного меню **New**);
- ◆ сохранение изменений (кнопка **Apply All Operations**).

На данном этапе изучения ОС Ubuntu этого будет вполне достаточно, и мы можем перейти к разметке жесткого диска.

Разметка жесткого диска в программе GParted

Настало время сделать первый серьезный шаг, который отделял нас от самого процесса установки ОС Ubuntu, — выполнить разметку жесткого диска.

ПРИМЕЧАНИЕ

При всех манипуляциях с разметкой жесткого диска стоит быть очень внимательным и не торопиться. Нередки случаи, когда второпях уничтожались целые разделы с личной информацией, и при этом не имелось никаких копий на других носителях. По этой причине рекомендуется сделать резервную копию всей важной для вас информации и только после этого приступать к процессу разметки жесткого диска в программе GParted.

Так как установить ОС Ubuntu прямо в раздел, созданный ОС Windows, невозможно, следовательно, нам придется выделить место на разделе диска D:. В программе GParted этот раздел обозначается как `/dev/sda3`. Щелкните правой кнопкой мыши по этому разделу и выберите пункт меню **Resize/Move** (рис. 3.6).

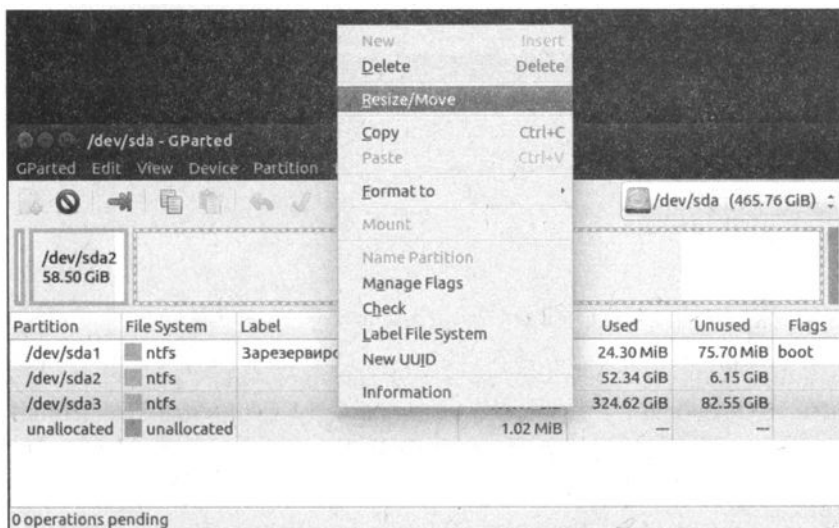


Рис. 3.6. Изменение размера раздела в программе GParted

Давайте выделим 30 Гбайт. Для этого в поле счетчика **Free space following (MiB)** введите значение 30000, т. е. размер в мегабайтах, и после этого нажмите кнопку **Resize/Move** (рис. 3.7).

Этим действием мы выделяем свободное пространство с диска D:, которое нам понадобится для установки ОС Ubuntu. Получится примерно такая картина (рис. 3.8).

После этих манипуляций у нас появился неразмеченный раздел, который значится как **unallocated**. Именно его мы и будем сейчас разметать.

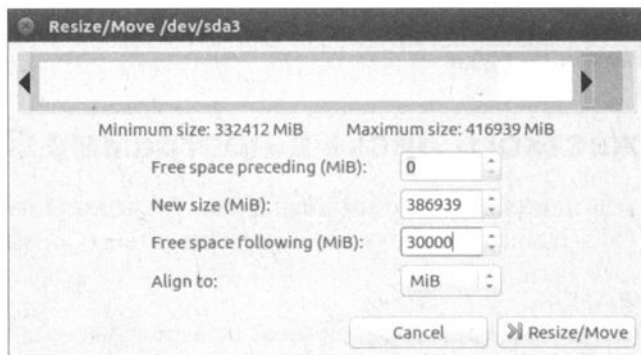


Рис. 3.7. Выделение размера для установки ОС Ubuntu

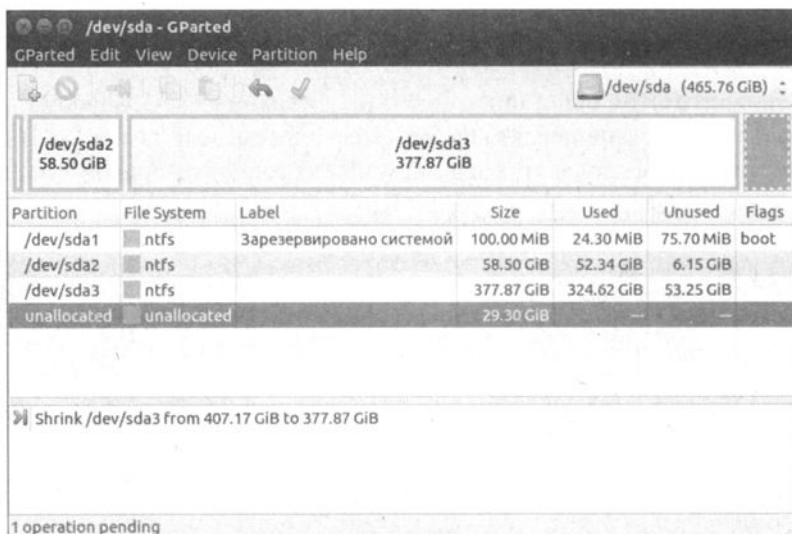


Рис. 3.8. Программа GParted выделила 30 Гбайт с диска D:

Обратите внимание, что у нас уже имеются три первичных раздела, а как мы уже знаем, MBR не позволяет создавать больше четырех первичных разделов. Если не-размеченный раздел пометить как *первичный*, то нам больше не удастся создать дополнительных разделов, и при попытке создания будет выведено сообщение об ошибке (рис. 3.9).

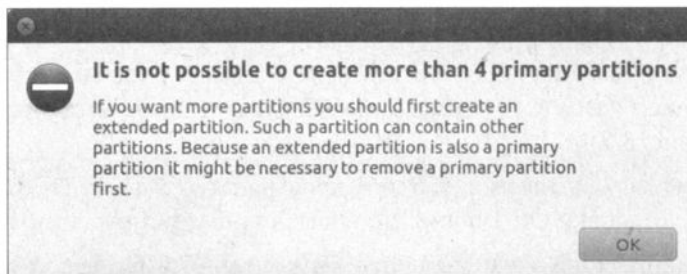


Рис. 3.9. Ошибка создания свыше четырех первичных разделов

Да, мы можем создать четвертый первичный раздел и пометить его как корневую файловую систему, но не лучше ли создать еще два раздела: файл подкачки и раздел для хранения личных документов?

Вот и пришло время вспомнить, что существует такое понятие, как *расширенный раздел*. Вспомнили? Отлично, потому что расширенный раздел позволит создать несколько логических разделов, которые нам так необходимы при установке ОС Ubuntu.

Для того чтобы создать расширенный раздел, щелкните правой кнопкой мыши по неразмеченной области диска, т. е. по **unallocated**, и выберите пункт меню **New**. Откроется окно **Create new Partition** (рис. 3.10).

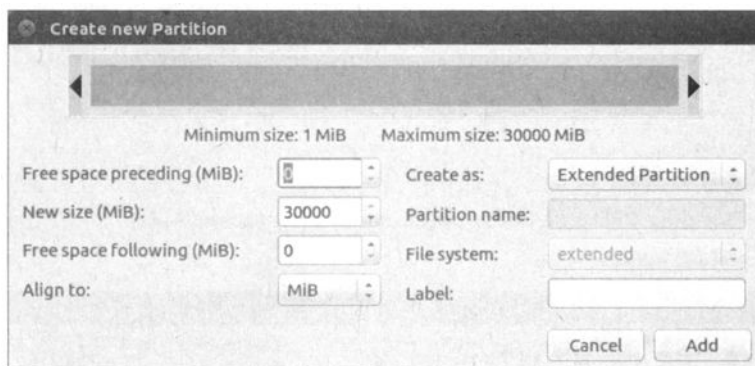


Рис. 3.10. Создание расширенного раздела

В раскрывающемся списке **Create as** выберите вариант **Extended Partition** и нажмите кнопку **Add** для сохранения изменений. Это позволит нам создать расширенный раздел, в котором мы уже будем создавать корневую файловую систему, файл подкачки и раздел для хранения документов.

Если вы все сделали правильно, то у вас получится примерно так, как изображено на рис. 3.11.

Обратите внимание, что в нижней части окна программы GParted появился список действий, которые будут применены после нажатия кнопки **Apply All Operations**. Так, первым заданием станет изменение размера раздела с 407,17 Гбайт на 377,87 Гбайт. Второй очередью задания создается расширенный раздел размером в 29,3 Гбайт. Все просто и наглядно. Этим и отличается программа GParted от программ подобного типа.

Теперь создадим *файл подкачки*, который еще называют *своп-файлом* (swap). Так как предпочтительно создавать файл подкачки по объему равным чуть больше размера самой оперативной памяти, то выберем значение равным в 4500 Мбайт.

С размером файла подкачки мы определились. Перейдем к его созданию. Щелкните правой кнопкой мыши по неразмеченному разделу и выберите пункт меню **New** (рис. 3.12).

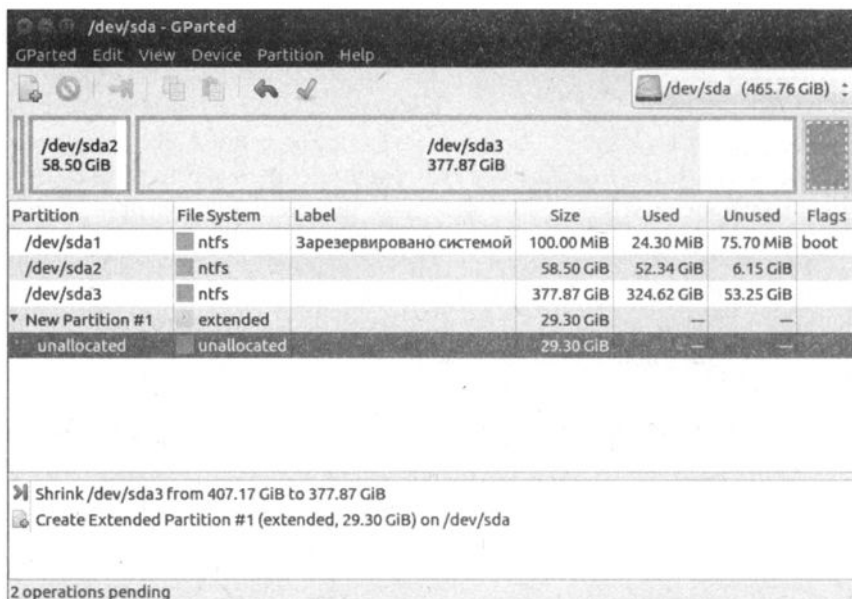


Рис. 3.11. Новый раздел помечен как расширенный

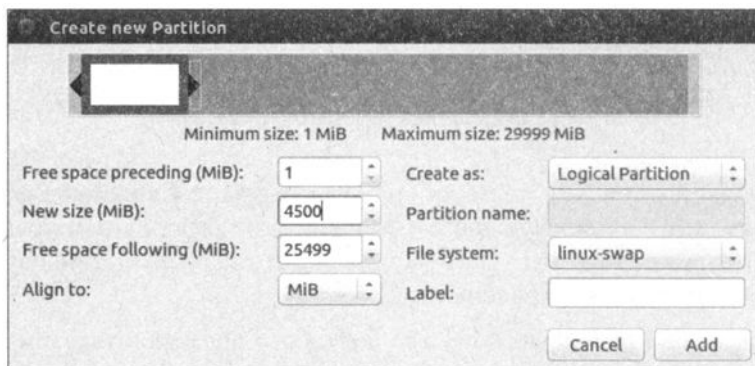


Рис. 3.12. Создание файла подкачки

В поле счетчика **New size (MiB)** установите размер файла подкачки. В нашем случае это 4500 Мбайт. Затем в раскрывающемся списке **File system** выберите значение **linux-swap**. Это позволит пометить созданный раздел как файл подкачки. Для подтверждения создания раздела нажмите кнопку **Add**.

Следующий шаг — создание корневой файловой системы, т. е. раздела для точки монтирования **/**. Для этого щелкните правой кнопкой мыши по неразмеченному разделу и выберите пункт меню **New**. Откроется окно создания нового раздела (рис. 3.13).

Давайте для корневой файловой системы выделим 15 Гбайт. Для этого в поле счетчика **New size (MiB)** установите значение равным 15000. Теперь нам нужно задать тип файловой системы для создаваемого раздела. Как мы уже знаем, лучше всего

для наших задач подходит файловая система **ext4**, ее мы и выберем в раскрывающемся списке **File system**. Для подтверждения создания раздела нажмите кнопку **Add**.

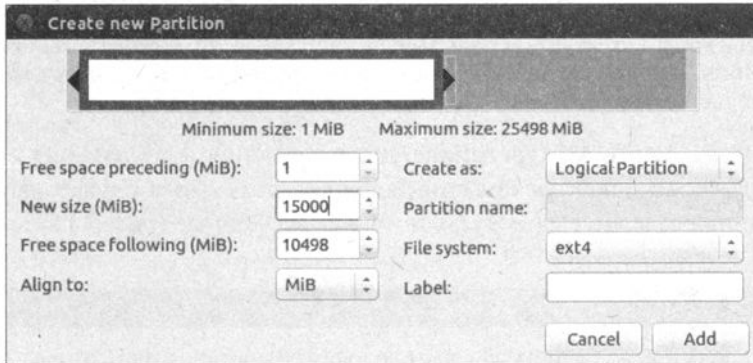


Рис. 3.13. Создание корневого раздела файловой системы

Завершающим и в то же время необязательным этапом в разбивке жесткого диска на разделы будет создание раздела для хранения личных файлов и документов. Точка монтирования этого раздела будет **/home**.

Для раздела **/home** можно отвести все оставшееся неразмеченное пространство жесткого диска. Как это сделать, мы уже знаем.

В конечном итоге у нас получится примерно такая конфигурация (рис. 3.14).

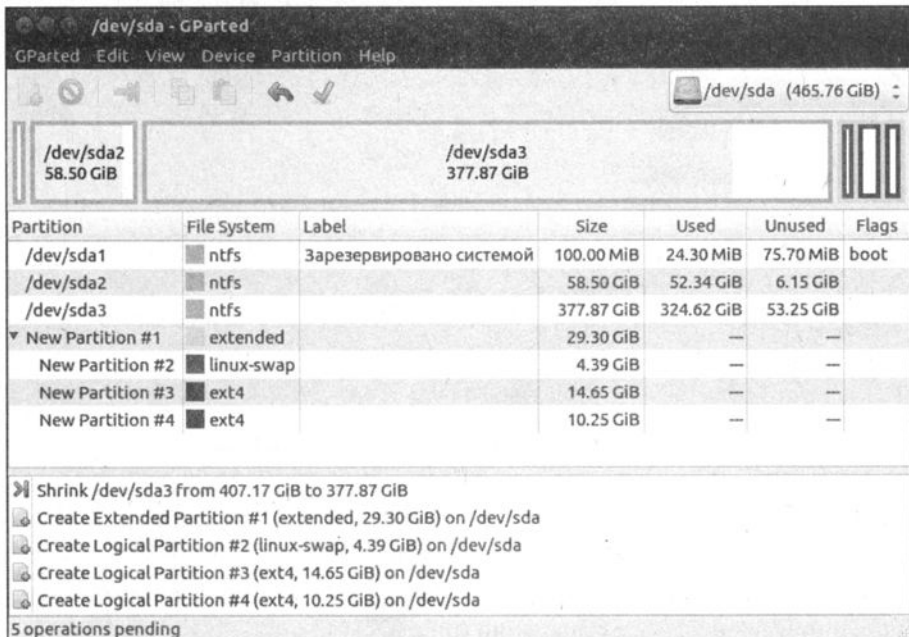


Рис. 3.14. Финальный этап создания разделов для установки ОС Ubuntu

Все разделы созданы, и нам остается только выполнить запланированные операции по разбивке жесткого диска на разделы. Для этого на панели инструментов есть кнопка **Apply All Operations** с изображенным на ней зеленым флажком.

ПРИМЕЧАНИЕ

Если вы хотите отменить проделанную операцию, то нажмите кнопку **Undo Last Operations**, которая находится на панели инструментов и обозначена красной стрелкой, направленной влево.

Нажмите кнопку **Apply All Operations** для завершения и сохранения всех изменений. В ответ на это появится предупреждение о том, что в случае неправильного редактирования разделов вы рискуете потерять все данные (рис. 3.15).

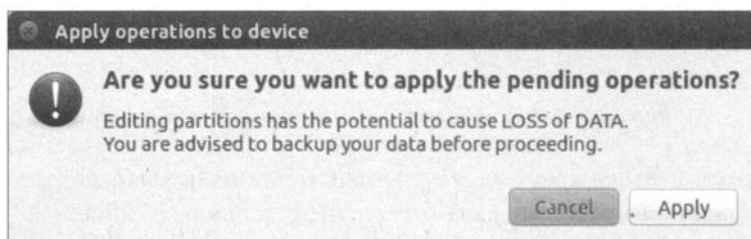


Рис. 3.15. Подтверждение проделанных изменений

Нажмите кнопку **Apply**, и появится окно, отображающее ход выполнения изменений (рис. 3.16).

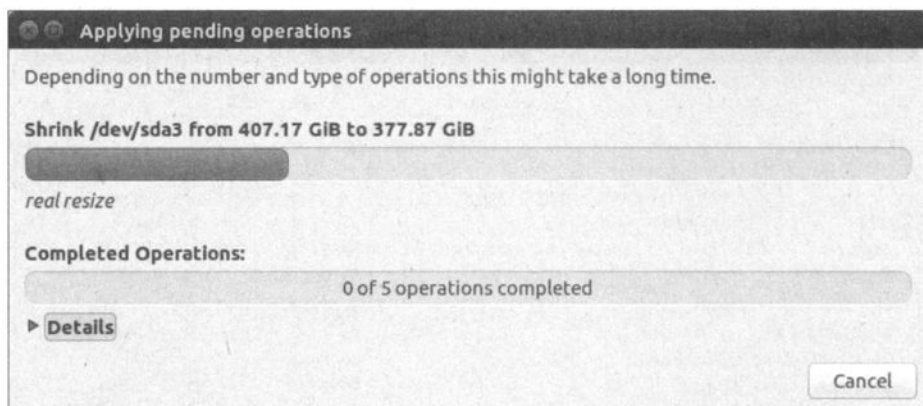


Рис. 3.16. Процесс разметки созданных разделов

Этот процесс довольно долгий, поэтому можно успеть сходить попить чай или запустить браузер Firefox прямо из Live-режима и почитать что-нибудь в Сети.

По окончании процесса разметки жесткого диска будет выведено окно с сообщением об успешном процессе всех операций (рис. 3.17).

Теперь разметка жесткого диска выглядит следующим образом (рис. 3.18).

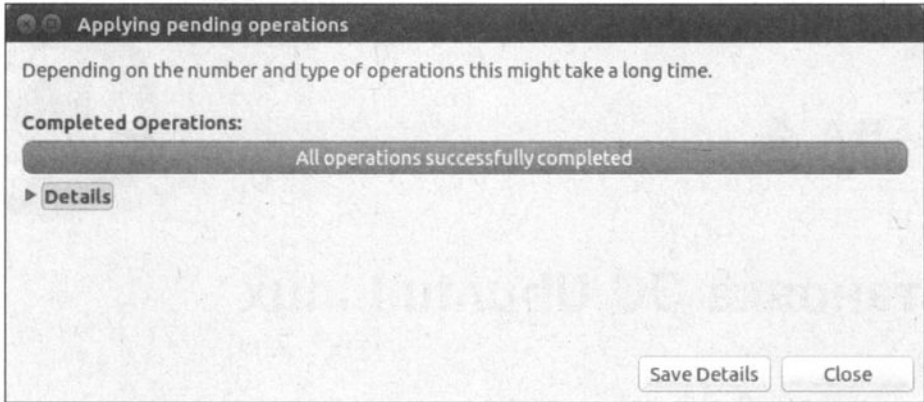


Рис. 3.17. Программа GParted сообщила об успешном завершении всех операций

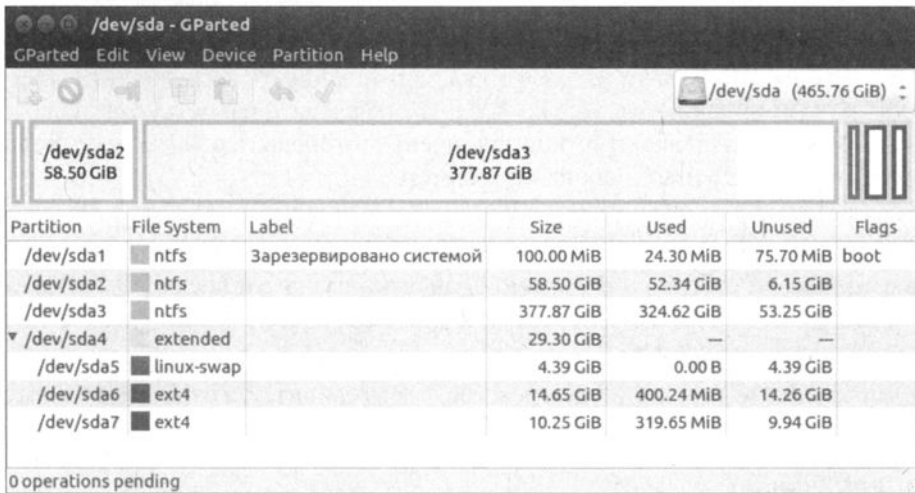


Рис. 3.18. Созданные разделы на жестком диске

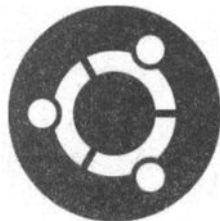
Как оказалось, нет ничего сложного в ручной разметке жесткого диска. Все операции выполняются довольно легко и интуитивно понятны. Главное при этом — соблюдать осторожность во избежание случайной потери данных при неправильной разметке жесткого диска.

Выводы

Мы научились загружать компьютер с загрузочного диска и пользоваться Live-режимом, который поддерживается загрузочным диском ОС Ubuntu.

Мы также научились создавать разделы для установки операционной системы Ubuntu Linux по соседству с ОС Windows.

ГЛАВА 4



Установка ОС Ubuntu Linux

Запуск операционной системы в Live-режиме

Вот мы и подошли к самому процессу установки ОС Ubuntu. Нам пришлось пройти путь от загрузки ОС с сайта разработчика, затем записать загруженный дистрибутив с ОС на DVD-диск и уже только после этого сделать разметку жесткого диска. На этом подготовительные действия закончены, и теперь перейдем непосредственно к установке ОС Ubuntu Linux на компьютер.

В предыдущей главе мы немного познакомились с Live-режимом и запускали только программу GParted. Однако Live-режим имеет гораздо более продвинутые возможности, нежели работа с программой разметки жесткого диска GParted. Помимо знакомства с ОС Ubuntu, Live-режим позволяет произвести установку ОС на жесткий диск вашего компьютера.

Итак, запустите ОС Ubuntu в Live-режиме, и вы увидите уже знакомый нам рабочий стол (рис. 4.1).

ПРИМЕЧАНИЕ

Обратите внимание, что на панели запуска присутствует та же самая кнопка.

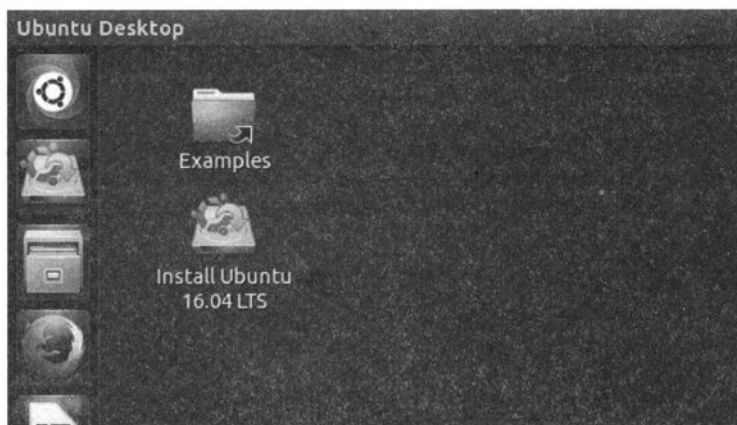


Рис. 4.1. Рабочий стол ОС Ubuntu

На рабочем столе присутствуют два ярлыка:

- ◆ **Examples;**
- ◆ **Install Ubuntu 16.04 LTS.**

Сейчас нас интересует второй ярлык — **Install Ubuntu 16.04 LTS**, и для чего он предназначен, мы сейчас узнаем.

Установка операционной системы из Live-режима

Для запуска процесса установки операционной системы дважды щелкните левой кнопкой мыши по ярлыку **Install Ubuntu 16.04 LTS**.

Запустится мастер установки ОС Ubuntu (рис. 4.2).



Рис. 4.2. Мастер установки ОС Ubuntu предлагает выбрать язык установки ОС

Здесь нам предлагается выбрать основной язык для ОС Ubuntu. В левой части окна выберите вариант **Русский** и нажмите кнопку **Продолжить**.

На следующем этапе установки вам будет предложено загрузить обновления во время установки Ubuntu и установить стороннее программное обеспечение для видеокарт и устройств Wi-Fi, а также Flash, MP3 и других медиаданных (рис. 4.3).

Если ваш компьютер подключен к Интернету, то можно загрузить обновления во время установки операционной системы. Этот вариант является необязательным, т. к. загрузить обновления можно и после установки ОС.

При выборе второго варианта устанавливается стороннее программное обеспечение, которое необходимо для воспроизведения некоторых форматов мультимедиа,

а также будут установлены некоторые драйверы устройств. Этот вариант также не является обязательным, потому что установить стороннее программное обеспечение можно и после установки ОС.

Отметьте нужные вам пункты и нажмите кнопку **Продолжить**.

На следующем этапе будет предложено выбрать тип установки операционной системы (рис. 4.4).

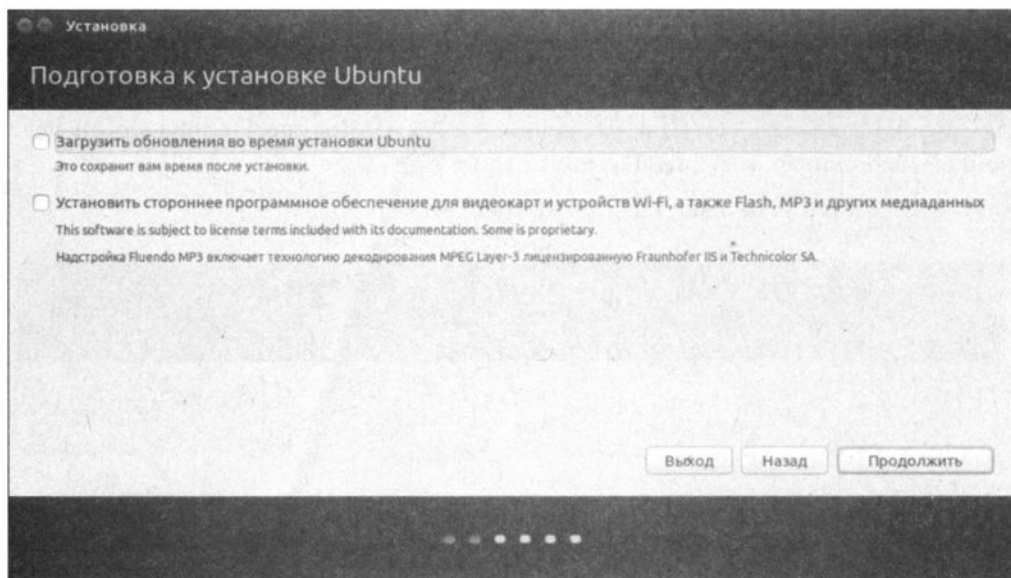


Рис. 4.3. Выбор загрузки обновлений и стороннего программного обеспечения

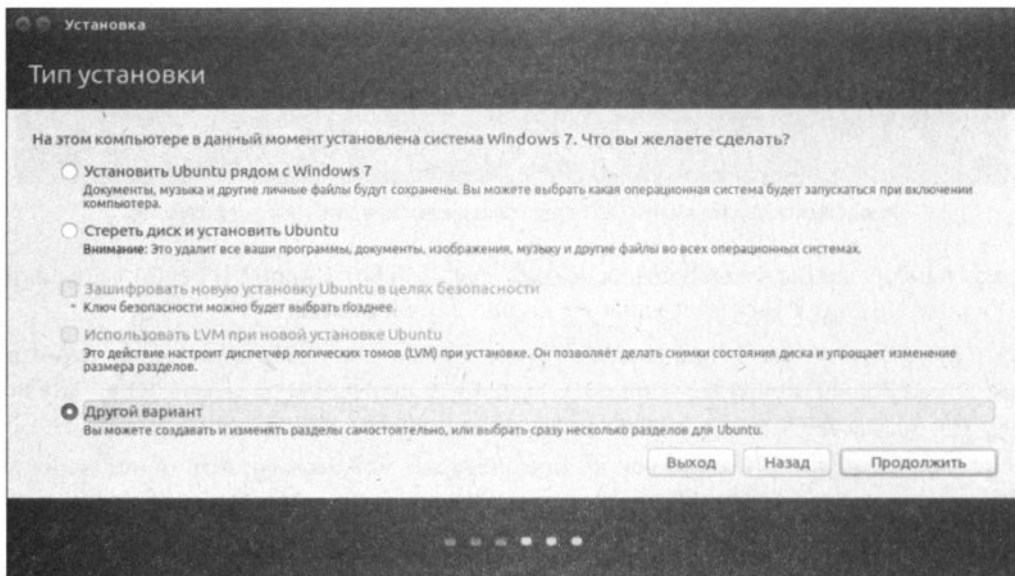


Рис. 4.4. Выбор типа установки операционной системы

Здесь нам предлагается на выбор три типа установки.

- ♦ **Установить Ubuntu рядом с Windows 7.** В этом случае сохраняются все личные документы, а также файлы ОС Windows. Однако данный тип установки самостоятельно разметит диск, и установка будет произведена в соответствии с предпочтениями самой ОС. Нам этот вариант не годится, т. к. мы уже самостоятельно разметили разделы диска.
- ♦ **Стереть диск и установить Ubuntu.** Этот вариант подходит лишь в том случае, когда вы устанавливаете ОС на новый компьютер либо хотите отдать весь диск для нужд ОС Ubuntu. Он нам также не подходит по той причине, что будут удалены все данные на жестком диске.
- ♦ **Другой вариант.** Данный вариант позволит самостоятельно выбрать разделы для установки операционной системы. Именно это нам и нужно.

Установите переключатель **Другой вариант** и нажмите кнопку **Продолжить**. Появится окно со списком разделов жесткого диска (рис. 4.5).

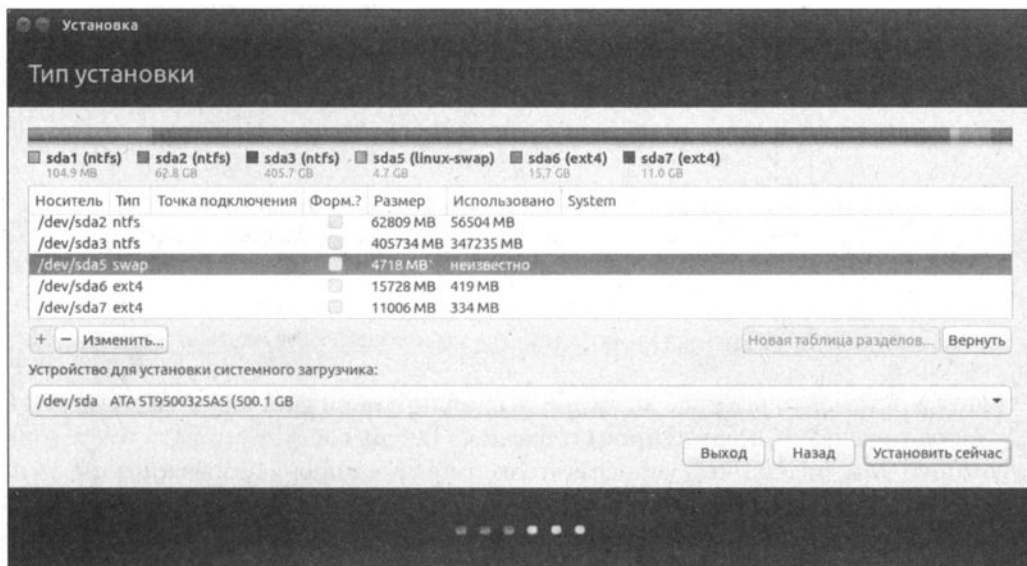


Рис. 4.5. Окно со списком разделов жесткого диска

В этом окне отображаются уже созданные нами разделы. Программа установки поддерживает некоторые операции изменения разметки, которые есть в программе GParted, однако не все.

Сейчас многие зададутся вопросом: зачем мы запускали программу GParted и делали разметку диска, если можно было все сделать во время установки?

Ответ на этот вопрос прост: установка ОС Ubuntu Linux будет производиться "по соседству" с ОС Windows. Если же выбрать один из первых двух вариантов установки, то мы рискуем потерять все данные, в том числе и установленную операционную систему Windows.

Итак, выделите раздел, отведенный для файла подкачки (в нашем случае он называется `/dev/sda5`) и нажмите кнопку **Изменить**. Появится окно редактирования раздела (рис. 4.6).

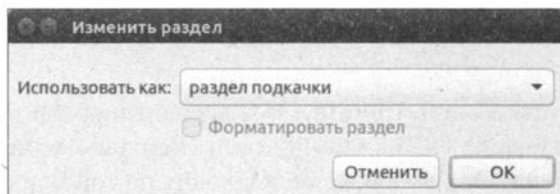


Рис. 4.6. Изменение раздела для файла подкачки

Убедитесь в том, что выбрано значение **раздел подкачки**, и нажмите кнопку **ОК**.

Теперь выделите раздел, предназначенный для монтирования корневой файловой системы (в нашем случае он называется `/dev/sda6`) и нажмите кнопку **Изменить**. Появится окно редактирования раздела корневой файловой системы (рис. 4.7).

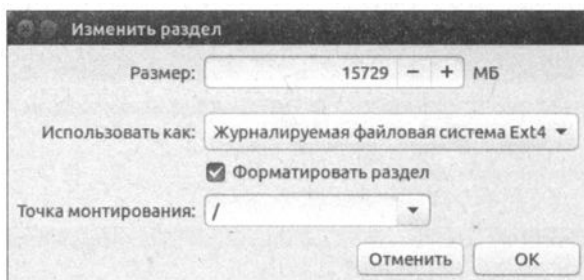


Рис. 4.7. Редактирование раздела для корневой файловой системы

Убедитесь в том, что выбрано значение **Журналируемая файловая система Ext4**, и установите флажок **Форматировать раздел**. Теперь следует выбрать точку монтирования. Мы знаем, что корневая точка монтирования обозначается прямым слешем, поэтому в раскрывающемся списке **Точка монтирования** выберите соответствующее значение, т. е. `/`. Нажмите кнопку **ОК**.

Если вы создали раздел для хранения личных документов, т. е. раздел `/home`, то выделите и его (в нашем случае это раздел `/dev/sda7`), после чего нажмите кнопку **Изменить**.

Появится уже знакомое окно изменения раздела. Выберите значение **Журналируемая файловая система Ext4** и установите флажок **Форматировать раздел**, а точку монтирования задайте равной значению `/home`. Нажмите кнопку **ОК** для сохранения изменений.

Окончательная разметка разделов жесткого диска выглядит так, как представлено на рис. 4.8.

Так как у нас на компьютере установлена только ОС Windows и имеется один жесткий диск, то в раскрывающемся списке **Устройство для установки системного**

загрузчика ничего менять не понадобится. Оставьте все как есть по умолчанию, т. е. устройство `/dev/sda`, и нажмите кнопку **Установить сейчас**. Появится окно подтверждения изменений (рис. 4.9).

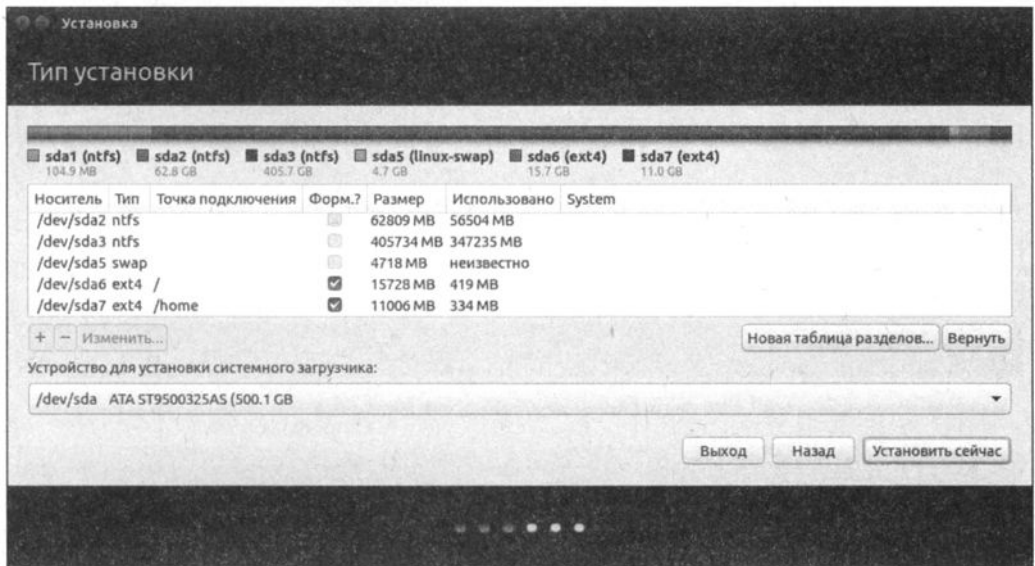


Рис. 4.8. Окончательная разметка разделов жесткого диска

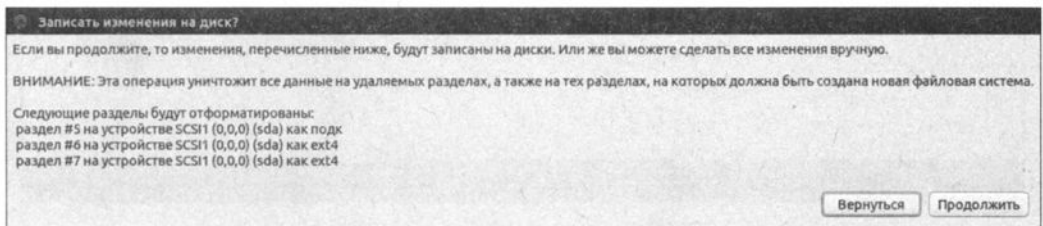


Рис. 4.9. Окно подтверждения проделанных изменений

Нажмите кнопку **Продолжить**. Появится окно выбора часового пояса компьютера (рис. 4.10). Укажите часовой пояс, выбрав нужный город на карте либо введя нужное значение в поле ввода, и нажмите кнопку **Продолжить**.

На следующем шаге установки потребуется выбрать раскладку клавиатуры (рис. 4.11). Выберите нужную раскладку клавиатуры и нажмите кнопку **Продолжить**.

Следующим шагом будет задание имени пользователя компьютера и его пароля (рис. 4.12).

На данном этапе предлагается создать пользователя компьютера. Именно этот пользователь (вернее, его учетная запись) будет являться аналогом учетной записи администратора в ОС Windows.

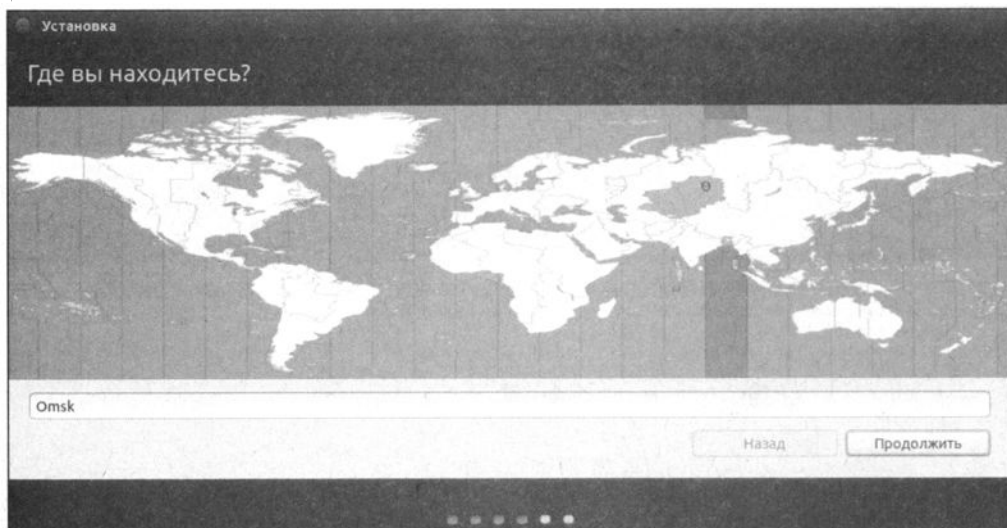


Рис. 4.10. Выбор часового пояса компьютера

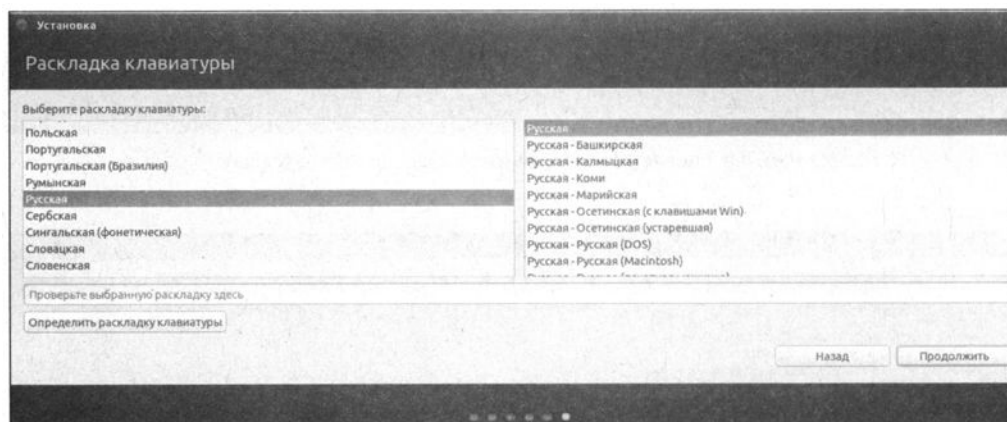


Рис. 4.11. Установщик ОС Ubuntu предлагает выбрать раскладку клавиатуры

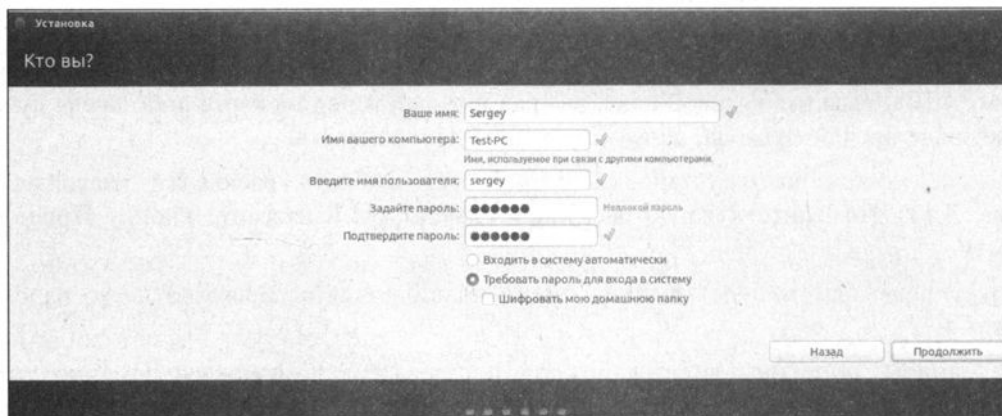


Рис. 4.12. Создание пользователя компьютера и его пароля

Поле ввода **Ваше имя** заполнять не обязательно, но для завершенности процесса давайте зададим имя. При этом автоматически заполнятся поля **Имя вашего компьютера** и **Введите имя пользователя**. Однако вы вправе изменить эти два поля или оставить как есть.

Стоит отметить, что логин, т. е. содержимое поля **Введите имя пользователя**, должен вводиться только латинскими буквами и желательно в нижнем регистре.

Следует также задать пароль пользователя. Сильно слабый (короткий) пароль будет отмечен как "Плохой пароль", однако это не мешает процессу установки.

ПРИМЕЧАНИЕ

Рекомендуется выбирать устойчивый пароль, иначе слабый пароль будет очень легко подобрать, что ставит под угрозу защиту ваших персональных данных на компьютере.

Переключатель **Входить в систему автоматически** следует устанавливать только в случае, когда вы уверены, что кроме вас вашим компьютером никто пользоваться не будет. В противном случае, рекомендуется установить переключатель в положение **Требовать пароль для входа в систему**.

Устанавливать ли флажок **Шифровать мою домашнюю папку** — решать вам. Разумеется, при этом придется вводить пароль при каждом входе в систему. Шифрование папки позволит защитить ваши личные данные извне, т. е. если кто-то попытается прочесть ваши данные, загрузившись с Live-CD, то это ему не удастся.

После того как вы задали нужные вам параметры безопасности, нажмите кнопку **Продолжить**. Начнется процесс установки ОС Ubuntu Linux (рис. 4.13).

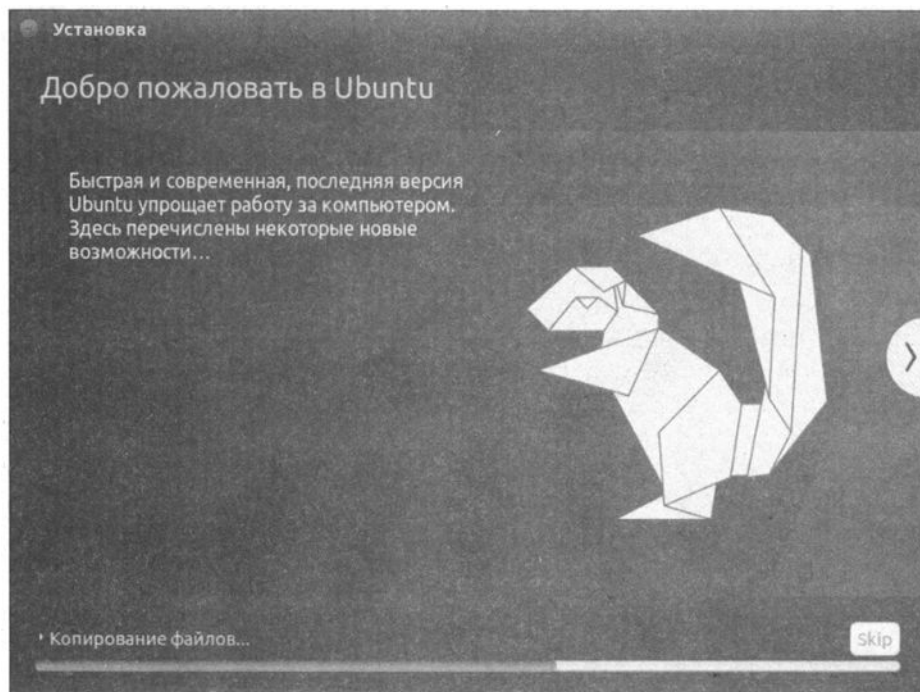


Рис. 4.13. Начался процесс установки ОС Ubuntu Linux

Во время установки и копирования файлов вам будет показана информация о возможностях системы в виде слайд-шоу.

По окончании установки появится окно с сообщением об успешном завершении процесса (рис. 4.14).

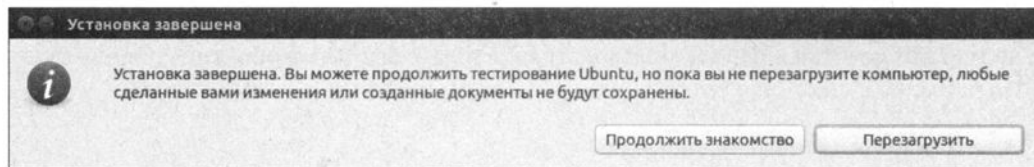


Рис. 4.14. Установщик сообщил об успешной установке ОС Ubuntu

Нажмите кнопку **Перезагрузить** и извлеките из компьютера дистрибутив с ОС Ubuntu.

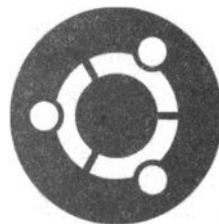
На данном этапе установка операционной системы закончена, и пора приступить к ее первому запуску уже с жесткого диска компьютера.

Выводы

Мы научились устанавливать ОС Ubuntu на заранее созданные разделы жесткого диска по соседству с операционной системой Windows.

Мы также научились устанавливать точки монтирования для корневой файловой системы и для хранения личных данных и узнали, в какой последовательности они должны быть созданы.

ГЛАВА 5



Первый запуск и первоначальная настройка ОС Ubuntu

Запуск ОС Ubuntu

Вот и настал торжественный момент, когда ОС Ubuntu установлена, и теперь нам предстоит произвести ее первый запуск. Нажмите кнопку включения компьютера, и после процедуры POST (Power-On Self-Test) появится окно меню загрузчика GRUB (рис. 5.1).

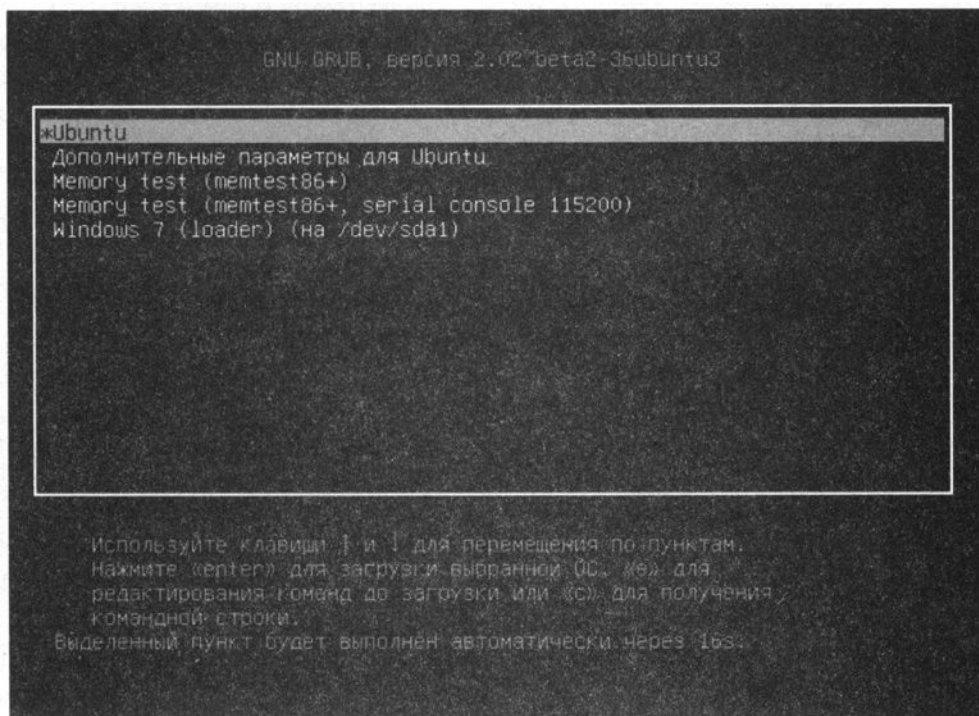


Рис. 5.1. Меню загрузчика GRUB

Загрузчик GRUB (англ. GRand Unified Bootloader) — это основной единый загрузчик операционных систем. Данный загрузчик используется в ОС на базе Linux, а также передает управление другим ОС, например ОС Windows.

В нашем случае GRUB по умолчанию загружает ОС Ubuntu, если не был выбран другой вариант загрузки. Если в течение определенного времени не выбрать другой вариант загрузки, то загрузится ОС Ubuntu. В случае выбора второго варианта (**Дополнительные параметры для Ubuntu**) будет открыто меню, которое позволяет загрузиться в режим восстановления ОС Ubuntu.

Третий и четвертый пункты загрузки предназначены для запуска проверки памяти, установленной на компьютере, а самый последний вариант позволяет запустить ОС Windows 7.

Выбор вариантов меню загрузки осуществляется клавишами-стрелками, а подтверждение выбора — клавишей <Enter>.

Выберите первый вариант или подождите несколько секунд, и будет загружена ОС Ubuntu Linux. Появится окно запроса ввода пароля для входа в систему (рис. 5.2).

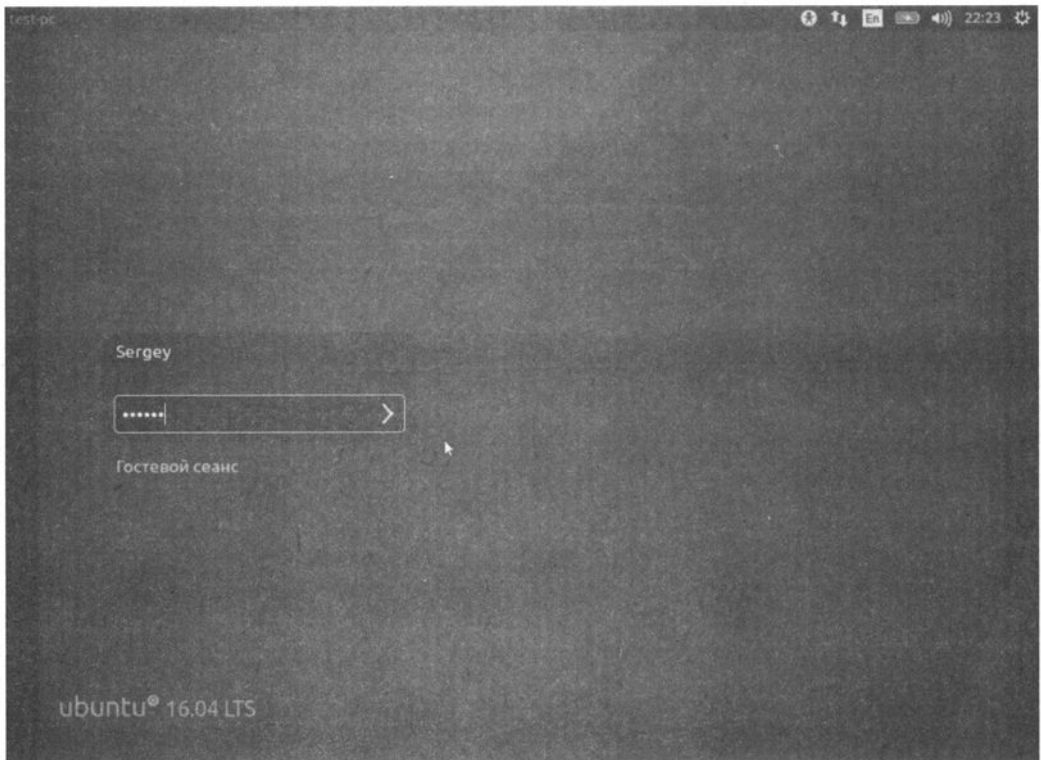









Рис. 5.2. ОС Ubuntu запрашивает пароль пользователя

В случае если при установке был выбран автоматический вход в систему без ввода пароля, то такое окно будет пропущено, и вы сразу попадете на *рабочий стол* ОС Ubuntu.

В верхней части окна, в правом верхнем углу, есть несколько значков:

- ♦  — выбор специальных возможностей, например, можно включить экранную клавиатуру и настроить контрастность;
- ♦  — текущие подключения позволяют выбрать подключение к сети Интернет;
- ♦  — индикатор языка клавиатуры позволит сменить язык для требуемого ввода символов для пароля;
- ♦  — значок батареи присутствует в случае, если ОС установлена на ноутбук, а не на стационарный компьютер;
- ♦  — индикатор звука позволит настроить уровень громкости звуков для всей ОС в целом;
- ♦  — индикатор времени;
- ♦  — и наконец, значок включения/выключения компьютера, который изображен в виде шестеренки с тумблером.

Ну что ж, введите свой пароль, заданный при установке ОС, и нажмите клавишу <Enter> (или щелкните левой кнопкой мыши по закрывающей угловой скобке). Вы попадете на рабочий стол ОС Ubuntu (рис. 5.3).

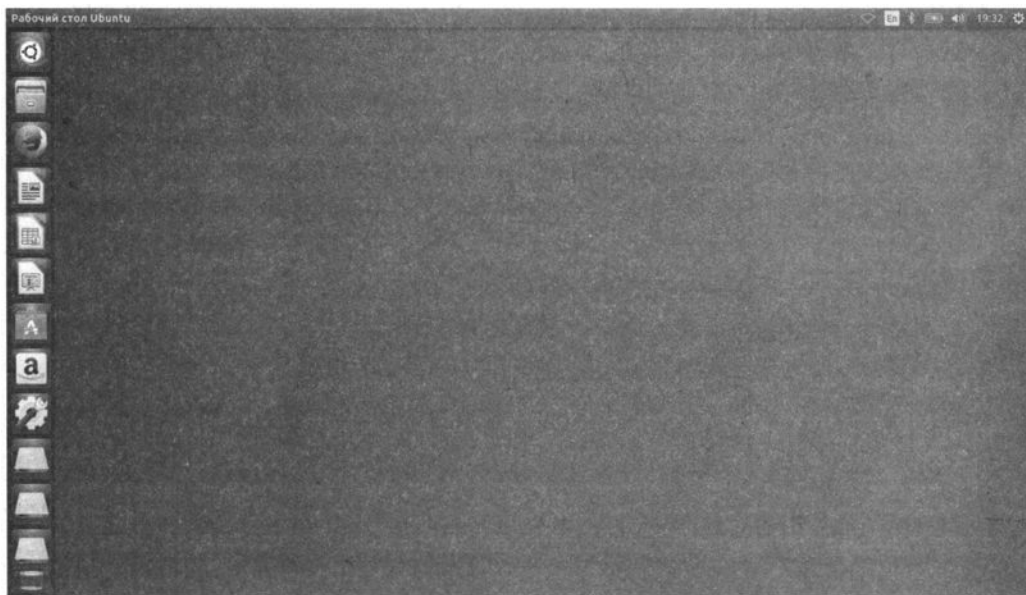


Рис. 5.3. Рабочий стол ОС Ubuntu

Рабочий стол выглядит так же, как при запуске в Live-режиме. Можно заметить, что исчез значок **Install Ubuntu 16.04 LTS** и интерфейс ОС теперь на русском языке.

ОС Ubuntu готова к работе, однако, как и любую операционную систему, ее необходимо настроить в соответствии со своими предпочтениями, но на данном этапе мы займемся настройкой подключения к Интернету.


Настройка подключения к Интернету в Ubuntu

Современный компьютер сложно представить без подключения к Сети. Так, Интернетом ежедневно пользуется каждый из нас, например, когда работаем с электронной почтой или общаемся в социальных сетях.

ОС Ubuntu, как и любая ОС, для своей работы использует ресурсы глобальной сети. Да, при установке мы пропустили подключение к Интернету, т. к. установка ОС Ubuntu может производиться автономно, однако для нормальной дальнейшей работы с операционной системой все же рекомендуется настроить подключение к Интернету.

В большинстве случаев ОС Ubuntu корректно определяет установленное сетевое оборудование, и дополнительных настроек не требуется. Тем не менее мы рассмотрим основные способы подключения к Сети.

Подключение к сети по Wi-Fi

Для управления подключениями к Интернету в ОС Ubuntu используется программа NetworkManager. Запустить ее можно прямо из меню состояния на панели задач, и выглядит она как значок Wi-Fi . Нажмите на данную кнопку (индикатор сети), и отобразится список доступных подключений (рис. 5.4). В этом меню отображаются доступные беспроводные подключения, а также уже активные (установленные) соединения.

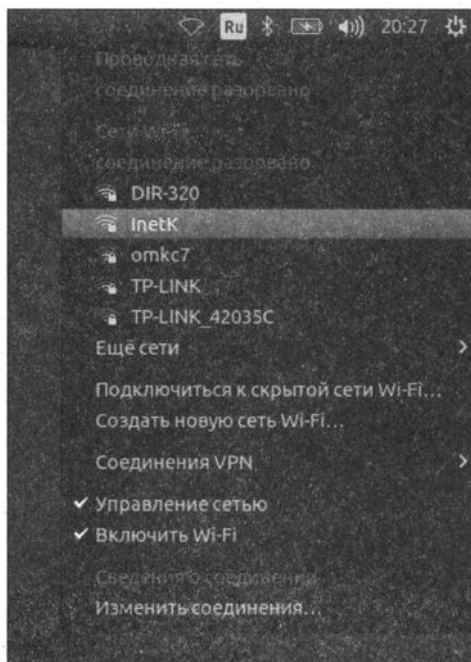


Рис. 5.4. Программа NetworkManager отобразила список доступных подключений

Для того чтобы подключиться к беспроводной точке доступа, щелкните левой кнопкой мыши по нужной беспроводной точке, и появится окно ввода пароля для доступа к запрашиваемому соединению (рис. 5.5).

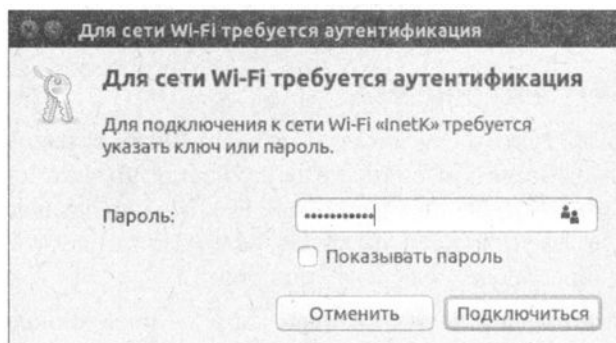


Рис. 5.5. Запрос пароля для подключения к сети Wi-Fi

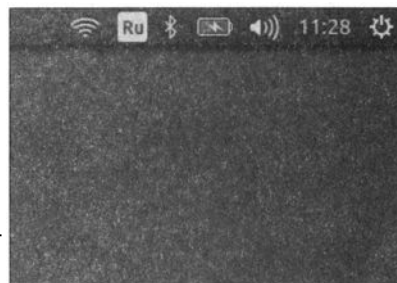


Рис. 5.6. Соединение Wi-Fi установлено

Введите пароль вашей точки доступа и нажмите кнопку **Подключиться**. Подключение к беспроводной сети будет установлено, и значок подключения изменится на привычный нам значок Wi-Fi (рис. 5.6).

Для того чтобы отключить текущее беспроводное соединение, нажмите на значок Wi-Fi и выберите команду **Отключить**. Адаптер Wi-Fi отключается при выборе пункта меню **Включить Wi-Fi**.

Настройка подключений в NetworkManager

Для настройки подключения нужно щелкнуть по значку индикатора сети, т. е. на индикаторе программы NetworkManager, и выбрать пункт меню **Изменить соединения....** Появится окно, в котором вы сможете управлять сетевыми соединениями (рис. 5.7).

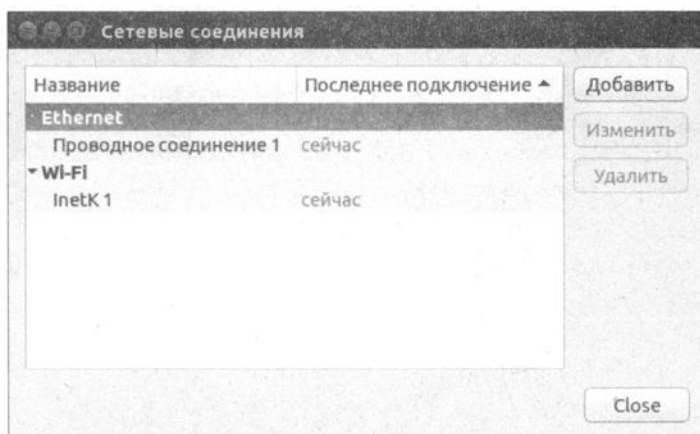


Рис. 5.7. Редактор сетевых соединений программы NetworkManager

В нашем случае в окне редактора соединений уже отображается сохраненное беспроводное подключение, к которому мы подключались ранее.

Нажатие кнопки **Добавить** позволяет создать новое подключение, такое как Ethernet, Bluetooth, DSL, Wi-Fi, VLAN, VPN, мобильное и т. п.

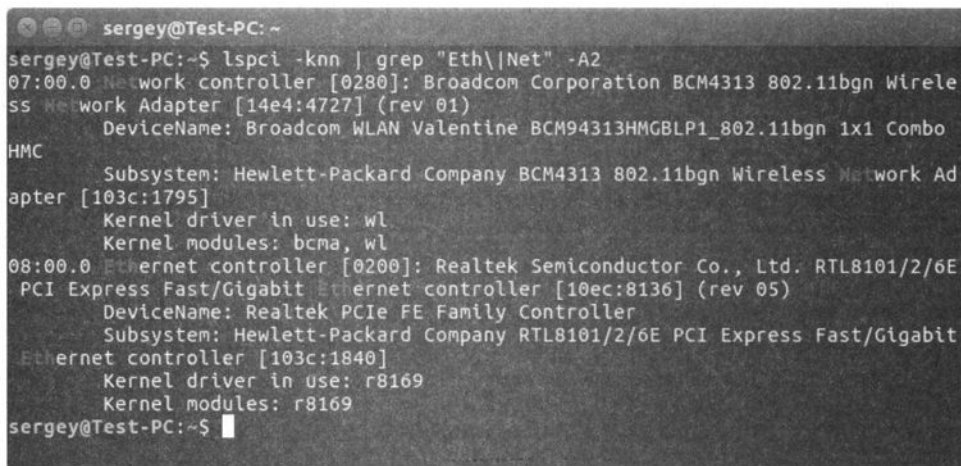
Если не определилась сетевая карта

Бывают случаи, когда ОС Ubuntu не может самостоятельно распознать сетевую карту, установленную в вашем компьютере. Да, такое тоже случается, и поэтому необходимо заранее позаботиться о подготовке драйверов для нее. Можно заранее скачать необходимые драйверы, а потом установить их в ОС Ubuntu. Но что делать, если вы не знаете, какой именно драйвер нужен и как его установить?

Для того чтобы в ОС Ubuntu узнать информацию о сетевой карте, откройте окно программы Терминал и выполните следующую команду:

```
lspci -knn | grep "Eth\|Net" -A2
```

Запустить Терминал можно по сочетанию клавиш <Ctrl>+<Alt>+<T> либо на рабочем столе щелкнуть правой кнопкой мыши и выбрать команду **Открыть терминал**. Откроется окно программы Терминал, в котором и нужно вводить приведенную выше команду (рис. 5.8).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ lspci -knn | grep "Eth\|Net" -A2  
07:00.0 Network controller [0280]: Broadcom Corporation BCM4313 802.11bgn Wirele  
ss Network Adapter [14e4:4727] (rev 01)  
    DeviceName: Broadcom WLAN Valentine BCM94313HMGBlP1_802.11bgn 1x1 Combo  
HMC  
    Subsystem: Hewlett-Packard Company BCM4313 802.11bgn Wireless Network Ad  
apter [103c:1795]  
    Kernel driver in use: wl  
    Kernel modules: bcma, wl  
08:00.0 Ethernet controller [0200]: Realtek Semiconductor Co., Ltd. RTL8101/2/6E  
PCI Express Fast/Gigabit Ethernet controller [10ec:8136] (rev 05)  
    DeviceName: Realtek PCIe FE Family Controller  
    Subsystem: Hewlett-Packard Company RTL8101/2/6E PCI Express Fast/Gigabit  
Ethernet controller [103c:1840]  
    Kernel driver in use: r8169  
    Kernel modules: r8169  
sergey@Test-PC:~$
```

Рис. 5.8. Терминал вывел список установленных сетевых адаптеров

В этом списке отображаются все установленные сетевые адаптеры. Нас интересуют следующие две строки:

```
07:00.0 Network controller [0280]: Broadcom Corporation BCM4313 802.11bgn  
Wireless Network Adapter [14e4:4727] (rev 01)  
...  
08:00.0 Ethernet controller [0200]: Realtek Semiconductor Co., Ltd.  
RTL8101/2/6E PCI Express Fast/Gigabit Ethernet controller [10ec:8136] (rev 05)
```

Далее выполните поиск в Интернете с другого компьютера по следующим запросам:

- ◆ BCM4313 драйверы для Ubuntu;
- ◆ RTL8101/2/6E драйверы для Ubuntu.

А затем следуйте приведенным в поиске инструкциям по установке сетевого оборудования.

ПРИМЕЧАНИЕ

Чуть позже мы рассмотрим установку драйвера для сетевой карты Realtek, а также установку драйверов для беспроводного сетевого подключения.

Создание подключения Ethernet

Подключение типа Ethernet позволяет настроить подключение к Интернету посредством витой пары. Допустим, ваш провайдер проводит к вам в дом кабель типа "витая пара" и подключает его к вашему Wi-Fi-роутеру, который настраивает соответствующим образом. После этого к роутеру через витую пару можно подключить компьютер, который не имеет Wi-Fi-адаптера. Давайте создадим такое подключение.

Итак, для начала подключите к сетевой карте вашего компьютера Ethernet-кабель, посредством которого вы хотите подключиться к Интернету через ваш роутер, а затем выберите пункт меню **Изменить соединения...** программы NetworkManager и нажмите кнопку **Добавить**. Откроется окно выбора типа соединения (рис. 5.9).

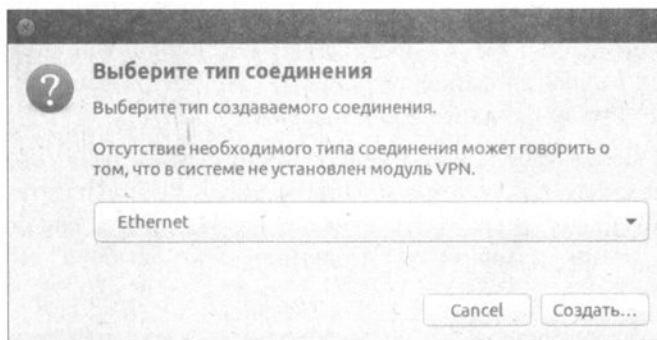


Рис. 5.9. Создание нового подключения

По умолчанию выбран именно вариант **Ethernet**. Нажмите кнопку **Создать...**. Откроется окно создания проводного соединения (рис. 5.10).

В списке выбора **Устройство** выберите вашу сетевую карту, к которой вы подключили Ethernet-кабель, и нажмите кнопку **Сохранить**.

В большинстве случаев этих настроек достаточно. Однако если ваш роутер настроен специфическим образом, то вам придется вручную вводить IP-адрес и т. п. Сказать, какие именно настройки вам придется задавать в этом случае, невозможно, т. к. все зависит от конкретного провайдера, услугами которого вы пользуетесь.

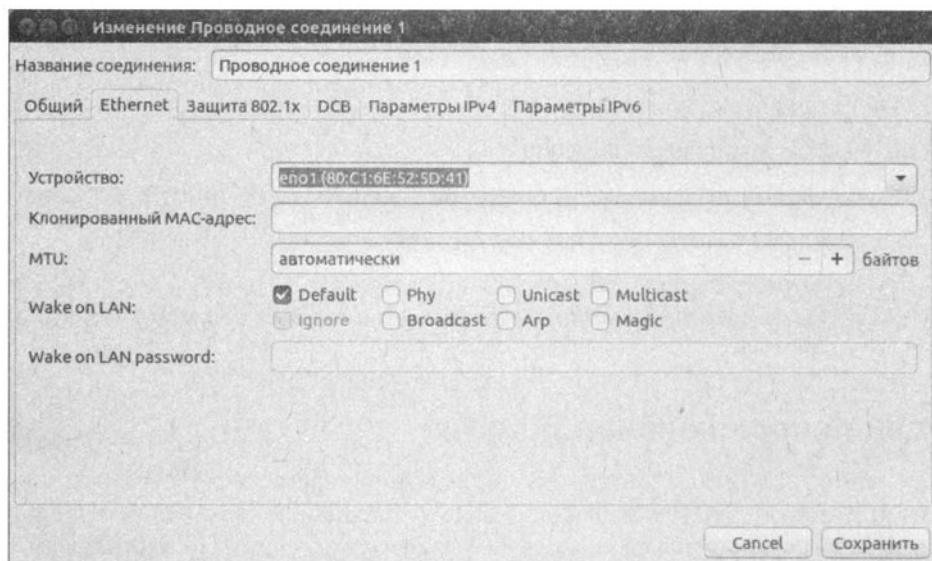


Рис. 5.10. Создание проводного соединения

В любом случае техподдержка провайдера всегда готова дать консультацию по этому вопросу.

Подключение через DSL

Бывает, что подключение к Интернету производится с помощью DSL-модема. В этом случае кабель от DSL-модема вставляется в сетевую карту вашего компьютера, и уже на нем выполняется определенная настройка. Вся информация по настройке выдается вашим провайдером. Обычно это логин и пароль. В редких случаях приходится задавать IP-адрес и DNS-сервер.

Итак, вставьте Ethernet-кабель в соответствующий разъем на сетевой карте вашего компьютера и нажмите на значок индикатора сети, т. е. щелкните по значку программы NetworkManager, и выберите пункт меню **Изменить соединения....** После этого нажмите кнопку **Добавить**. Появится окно выбора типа соединения (рис. 5.11).

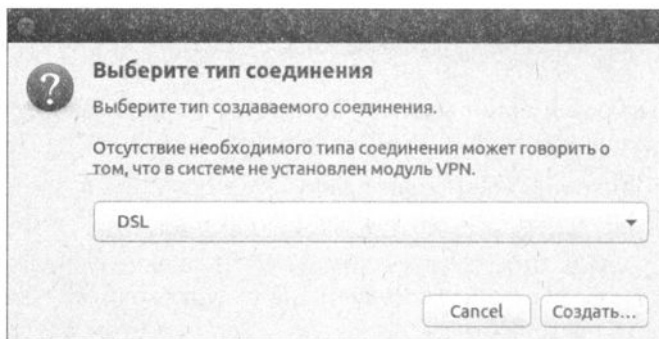


Рис. 5.11. Создание DSL-соединения

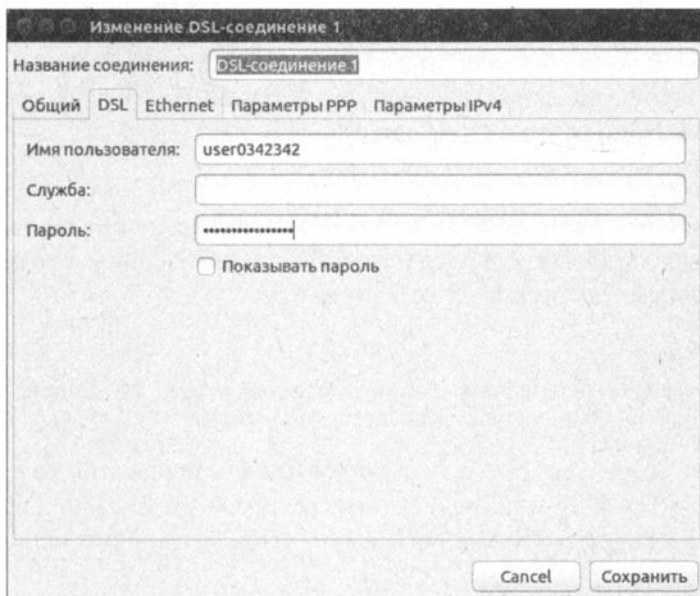


Рис. 5.12. Редактирование параметров DSL-соединения

В списке выбора типа соединения выберите вариант **DSL** и нажмите кнопку **Создать....** Появится окно редактирования параметров соединения (рис. 5.12).

В поле ввода **Имя пользователя** задайте логин, выданный вам при подключении к Интернету, а также пароль в одноименном поле. Перейдите на вкладку **Ethernet** (рис. 5.13).

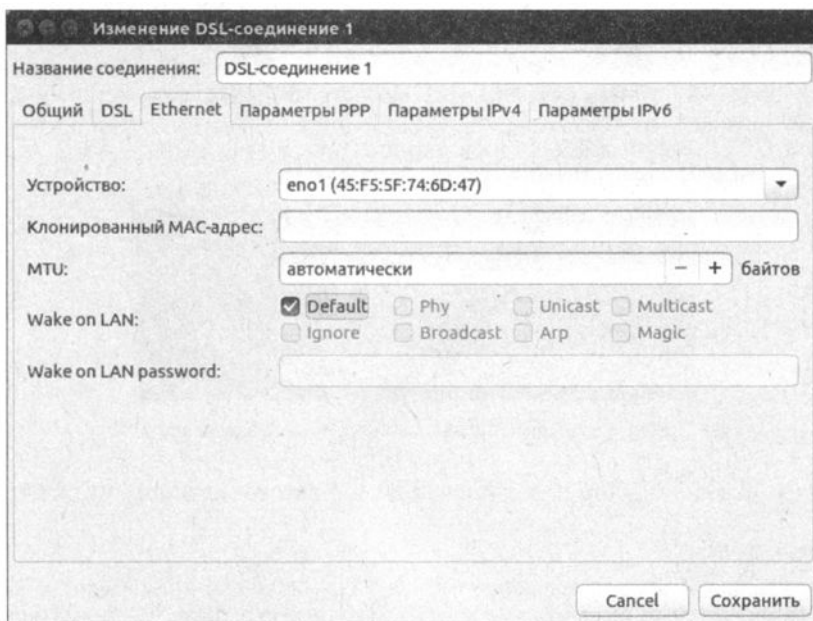


Рис. 5.13. Выбор сетевого устройства для доступа к Интернету

На этой вкладке необходимо задать сетевое устройство, к которому *подключен Ethernet*-кабель. Для этого в раскрывающемся списке **Устройство** выберите нужное сетевое устройство и нажмите кнопку **Сохранить**. Логическое имя вашего сетевого устройства будет примерно таким:

```
eno1
```

Наличие устройств в раскрывающемся списке **Устройство** свидетельствует о том, что ваша сетевая карта была успешно определена ОС Ubuntu, и вам не нужно искать и устанавливать драйверы для сетевой карты.

ПРИМЕЧАНИЕ

В скобках после логического имени устройства указан MAC-адрес (Media Access Control — управление доступом к среде) сетевой карты.

Теперь в списке сетей, которые можно просмотреть при нажатии на значке индикатора сети, появится DSL-соединение, имя которого вы задали или оставили по умолчанию при создании соединения. Вам остается только подключиться к созданному соединению.

Особый случай подключения через DSL

Встречаются ситуации, когда приведенный выше способ создания DSL-соединения не работает. То есть вы создали соединение, но оно отказывается работать. В этом случае можно воспользоваться альтернативным способом настройки соединения. Для этого запустите Терминал и выполните следующую команду:

```
sudo pppoeconf
```

ПРИМЕЧАНИЕ

Запустить Терминал можно по сочетанию клавиш <Ctrl>+<Alt>+<T> либо на рабочем столе щелкнуть правой кнопкой мыши и выбрать команду **Открыть терминал**.

В ответ на это Терминал запросит ваш пароль (рис. 5.14).

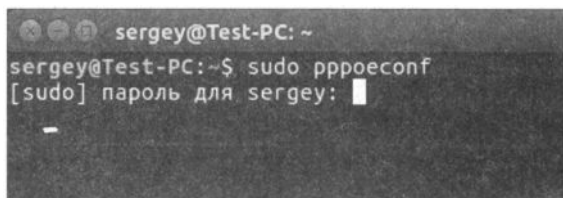


Рис. 5.14. Терминал запросил пароль пользователя

Введите пароль, заданный при установке ОС, и нажмите клавишу <Enter>.

ПРИМЕЧАНИЕ

Обратите внимание, что при вводе пароль в Терминале не отображается. При этом не отображаются даже символы звездочек или точек. Не пугайтесь и внимательно вводите пароль.

После ввода пароля появится программа настройки PPPoE-соединений `pppoeconf`, которая начнет сканирование устройств (рис. 5.15).

По окончании сканирования программа настройки сообщит о готовности к внесению изменений (рис. 5.16).

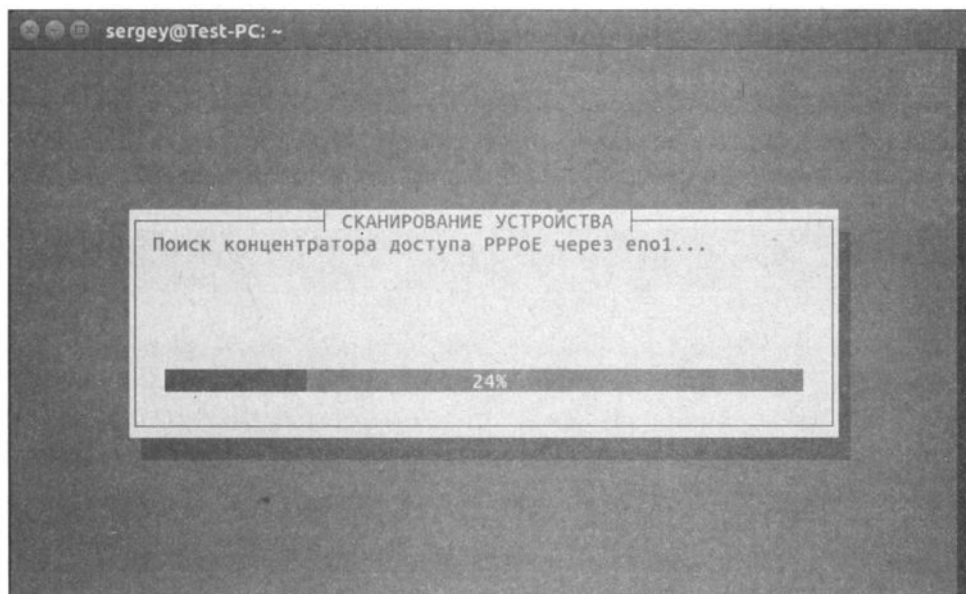


Рис. 5.15. Программа настройки PPPoE-соединений

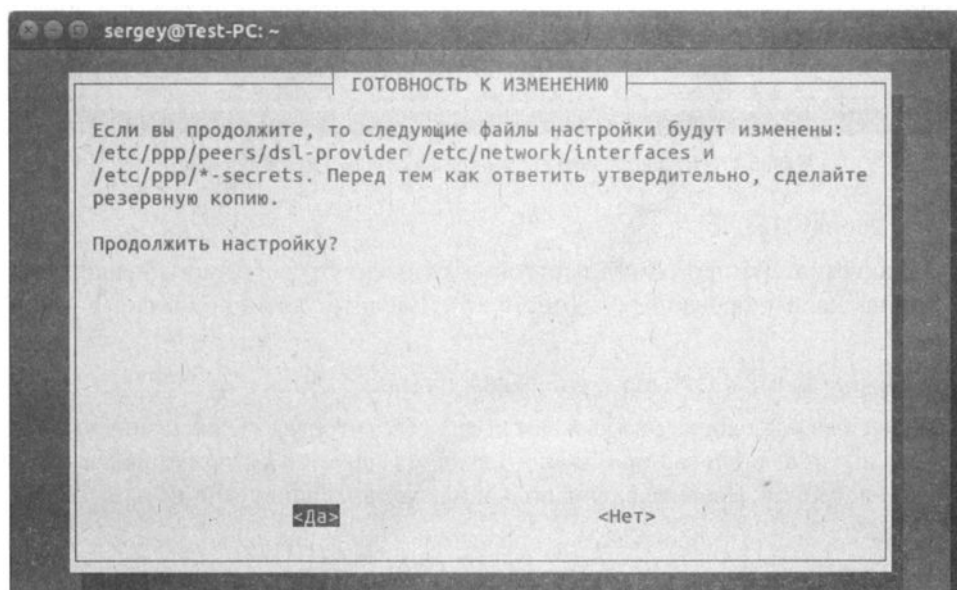


Рис. 5.16. Программа `pppoeconf` сообщила о готовности к внесению изменений

Изменения затронут следующие файлы:

- ◆ /etc/ppp/peers/dsl-provider;
- ◆ /etc/network/interfaces;
- ◆ /etc/ppp/*-secrets.

В эти файлы будут внесены изменения в соответствии с заданными позже настройками. Нажмите клавишу <Enter>, чтобы утвердительно ответить на вопросы и перейти к следующему шагу настройки.

На следующем шаге программа настройки запросит у вас удаление опции **nodetach** и попросит добавить опции **defaultroute** (служит для определения маршрута по умолчанию) и **noauth** (не требовать аутентификации от PPPoE-сервера) (рис. 5.17).

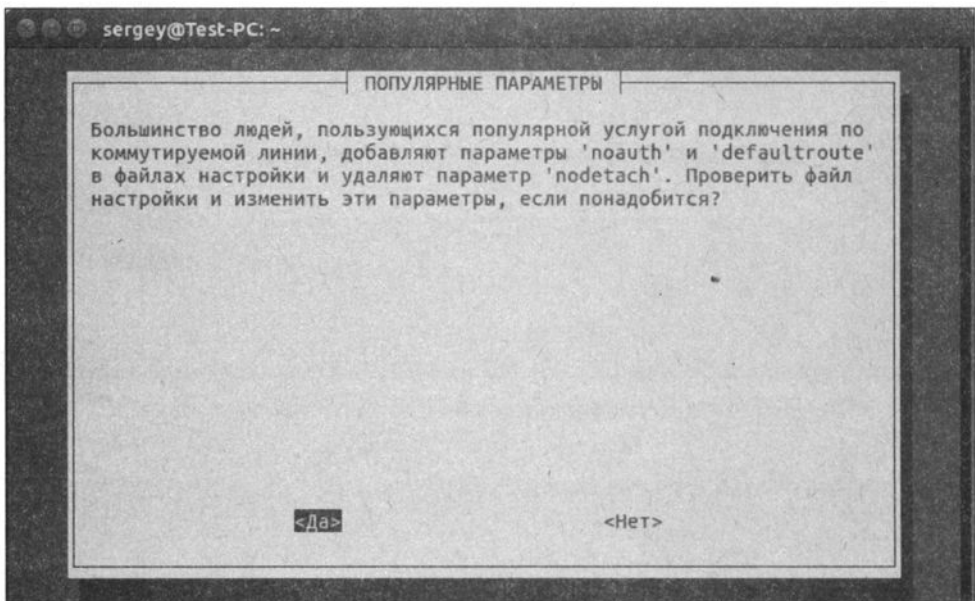


Рис. 5.17. Использование дополнительных опций для соединения

Нажмите кнопку **Да**.

На следующем этапе программа настройки запросит от вас ввода логина, который был выдан вашим провайдером (рис. 5.18). Введите логин и нажмите клавишу <Enter>.

Следующим этапом будет запрос ввода пароля (рис. 5.19).

При вводе символы пароля отображаются для того, чтобы избежать ошибки. Будьте внимательны, т. к. в случае ошибочного задания пароля вам придется заново производить настройку. Введите ваш пароль и нажмите клавишу <Enter>.

Следующим шагом программа `pppoeconf` попросит включения в файл `/etc/resolv.conf` IP-адресов DNS-сервера (рис. 5.20).

Нажмите кнопку **Да** для перехода к очередному шагу (рис. 5.21).

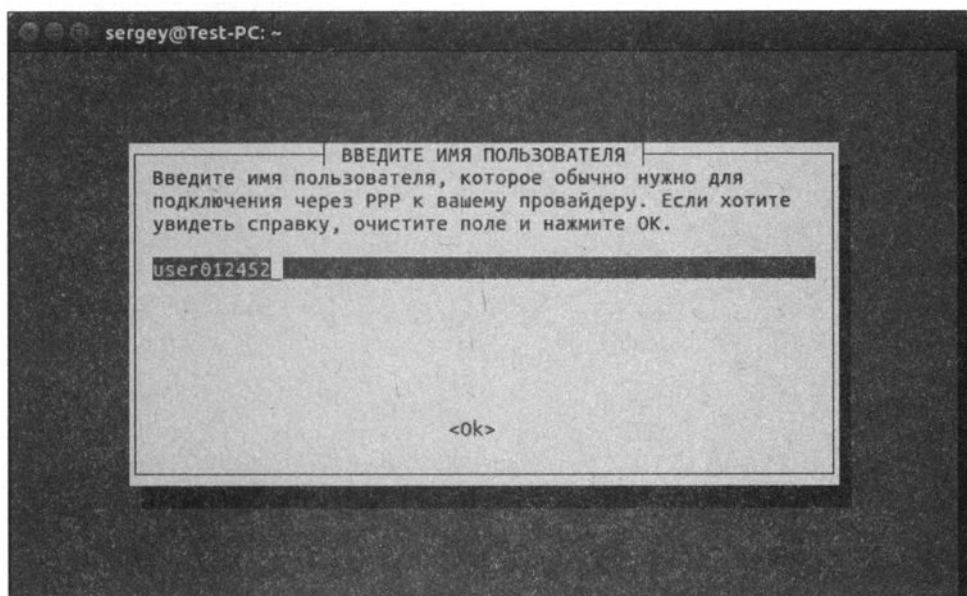


Рис. 5.18. Программа настройки запросила логин для подключения

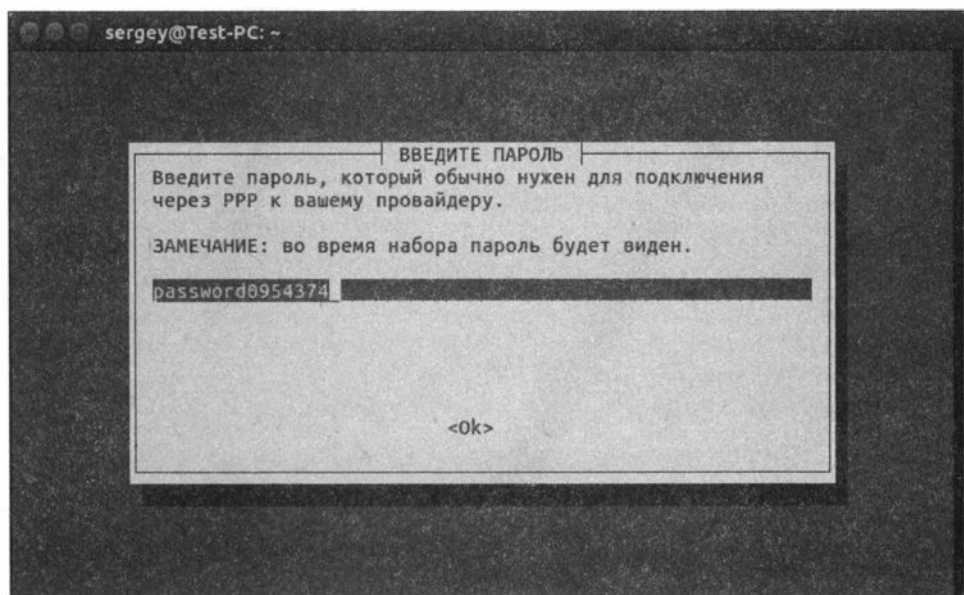


Рис. 5.19. Запрос ввода пароля

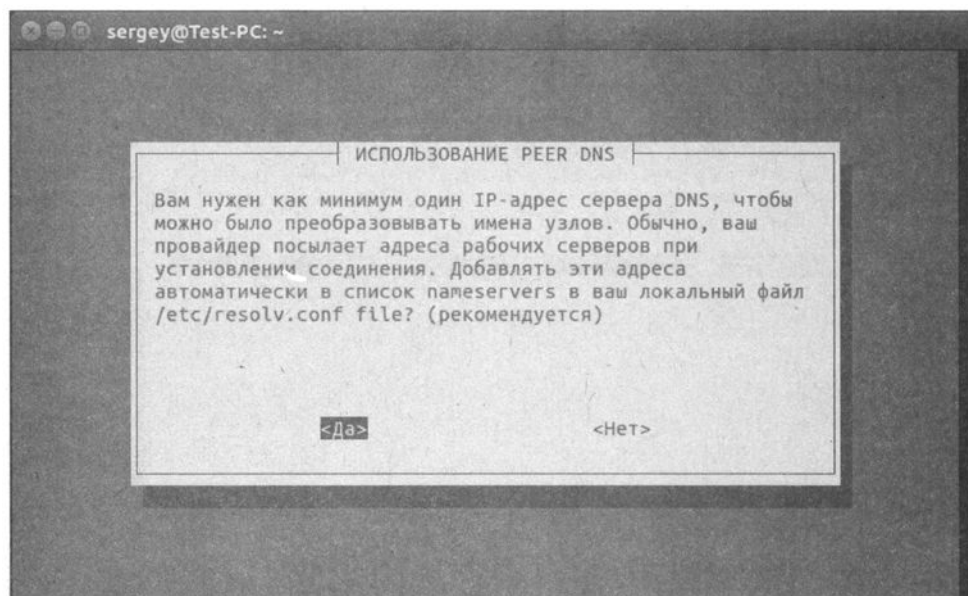


Рис. 5.20. Программа настройки запрашивает добавление адреса DNS-сервера

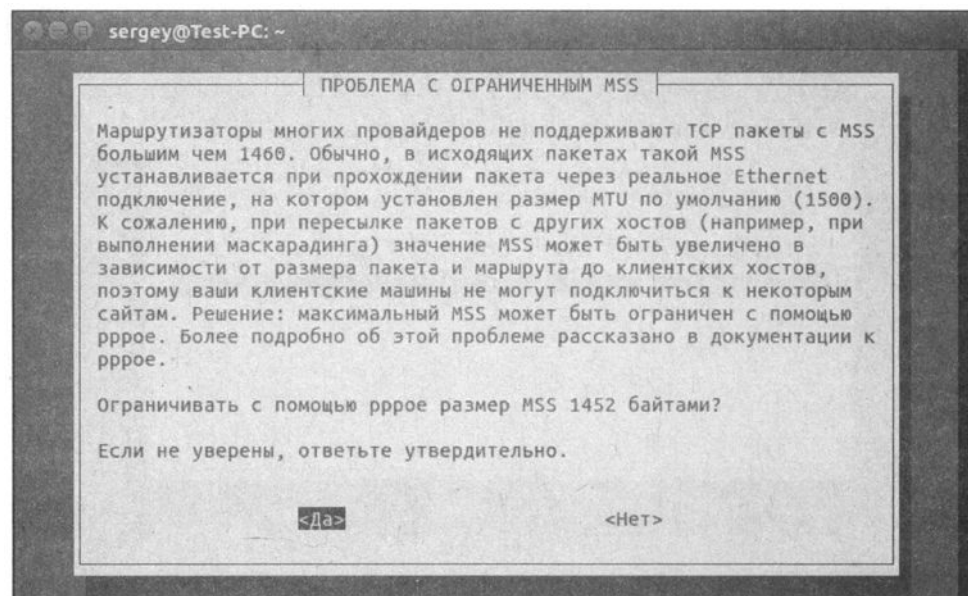


Рис. 5.21. Ограничение размера MSS

На этом шаге мастер настройки предлагает установить максимальный размер полезного блока для TCP-пакета равным 1452 байтам. Нажмите кнопку **Да**.

Далее программа настройки запросит установку соединения при загрузке ОС (рис. 5.22).

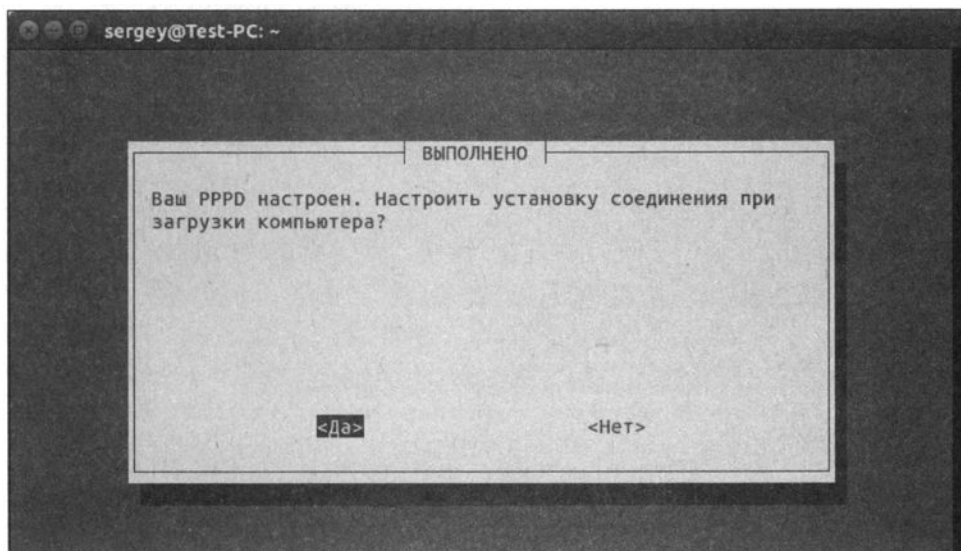


Рис. 5.22. Программа настройки предлагает устанавливать соединение при загрузке ОС

Выберите нужный вариант и подтвердите выбор клавишей **<Enter>**. После этого программа `pproconf` попросит установить соединение (рис. 5.23). Нажмите кнопку **Да**.

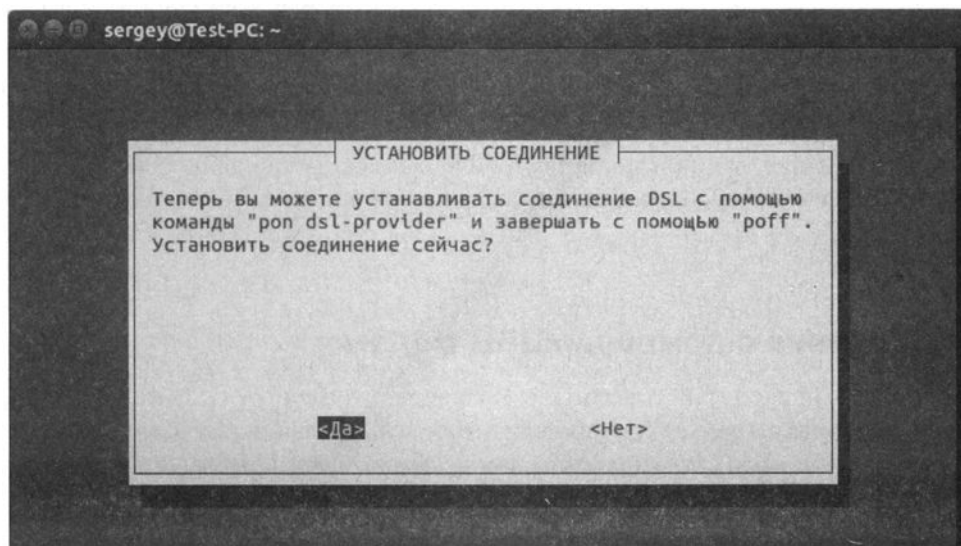


Рис. 5.23. Запрос установки соединения

ПРИМЕЧАНИЕ

В дальнейшем в случае необходимости вы в любой момент сможете разорвать соединение, выполнив в Терминале команду `sudo poff dsl-provider`. Установить DSL-соединение можно командой `sudo pon dsl-provider`.

На последнем этапе программа настройки сообщит о том, что для просмотра состояния соединения можно воспользоваться командой `plog`, а для просмотра состояния активного интерфейса — командой `ip addr show ppp0` (рис. 5.24).

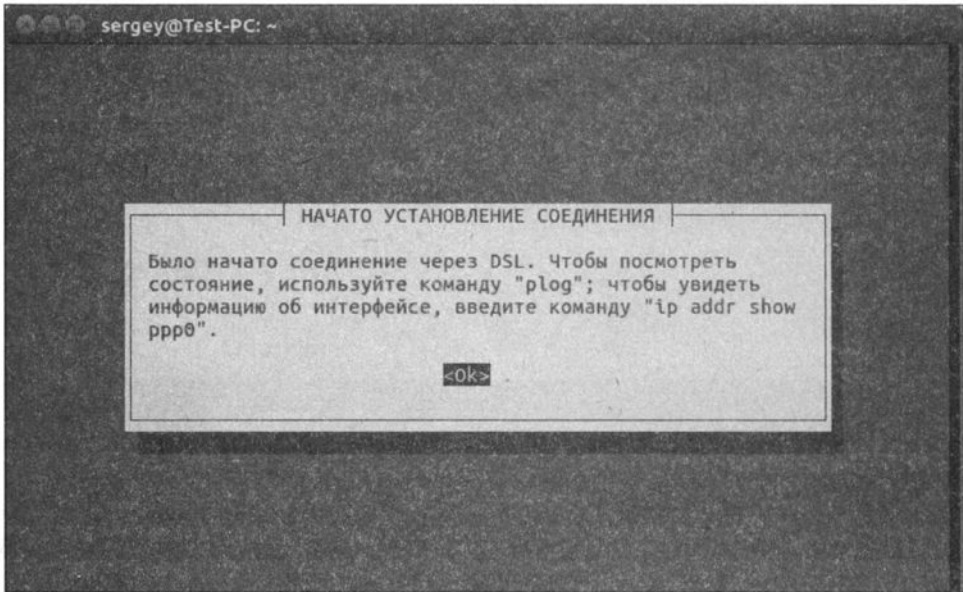


Рис. 5.24. Соединение установлено

Нажмите клавишу <Enter> для завершения настройки.

На этом настройка DSL-соединения будет завершена. Если вы все сделали правильно, то соединение с Интернетом будет установлено.

Обратите внимание, что при такой настройке в списке сетей данное сетевое соединение не будет отображаться. Для управления соединением используйте команды `sudo poff dsl-provider` и `sudo pon dsl-provider` для отключения и подключения соответственно.

Подключение с помощью USB-модема

Зачастую нам приходится выходить в Интернет с помощью USB-модема, например, когда мы находимся за городом, где недоступно проводное соединение, а имеется только мобильная сеть. Для этого нам необходим сам USB-модем, который мы сейчас и будем настраивать.

Подключите ваш USB-модем к компьютеру, нажмите на знакомый значок индикатора сети (программы NetworkManager) и выберите пункт меню **Изменить соеди-**

нения.... Затем нажмите кнопку **Добавить**. Откроется уже окно выбора типа соединения, в котором нужно задать тип соединения **Мобильный** и после этого нажать кнопку **Создать**.... Откроется окно создания мобильного подключения (рис. 5.25).

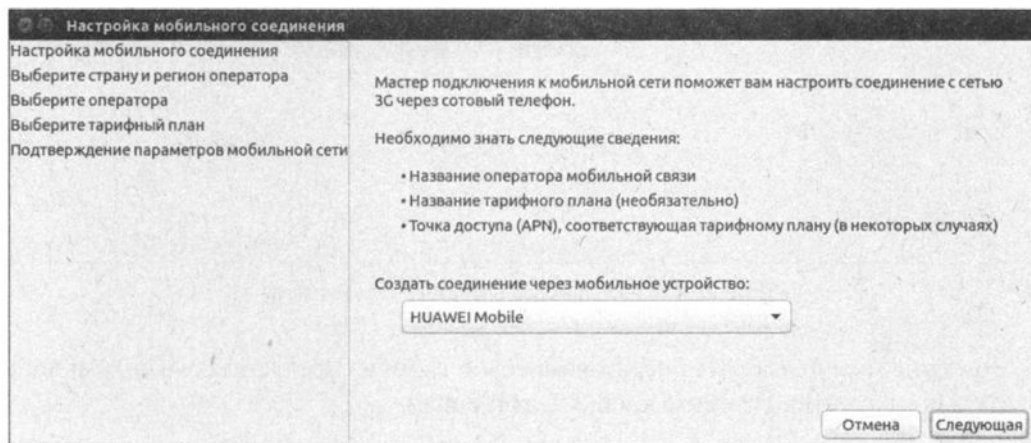


Рис. 5.25. Создание мобильного подключения

В раскрывающемся списке **Создать соединение через мобильное устройство** должно появиться ваше устройство, точнее, марка вашего USB-модема. Если данное название не появилось, следует подождать несколько секунд, пока произойдет инициализация USB-модема. Нажмите кнопку **Следующая**.

На этом этапе настройки предлагается выбрать страну оператора. Так как мы живем в России, то соответственно выбираем этот пункт (рис. 5.26).

Нажмите кнопку **Следующая**, чтобы выбрать самого оператора сотовой связи (рис. 5.27).

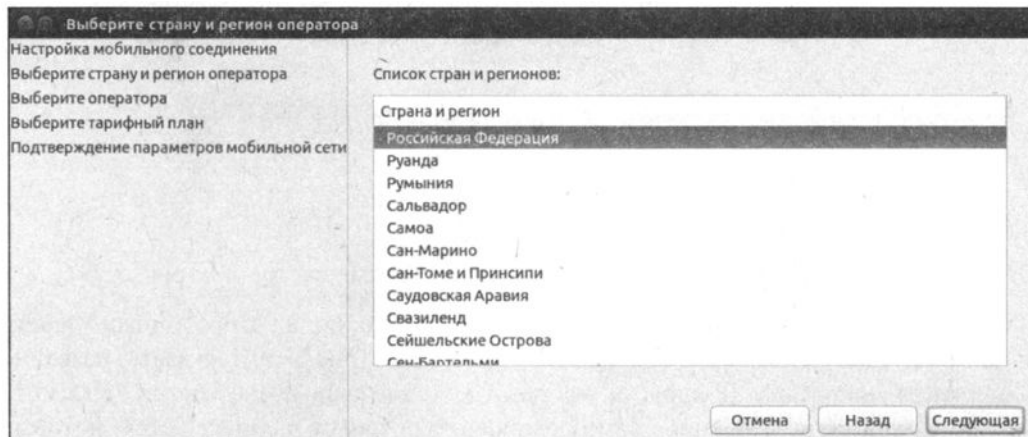


Рис. 5.26. Выбор страны оператора

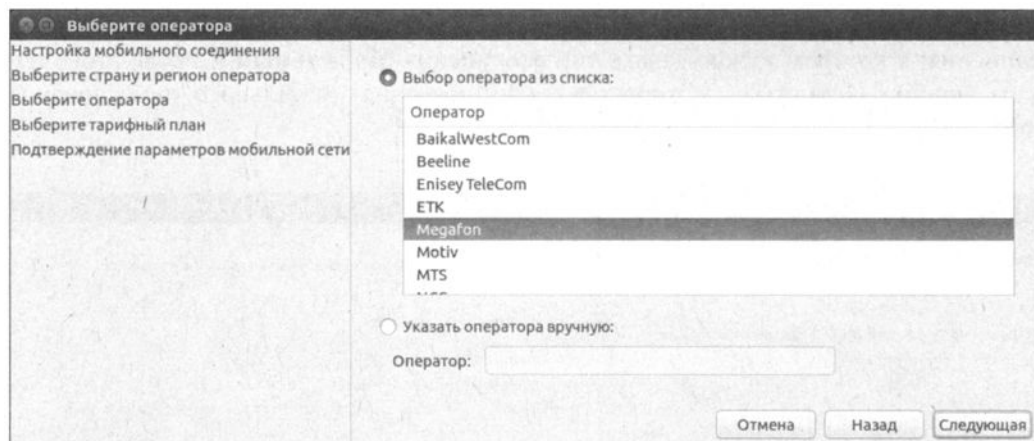


Рис. 5.27. Страница выбора оператора сотовой связи

В этом окне можно выбрать оператора сотовой связи из предлагаемого списка либо указать его вручную. Нажмите кнопку **Следующая**.

Далее будет предложено выбрать тарифный план и точку доступа (APN) (рис. 5.28). Менять тарифный план и точку доступа не имеет смысла, т. к. точка доступа указана верно. Нажмите кнопку **Следующая**.

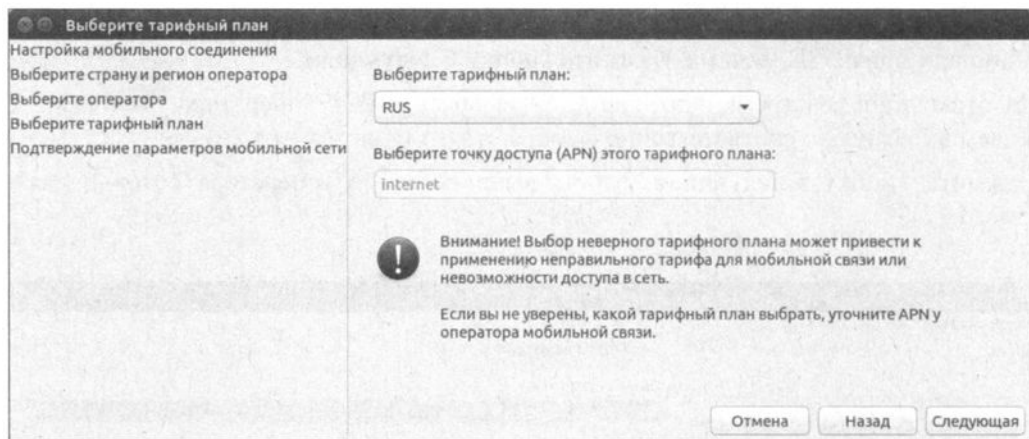


Рис. 5.28. Выбор точки доступа

На следующем этапе предлагается применить сделанные настройки (рис. 5.29).

Нажмите кнопку **Применить**, и откроется последнее окно, в котором можно задать дополнительные настройки мобильного Интернета (рис. 5.30): указать название соединения, настроить IP-адрес и выбрать дополнительные параметры. Нажмите кнопку **Сохранить**. Созданное вами соединение появится в списке сетей, которые отображаются по нажатию на значок программы NetworkManager.

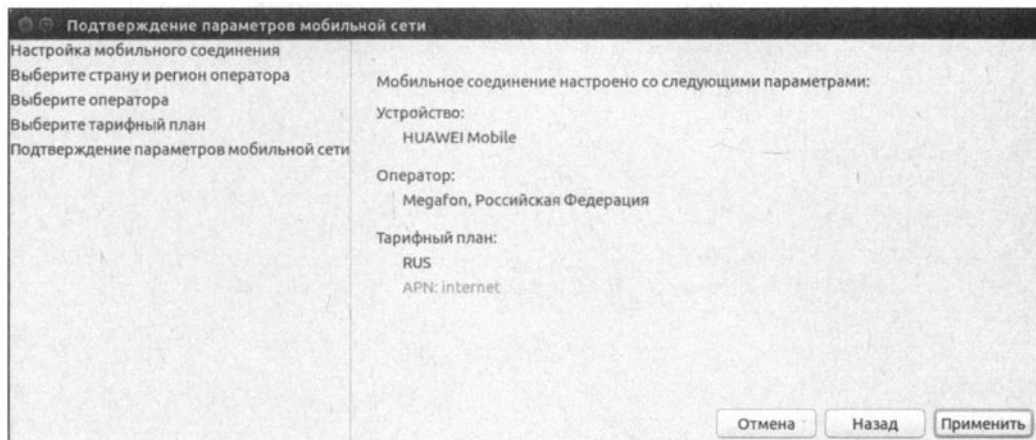


Рис. 5.29. Подтверждение заданных параметров

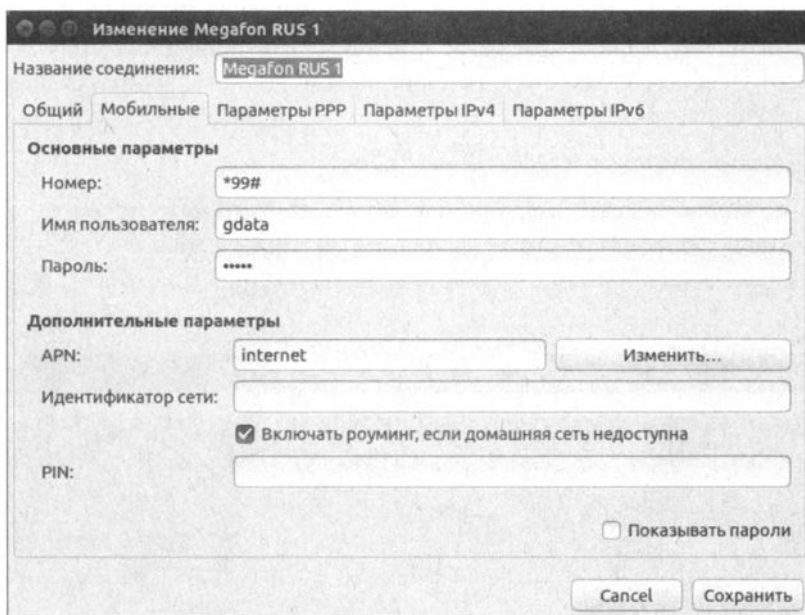


Рис. 5.30. Дополнительные настройки мобильного Интернета

Возможные проблемы с созданием подключений

Бывают такие ситуации, когда не удастся установить соединение с помощью программы NetworkManager. Причина в том, что в файле `/etc/network/interfaces` появляются неоднозначные записи, которые вводят в замешательство программу по установке интернет-соединения. Так, следующее содержание файла `/etc/network/interfaces` не всегда позволяет правильно задействовать DSL-соединение (листинг 5.1).

Листинг 5.1. Содержимое файла /etc/network/interfaces

```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto dsl-provider
iface dsl-provider inet ppp
pre-up /bin/ip link set eno1 up # line maintained by pppoeconf
provider dsl-provider

auto eno1
iface eno1 inet manual
```

Для того чтобы исправить ситуацию, необходимо вручную отредактировать данный файл, но перед этим следует удалить все созданные соединения.

Щелкните по значку индикатора сети, т. е. запустите программу NetworkManager, и по очереди удаляйте каждое соединение. Затем откройте окно Терминала и выполните следующую команду:

```
sudo nano /etc/network/interfaces
```

Вы увидите запрос вашего пароля, после ввода которого появится запрос на редактирование запрашиваемого файла. Нажмите клавишу <Y>, и файл /etc/network/interfaces станет доступен для редактирования (рис. 5.31).

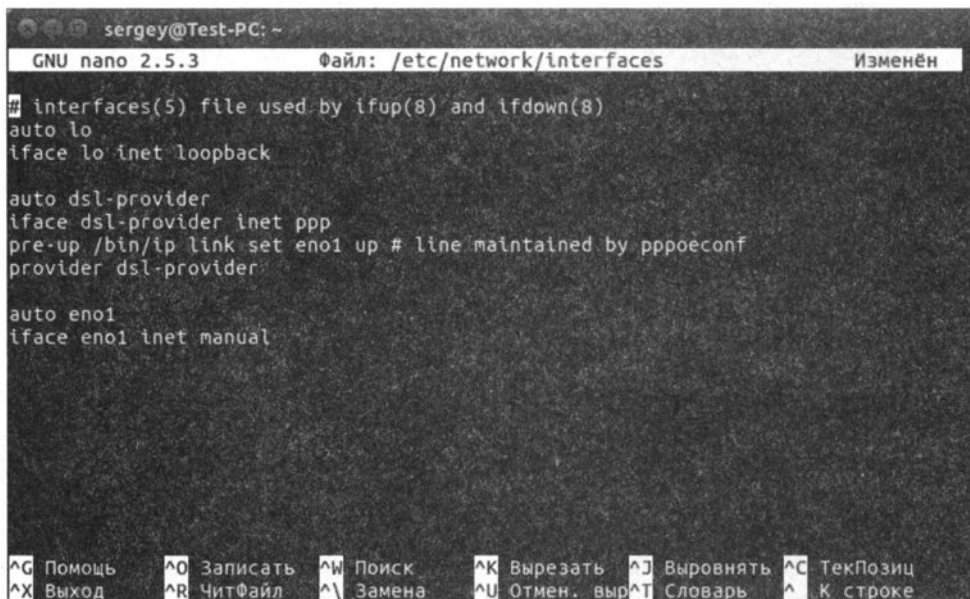


Рис. 5.31. Файл /etc/network/interfaces открыт для редактирования

Нам нужно отредактировать файл `/etc/network/interfaces` таким образом, чтобы в нем остались только первые три строки (листинг 5.2). Все остальные строчки необходимо удалить.

Листинг 5.2. Оставшиеся в файле `/etc/network/interfaces` строки

```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
```

ПРИМЕЧАНИЕ

Для перемещения по файлу используйте клавиши-стрелки, а для редактирования — соответствующие клавиши.

После редактирования необходимо сохранить файл, нажав сочетание клавиш `<Ctrl>+<O>` (либо `<F3>`), а затем клавишу `<Enter>`. После этого закройте Терминал и создайте нужное вам подключение.

Выводы

Мы обсудили наиболее популярные типы подключений к Интернету и рассмотрели способы устранения неполадок в случае их появления. Разумеется, сложно предвидеть каждую конкретную ситуацию и дать какие-то подробные инструкции.

ОС Ubuntu прекрасна тем, что все подключения выполняются довольно легко, а если у вас используются какие-то нестандартные параметры подключения к Интернету, то, зайдя с другого компьютера, можно найти множество информации по интересующей вас теме.

ГЛАВА 6



Установка обновлений и локализация операционной системы

Установка обновлений

ОС Ubuntu, как и все современные операционные системы, имеет встроенный механизм поддержки обновлений. Это означает, что в случае выхода обновлений для какой-либо программной составляющей (программы, утилиты и т. п.) ОС позволит загрузить и применить эти обновления непосредственно из своей среды. При этом не нужно прибегать к переустановке операционной системы. Да, глупо было бы в случае выхода обновлений переустанавливать все целиком.

Обновления нужны в первую очередь для исправления ошибок, которые были допущены в устаревшей версии программного обеспечения. Помимо этого обновления позволяют расширить возможности программного обеспечения за счет добавления новых функций.

Таким образом, в ОС Ubuntu есть механизм, который управляет всем этим процессом. Называется он Менеджером обновлений. Давайте запустим его, не прибегая к Терминалу. Для этого нажмите на значок главного меню в панели запуска и в строке поиска введите **обновление**. В ответ на это в результатах поиска появится программа **Обновление приложений** (рис. 6.1).

Запустите программу **Обновление приложений**. При этом на несколько секунд появится окно поиска обновлений, которое в случае успешного поиска сменится следующим окном (рис. 6.2).

Такие обновления могут содержать обновления безопасности для программ, которые имеют доступ к Интернету. Обновления могут затрагивать и другие программы, которые для своей работы не требуют доступа к Сети. Для того чтобы просмотреть перечень предлагаемых обновлений, разверните список **Сведения об обновлениях**, и вы сможете отметить те пункты, обновления для которых вы отказываетесь загружать. Нажмите кнопку **Установить сейчас** для установки предлагаемых обновлений.

На запрос аутентификации введите свой пароль и нажмите кнопку **Аутентифицировать** (рис. 6.3).

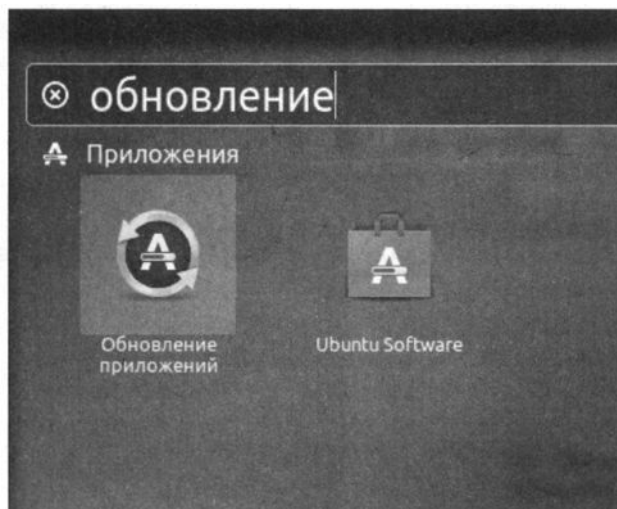


Рис. 6.1. Поиск программы Обновление приложений

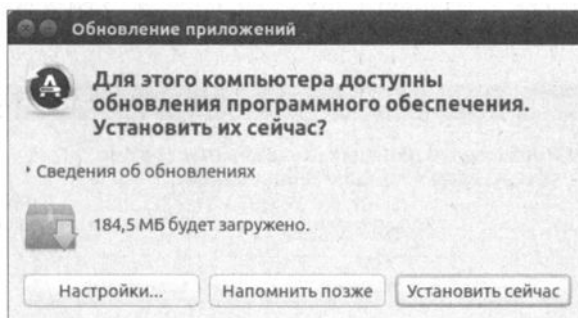


Рис. 6.2. Программа Обновление приложений сообщила о готовности обновлений к загрузке

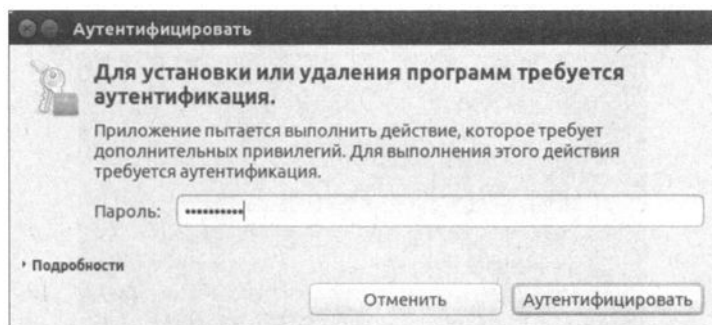


Рис. 6.3. Запрос ввода пароля

ПРИМЕЧАНИЕ

Для загрузки обновлений требуется подключение к Интернету. Учтите, что обновления могут иметь довольно большой размер, загружать которые при наличии у вас лимитного тарифного плана — несколько дорогое удовольствие.

Начнется процесс загрузки и установки обновлений (рис. 6.4).

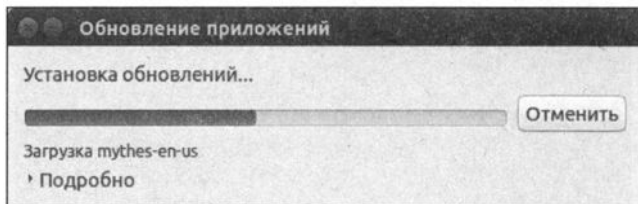


Рис. 6.4. Процесс загрузки обновлений

В разворачивающемся списке **Подробнее** вы можете наблюдать за процессом обновления.

В зависимости от количества устанавливаемых обновлений процесс обновления может занять некоторое время. Дождитесь окончания процесса и нажмите кнопку **Перезагрузить сейчас...** (рис. 6.5).

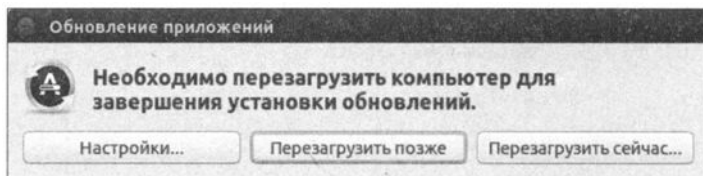


Рис. 6.5. Установка обновлений завершена

Ваш компьютер будет перезагружен, и после загрузки ОС вы сможете продолжить работу с ОС Ubuntu.

Каждый раз вручную проверять обновления совсем необязательно, т. к. ОС Ubuntu вам сообщит о выходе обновлений (рис. 6.6).

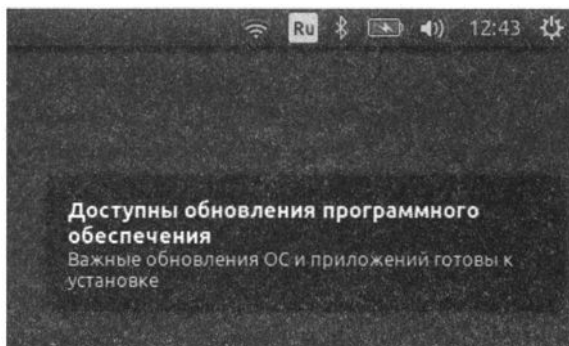


Рис. 6.6. Доступны обновления программного обеспечения

После этого нужно последовательно выполнить в Терминале следующие команды:

```
sudo apt update  
sudo apt-get upgrade
```

При вводе первой команды появится запрос на ввод вашего пароля, а при выполнении второй команды от вас потребуется подтверждение ваших намерений. Введите `y` и нажмите клавишу `<Enter>`. Обновления будут установлены.

ПРИМЕЧАНИЕ

Первая команда позволяет получить информацию об обновлениях, т. е. узнать, появились ли новые пакеты в репозиториях. Вторая команда дает возможность скачать и установить эти пакеты.

Установка пакетов локализации

После установки ОС Ubuntu возникает необходимость полностью локализовать операционную систему на родной язык. Разработчики дистрибутива Ubuntu не имеют возможности включить в загрузочный дистрибутив все языки, да это и не нужно, потому что всегда имеется возможность загрузить локализованные пакеты из Интернета.

Для того чтобы загрузить языковые пакеты, выполните следующие действия. Нажмите на значок системного меню (шестеренка с тумблером) и выберите пункт **Параметры системы....** Откроется одноименное окно (рис. 6.7).

Данное окно напоминает собой Панель управления, которую мы привыкли видеть в ОС Windows. В этом окне нас интересует раздел **Персональные**, в котором необходимо найти утилиту **Язык системы**. Щелкните по ее значку. Откроется окно языковых настроек, и начнется поиск доступных пакетов к установке. В случае если поддержка языков установлена не полностью, система сообщит об этом (рис. 6.8).

Нажмите кнопку **Установить** и на запрос аутентификации введите свой пароль. Начнется процесс установки требуемых языковых пакетов. Дождитесь окончания процесса и нажмите кнопку **Применить для всей системы**, а затем введите свой пароль для подтверждения изменений (рис. 6.9).

ПРИМЕЧАНИЕ

Если первым в списке стоит английский язык, то соответственно интерфейс всей ОС будет на английском языке. Переместите русский язык на первое место в списке и нажмите кнопку **Применить для всей системы**.

Кнопка **Установка и удаление языков...** позволяет установить дополнительные языки. В большинстве случаев это не требуется, если вы не используете язык, отличный от русского и английского.

Для закрытия окна **Язык системы** нажмите кнопку **Закрыть**.

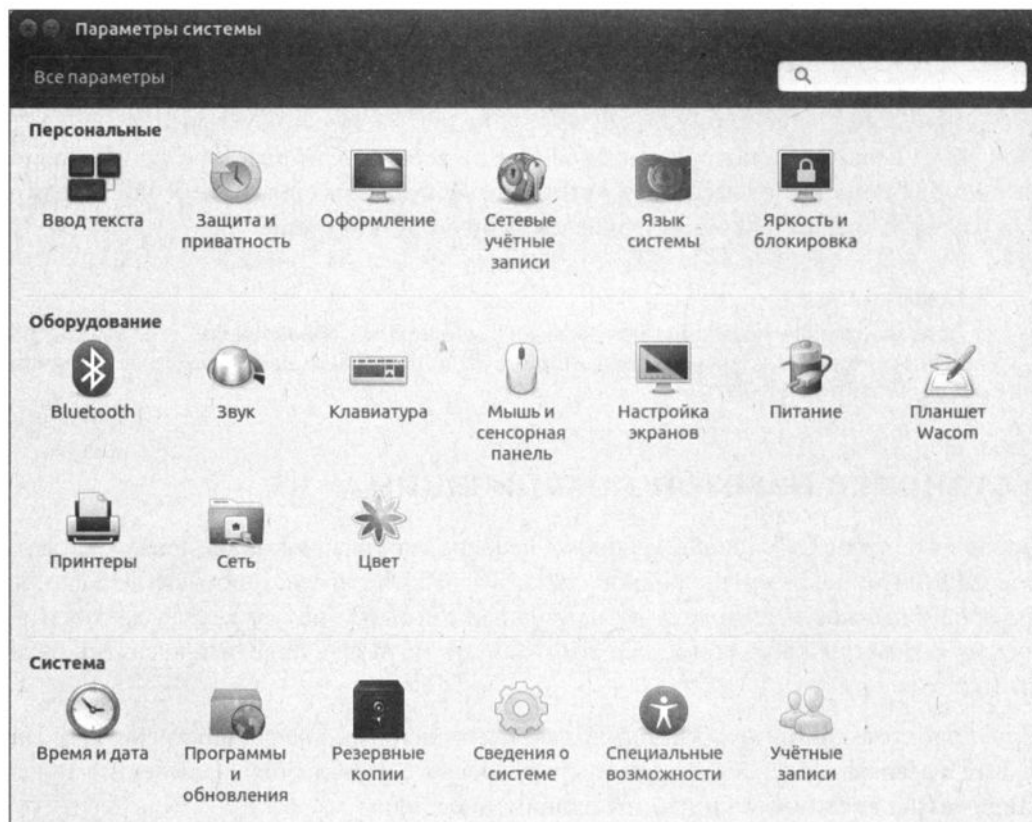


Рис. 6.7. Окно параметров системы

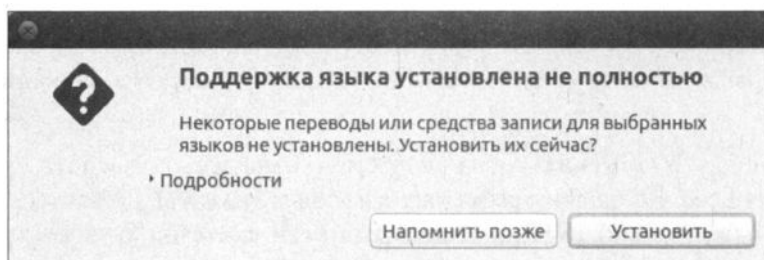


Рис. 6.8. Требуется установка языковых пакетов

Если вы при установке ОС Ubuntu выбрали английский язык интерфейса, а затем в окне **Язык системы** решили сменить на русский, то для окончательного применения изменений необходимо завершить текущий сеанс, а затем снова зайти под своей учетной записью. Для этого нажмите на значок системного меню (шестеренка с тумблером) и выберите пункт **Завершение сеанса...** После входа в систему интерфейс ОС уже будет на русском языке.

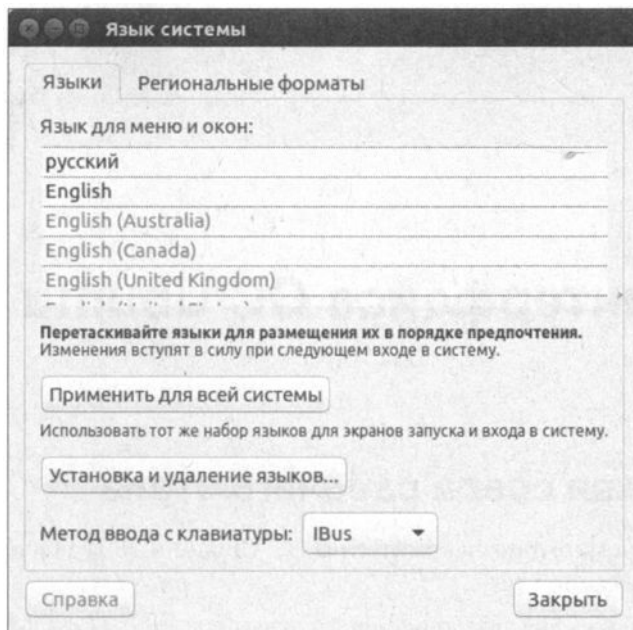


Рис. 6.9. Установка языковых пакетов завершена

Выводы

Мы научились устанавливать обновления для операционной системы и загружать языковые пакеты, которые необходимы нам для нормальной работы в ОС Ubuntu Linux.

Пользуясь полученными знаниями, вы теперь сможете самостоятельно менять язык интерфейса операционной системы, а также устанавливать необходимые обновления, которые будут выходить на протяжении всей поддержки используемой вами версии операционной системы Ubuntu Linux.

ГЛАВА 7



Обзор интерфейса ОС Ubuntu

Графическая среда рабочего стола

Мы уже успели немного познакомиться с ОС Ubuntu и ее графической оболочкой, которая носит название Unity.

Unity — это графическая оболочка среды рабочего стола GNOME, которая была специально разработана для ОС Ubuntu. Все что вы видите при загрузке ОС Ubuntu — это и есть графическая оболочка Unity. По сути, графическую оболочку Unity можно сравнить с Проводником в ОС Windows.

Помимо графической оболочки Unity, в ОС на базе Linux используются такие оболочки, как GNOME, KDE, Xfce и LXDE. Unity является своеобразной надстройкой над графической оболочкой GNOME, т. е. оболочкой GNOME.

Так, для графической среды GNOME можно использовать любую графическую оболочку, которая совместима с GNOME, например Cinnamon, либо оставить графическую среду GNOME Shell без дополнительных надстроек типа Unity и Cinnamon.

По умолчанию в ОС Ubuntu используется графическая оболочка Unity, поэтому рассматривать мы будем именно ее.

Рабочий стол в Unity

После ввода пароля при запуске ОС Ubuntu загружается уже знакомый нам *рабочий стол* (рис. 7.1).

Рабочий стол графической оболочки Unity можно мысленно разделить на три основные части:

- ◆ панель меню;
- ◆ панель запуска;
- ◆ рабочее место.

Сейчас мы более подробно рассмотрим эти три составляющие и узнаем, как с ними работать.

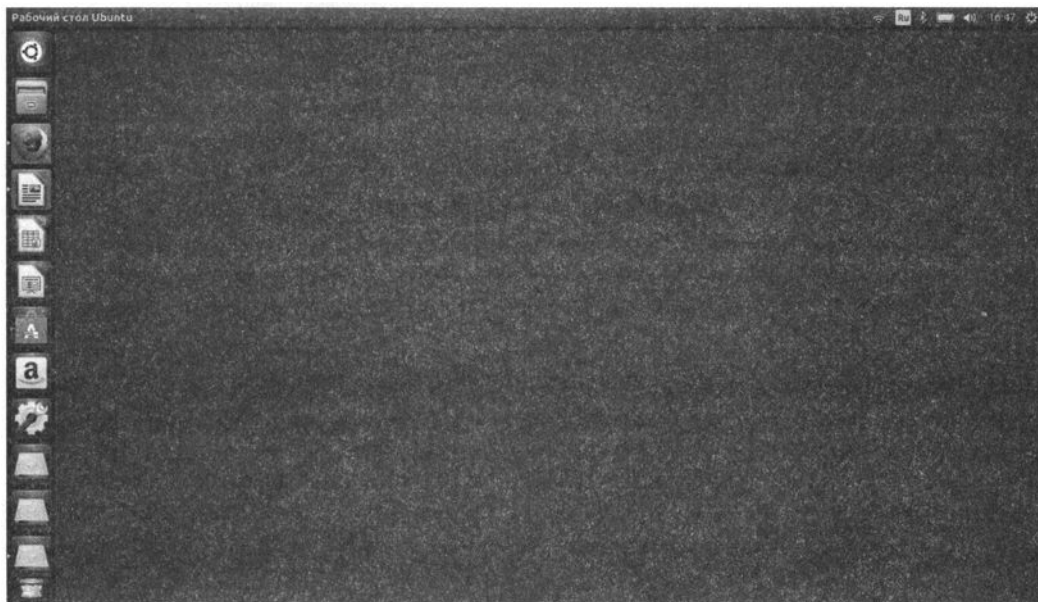


Рис. 7.1. Рабочий стол в графической оболочке Unity

Панель меню

Панель меню в ОС Ubuntu — это уже знакомая нам темная полоска вверху экрана, которая похожа на Панель задач в ОС Windows (рис. 7.2).



Рис. 7.2. Панель меню в графической оболочке Unity

Однако панель меню устроена немного по-иному и на первых порах использования кажется неудобной и непонятной. Давайте внесем ясность в то, как она работает в графической оболочке Unity.

Левая часть панели меню задействовано для нужд программы, которая запущена в тот или иной момент времени. Это означает, что в левой части меню, там, где мы привыкли видеть надпись "Рабочий стол Ubuntu", во время работы с какой-либо программой будет располагаться панель меню самой программы. Немного запутанно звучит, не так ли? Сейчас все встанет на свои места.

В качестве примера давайте запустим любую программу, например браузер Mozilla Firefox. Теперь панель меню будет выглядеть так, как представлено на рис. 7.3.



Рис. 7.3. Заголовок активного приложения

Надпись "Рабочий стол Ubuntu" сменилась надписью "Ubuntu Start Page - Mozilla Firefox", но и это еще не все. Если навести курсор мыши на панель меню, то она примет другой вид (рис. 7.4).

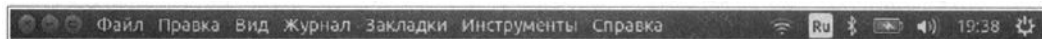


Рис. 7.4. Меню приложения

Теперь у нас появились кнопки управления окном (заккрыть, свернуть и развернуть окно — слева), а также пункты меню запущенной программы. Немного непривычно, но это было сделано с целью экономии рабочего пространства экрана.

Кнопки управления окном выглядят несколько по-другому, чем те, к которым мы привыкли в ОС Windows. Так, первой идет кнопка *закрытия окна*, затем кнопка *сворачивания окна* и только потом кнопка *разворачивания окна*.








В правой части панели меню расположилось *меню состояния* (рис. 7.5).



Рис. 7.5. Меню состояния

Аналог меню состояния мы привыкли наблюдать в ОС Windows, только там она называется *областью уведомлений* (системный трей).

В меню состояния у нас имеется семь индикаторов:

- ♦  — индикатор сети, это значок уже знакомой нам программы Network-Manager. С его помощью мы можем подключаться в Интернету, а также создавать и изменять соединения;
- ♦  — индикатор раскладки клавиатуры, который позволяет быстро ее сменить;
- ♦  — меню Bluetooth, с помощью которого вы можете устанавливать соединения с устройствами, поддерживающими эту технологию;
- ♦  — индикатор батареи, который присутствует на портативных устройствах, таких как ноутбук или планшет, и показывает текущее состояние батареи;
- ♦  — индикатор звукового меню, который предоставляет быстрый доступ к управлению громкостью звука, а также к управлению медиапроигрывателем;
- ♦  — индикатор часов для быстрого доступа к настройкам времени, а также календарю;
- ♦  — системное меню для доступа к управлению компьютером, например, к выключению, перезагрузке и настройкам системы.

Это далеко не весь список индикаторов, т. к. их количество изменяется по мере установки программ, а их состояние (вид) — в зависимости от состояния того или

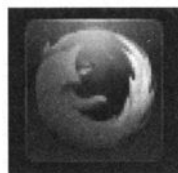
инного индикатора. Не зря они называются индикаторами и всегда находятся под пристальным взглядом пользователя.

Панель запуска

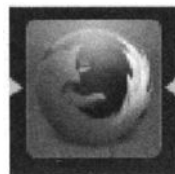
Панель запуска — это еще одна особенность графической оболочки Unity. Она представляет собой вертикальную панель, которая появляется сразу же после входа в ОС Ubuntu (рис. 7.6).



Рис. 7.6. Панель запуска в графической среде Unity



а



б

Рис. 7.7. Вид значка приложения до запуска (а) и после него (б)

На панели запуска расположены ярлыки для запуска приложений, например, браузера Mozilla Firefox, файлового менеджера Nautilus, Корзины и т. п.

Давайте для примера запустим какое-нибудь приложение, пусть это будет Mozilla Firefox. Обратите внимание, что значок браузера примет другой вид (рис. 7.7).

Как видите, после запуска приложения его значок изменил цвет, а слева и справа рамки значка появились маленькие белые треугольники. Левый треугольник означает, что запущена одна копия (окно) приложения, два — две копии приложения и т. д. Правый треугольник сообщает, что данное приложение в настоящий момент активно.

ПРИМЕЧАНИЕ

Когда значок приложения "пульсирует" после того, как вы на него нажали, то это означает, что в данный момент времени выполняется его запуск. Происходит это потому, что некоторые приложения для своего запуска требуют загрузки определенного программного кода, размер которого у всех приложений разный.

Поэкспериментируйте с запуском приложений, и вы скоро привыкнете к такому поведению графической оболочки Unity.

Однако это не весь набор спецэффектов значков приложений. Так, на некоторых значках программ могут появляться индикаторы хода выполнения определенной задачи, после выполнения которой значок начинает покачиваться и накаляться

(в случае если вы в этот момент работаете в другой программе), а левый белый треугольник становится голубым. Это означает, что текущая задача была выполнена и программа требует вашего внимания.

На некоторых значках могут появляться числа, но это в основном касается программ по обмену сообщениями. Таким способом программа сигнализирует вам о том, что имеются непрочитанные сообщения и т. п.

Наверняка у вас возник вопрос: как же поместить на панель запуска значок той или иной программы? Очень легко. Для этого запустите нужную программу, и ее значок появится на панели запуска. После этого необходимо щелкнуть правой кнопкой мыши по значку приложения и выбрать команду **Закрепить на панели** (рис. 7.8).

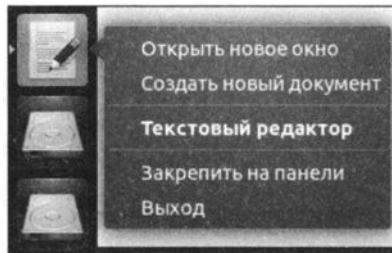


Рис. 7.8. Закрепление значка на панели запуска

Открепляется значок по пункту меню **Изъять из панели**.

ПРИМЕЧАНИЕ

Количество пунктов меню может отличаться в зависимости от функциональных возможностей той или иной программы.

Расположить значки программ в нужном вам порядке можно простым перетаскиванием. Как видите, ничего сложного, хоть и непривычно на первый взгляд.

Рабочее место

Рабочее место в ОС Ubuntu — это все оставшееся место, между панелью меню и панелью запуска. Другими словами, большая часть экрана и есть рабочее место.

По умолчанию в ОС Ubuntu рабочее место только одно, однако имеется возможность включить сразу четыре рабочих места.

ПРИМЕЧАНИЕ

Количество рабочих мест можно увеличить с помощью программы Ubuntu Tweak, но сейчас пойдет речь о стандартных средствах запуска рабочих мест.

Рабочие места содержат одни и те же панель меню и панель запуска. В общем виде рабочие места рассматривать как несколько мониторов, на которые можно переключаться, и которые содержат различные окна и запущенные приложения.

Итак, нажмите кнопку системного меню и выберите пункт **Параметры системы...**, далее запустите утилиту **Оформление**, в ее окне перейдите на вкладку **Режим** и установите флажок **Задействовать рабочие места** (рис. 7.9).

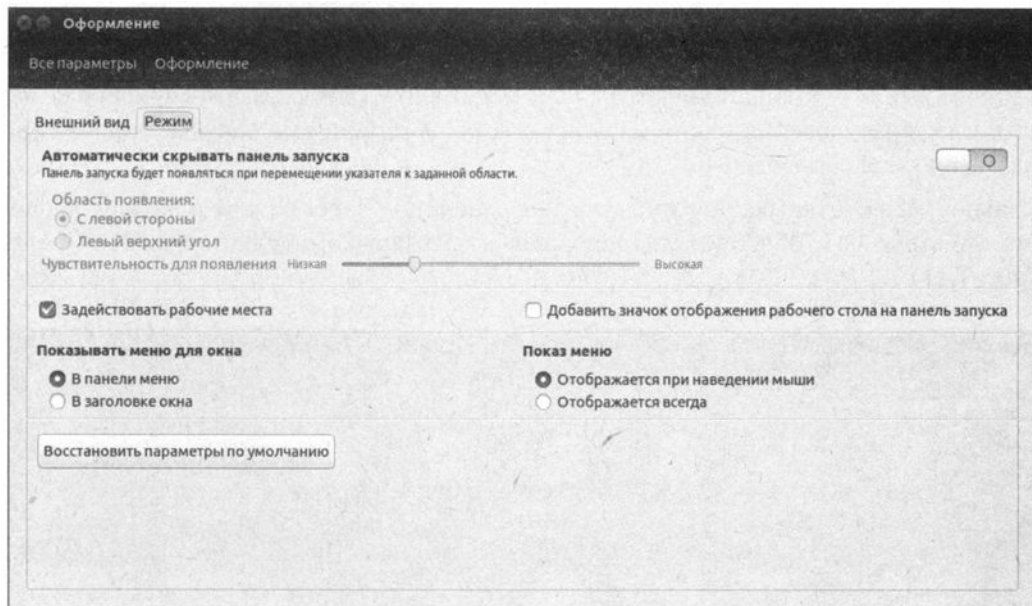


Рис. 7.9. Включение режима рабочих мест

После этого на панели запуска появится значок рабочих мест (рис. 7.10).

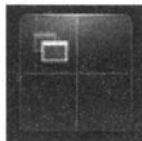


Рис. 7.10. Значок рабочих мест на панели запуска

Для того чтобы переключиться на другое рабочее место, достаточно щелкнуть мышью по соответствующему значку и выбрать нужное рабочее место. То же самое действие можно выполнить, нажав сочетание клавиш `<Ctrl>+<Alt>+<стрелки курсора клавиатуры>`.

Для того чтобы перевести одну рабочую область в другой экран, нажмите `<Ctrl>+<Shift>+<Alt>+<стрелки курсора>`, и активное окно (рабочее место) будет перенесено на другую рабочую область.

Функция рабочих мест полезна тогда, когда вы хотите "разгрузить" рабочее пространство на вашем компьютере и переместить некоторые окна на другой монитор или рабочую область.

ПРИМЕЧАНИЕ

При перемещении рабочей области в другую рабочую область белые треугольники возле значков программ примут вид закрывающегося тега ">". Это значит, что данная программа находится в другой рабочей области.

Главное меню

Главное меню — кнопка запуска панели меню Unity (англ. Dash — приборная панель), которая является одним из основных средств поиска и запуска приложений в графической оболочке Unity.

Главное меню открывается по нажатию клавиши <Super> (знакомая нам клавиша <Win> в ОС Windows) либо по щелчку на значке с логотипом ОС Ubuntu (рис. 7.11).

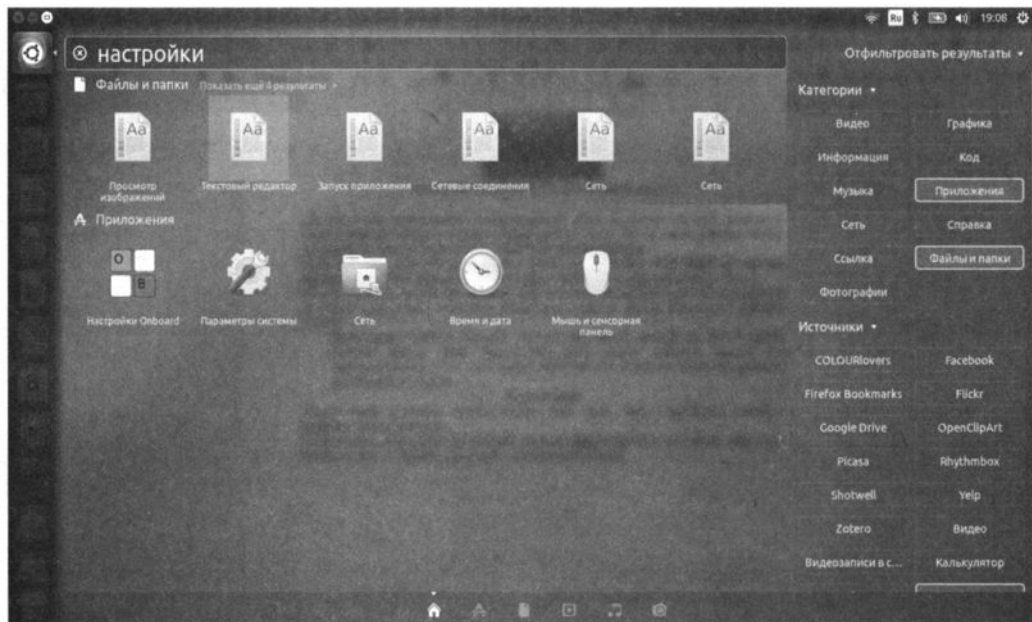


Рис. 7.11. Главное меню

Главное меню может вам напомнить меню **Пуск** из ОС Windows, однако здесь оно несет несколько иной функционал, хотя какие-то общие черты и прослеживаются. Не зря же придуман графический интерфейс, чтобы удобнее было работать с операционной системой.

Давайте более детально рассмотрим главное меню.

По нажатию кнопки главного меню открывается полупрозрачное окно со строкой поиска и дополнительными параметрами запроса.

Вверху окна располагается строка поиска, а справа — панель фильтров, т. е. дополнительных параметров, которые вы можете задать для уточнения поиска.

После того, как вы начали в строке поиска вводить слово, вам моментально предлагаются результаты поиска. К примеру, у нас в строке поиска введено слово *настройки* и представлены результаты запроса. В правой части главного меню мы оставили все как есть, т. к. результат запроса получился маленький, и нам не потребовалось его уточнять.

Ну и наконец, в самой нижней части главного меню находятся так называемые *линзы*, которые позволяют группировать поисковые результаты по следующим категориям:

- ♦ *домашняя линза* — представляет широкий диапазон поиска: приложения, документы, музыка, видео и т. п.;
- ♦ *линза поиска недавних приложений* — результаты поиска из недавно использовавшихся приложений;
- ♦ *линза поиска файлов и папок* — позволяет отсортировать результаты и ограничить поисковую выдачу лишь папками и документами;
- ♦ *линза поиска видео* — в результатах дает возможность ограничить выдачу лишь видеофайлами;
- ♦ *линза поиска музыкальных композиций* — результат поиска только из музыкальных файлов;
- ♦ *линза поиска фотографий* — ограничивает результат лишь файлами графических форматов.

Как вы уже поняли, линзы призваны сузить поиск в соответствии с типом файлов. Переключаться между линзами можно кнопкой мыши либо с помощью сочетания клавиш <Ctrl>+<Tab>.

Меню HUD

HUD (Head-Up Display — индикатор на лобовом стекле) — меню поиска команды в открытом приложении.

Представим такую ситуацию: вы набираете текст в программе LibreOffice Writer и забываете, где находится меню вставки рисунка. Искать по всем пунктам меню программы не хочется. Как быть в такой ситуации?

Разработчики графической оболочки Unity предусмотрели такой вариант и реализовали поиск для команд меню. Для этого следует нажать клавишу <Alt> и в строке поиска ввести примерное название команды (рис. 7.12).

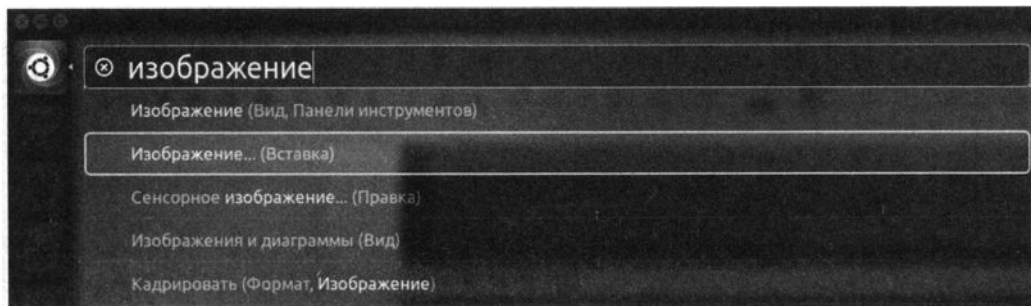


Рис. 7.12. Меню HUD предлагает найденные команды

В предложенных вариантах и будет находиться искомая команда. Вам останется только выбрать ее и продолжить свою работу, а HUD при каждом вашем обращении к нему будет предлагать наиболее подходящие для вас варианты.

Выводы

Мы познакомились с графической средой рабочего стола под названием Unity. Мы узнали, что для ОС Ubuntu существует несколько графических оболочек, а Unity является стандартной и самой современной, т. к. она адаптирована под мобильные устройства, для которых свободное пространство рабочего стола является наиболее приоритетной задачей.

Мы научились работать с панелью запуска, а также с панелью меню. Мы настроили рабочие места, которые зачастую помогают наиболее продуктивно организовать нашу работу, и познакомились с главным меню, успев поработать с меню HUD, о которых мы не знали до знакомства с ОС Ubuntu.

ГЛАВА 8



Настройка операционной системы и решение проблем

Параметры системы

Прежде чем переходить непосредственно к рассмотрению организации файловой системы, необходимо вкратце познакомиться с основными настройками операционной системы Ubuntu, к которым вам придется прибегать на протяжении всей работы с этой ОС.

Итак, окно настройки параметров системы можно вызвать следующими способами:

- ◆ нажать кнопку главного меню, в строке поиска ввести **Параметры системы** и после этого открыть соответствующее окно настройки системы;
- ◆ нажать на значок **Параметры системы**, который находится на панели запуска;
- ◆ нажать на значок системного меню и выбрать пункт **Параметры системы** для открытия соответствующего окна;
- ◆ с помощью команды `unity-control-center`, которую нужно выполнить в Терминале.

Мы воспользуемся четвертым вариантом, т. к. он является наиболее "линуксовым". Для этого в Терминале выполните следующую команду:

```
unity-control-center
```

Откроется окно **Параметры системы** (рис. 8.1).

Параметры системы напоминают **Панель управления** в ОС Windows. Действительно, идеология одна и та же.

В этом окне вы найдете три категории настроек:

- ◆ **Персональные** — касаются настроек конкретного пользователя в системе;
- ◆ **Оборудование** — позволяют настроить работу аппаратного обеспечения вашего компьютера;
- ◆ **Система** — настройки, позволяющие управлять самой системой.

Рассматривать каждую настройку в отдельности не имеет смысла, т. к. все зависит от личных предпочтений и нужд. Мы лишь затронем те настройки, которые порой необходимо изменить для более комфортной работы с ОС Ubuntu.

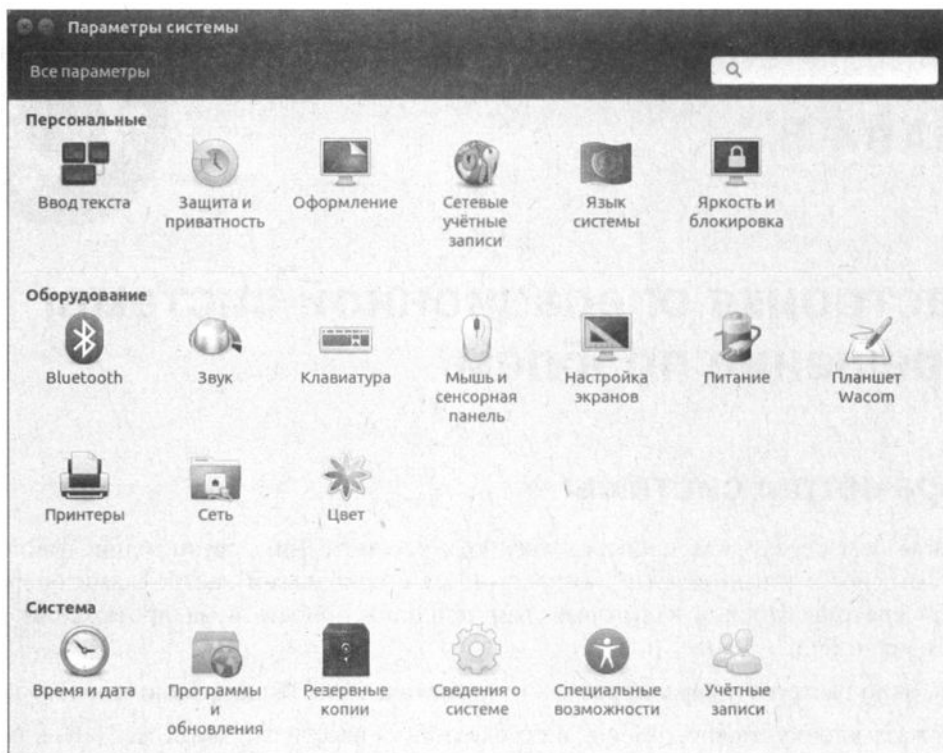


Рис. 8.1. Окно Параметры системы

Настройка раскладки клавиатуры

По умолчанию в ОС Ubuntu смена раскладки клавиатуры выполняется сочетанием клавиш **<Super>+<Пробел>**.

ПРИМЕЧАНИЕ

Клавиша **<Super>** — это привычная нам клавиша **<Win>** с логотипом этой операционной системы.

Согласитесь, поначалу трудно привыкнуть к другой раскладке клавиатуры и по привычке хочется нажать **<Shift>+<Alt>** для смены раскладки. Так как мы перешли в ОС Ubuntu Linux из мира ОС Windows, то всегда хочется использовать именно эти клавиши. В ОС Ubuntu по умолчанию назначено сочетание клавиш **<Super>+<Пробел>**, вместо **<Shift>+<Alt>**, но мы вправе поменять раскладку на свой вкус и усмотрение.

Давайте займемся сменой "горячих" клавиш раскладки клавиатуры. Откройте окно **Параметры системы**, если вы закрыли его, и в группе **Персональные** щелкните по значку **Ввод текста** (рис. 8.2).

Обратите внимание, что в разделе используемых источников ввода у нас указаны две раскладки: английская и русская.

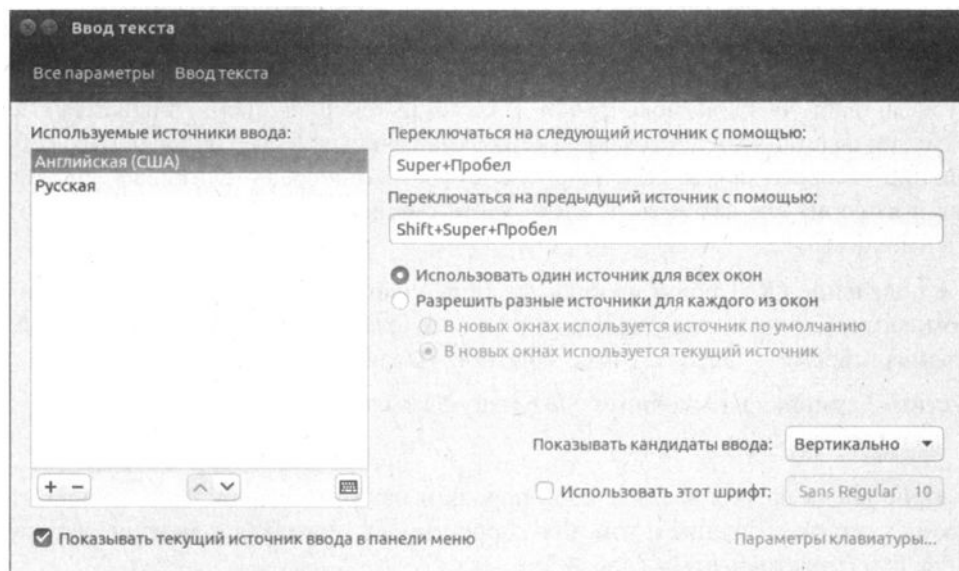



Рис. 8.2. Окно настроек "горячих" клавиш раскладки клавиатуры

Если вы хотите, чтобы при включении компьютера была активной именно русская раскладка клавиатуры, то выделите эту раскладку в списке источников ввода и нажмите кнопку перемещения .

Для смены "горячих" клавиш переключения раскладок клавиатуры щелкните в поле ввода **Переключиться на следующий источник с помощью** и нажмите сочетание клавиш **<Shift>+<Alt>** либо другое привычное вам сочетание.

Если у вас установлено несколько языков (три и более), то для удобства целесообразно назначить клавиши переключения раскладок клавиатуры на предыдущую в поле ввода **Переключаться на предыдущий источник с помощью**.

После этого закройте это окно, и все заданные настройки будут применены.

Более детальную настройку клавиатуры можно произвести после щелчка по ссылке **Параметры клавиатуры** в правом нижнем углу окна и задать "горячие" клавиши для поведения операционной системы при соответствующих нажатиях.

Смена системного времени

Наверняка вы уже заметили, что при работе в ОС Ubuntu и ОС Windows системное время различное. Например, вы работаете ОС Ubuntu, и вам понадобилось поработать в ОС Windows. Вы перезагружаете компьютер и после загрузки операционной системы замечаете, что системное время в ОС Windows странным образом поменялось. Если вы заглянете в меню BIOS, то увидите ту же картину. Связано это с тем, что ОС Ubuntu считает время по UTC (англ. Coordinated Universal Time — всемирное координированное время). Это означает, что при настройке времени операционная система Ubuntu выставляла время UTC, а при выводе добавляла к нему сме-

шение относительно текущего часового пояса, который вы выбрали при установке ОС.

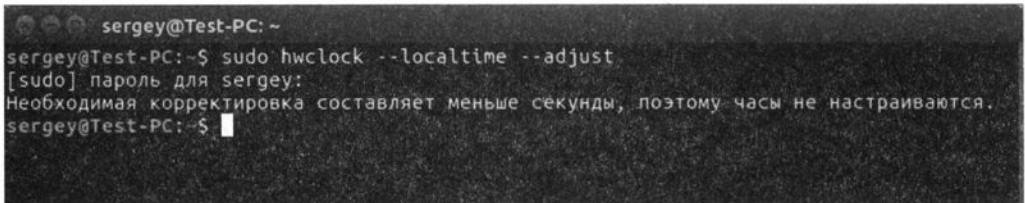
Допустим, ваш часовой пояс равен UTC+6 и текущее время равняется 19:42. ОС Ubuntu выставляет системное время относительно UTC и добавляет к нему смещение часового пояса. ОС Windows, в свою же очередь, считывает системное время и выводит его как есть, т. е. без учета смещения относительно часового пояса, а именно 13:42.

Такое поведение ОС Ubuntu удобно, но лишь если она является единственной ОС на компьютере. А если она установлена по соседству с ОС Windows, то каждый раз настраивать время — дело не самое занятое. Давайте исправим эту ситуацию.

Запустите Терминал и выполните следующую команду:

```
sudo hwclock --localtime --adjust
```

На запрос ввода пароля введите свой пароль и нажмите клавишу <Enter>. В ответ на это появится сообщение о том, что корректировка составляет меньше секунды и часы не будут настраиваться (рис. 8.3).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ sudo hwclock --localtime --adjust  
[sudo] пароль для sergey:  
Необходимая корректировка составляет меньше секунды, поэтому часы не настраиваются.  
sergey@Test-PC:~$
```

Рис. 8.3. Отмена корректировки времени по UTC

Не пугайтесь этого сообщения, оно означает, что ОС не будет больше выставлять время в соответствии с UTC.

ПРИМЕЧАНИЕ

Для того чтобы вернуть корректировку времени по UTC, выполните следующую команду: `sudo hwclock --utc -adjust`. В ответ на это появится то же самое сообщение, что мы могли видеть при выполнении первой команды.

После того как вы отменили корректировку времени, при следующей загрузке ОС Windows следует настроить системное время, и уже при последующих загрузках операционной системы время не будет сбиваться, а останется таким, каким вы его задали прежде.

В ранних версиях ОС Ubuntu корректировка времени отключалась в конфигурационном файле `/etc/default/rcS`. При этом необходимо было исправить директиву с `UTC=yes` на `UTC=no`.

Таким образом устраняются неприятности с корректировкой времени.

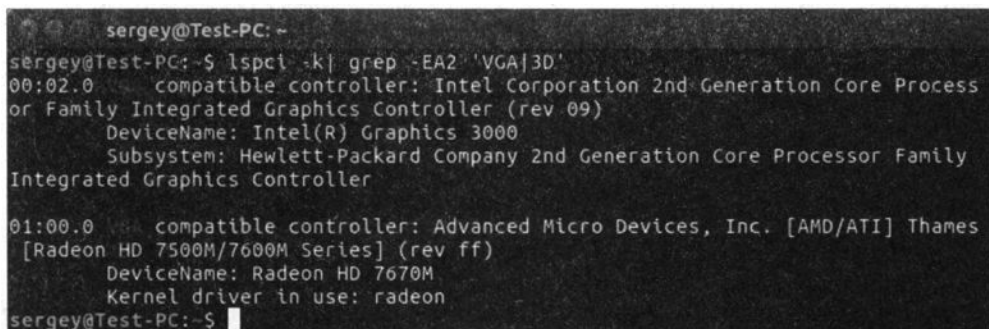
Настройка спящего режима

Нередки случаи, когда необходимо воспользоваться ждущим режимом. Однако в силу особенностей ОС Ubuntu ждущий режим не всегда работает должным образом. Так, на ноутбуках, оснащенных гибридной графикой с дискретной видеокартой AMD/ATI, ждущий режим отказывается работать. ОС Ubuntu просто не желает уходить в ждущий режим и продолжает работать. Для того чтобы выйти из режима, при котором компьютер как бы "зависает", нужно принудительно завершить работу компьютера долгим нажатием кнопки выключения питания. Эта проблема существует уже довольно много лет, и на просторах Интернета можно встретить разные советы, вплоть до обновления версии ядра операционной системы, однако суть проблемы кроется в неустановленных драйверах видеокарты. Операционная система не может корректно завершить работу с гибридной графикой, и компьютер оказывается в полурбочем состоянии: экран погашен, но операционная система продолжает работать. Сейчас мы попробуем решить проблему со ждущим режимом, и этот способ будет работоспособен в большинстве случаев.

Для начала необходимо определить используемый драйвер для видеокарты. Выполните в Терминале следующую команду:

```
lspci -k| grep -EA2 'VGA|3D'
```

В ответ на это будет выведена информация о видеокарте (рис. 8.4).



```
sergey@Test-PC: ~$ lspci -k| grep -EA2 'VGA|3D'
00:02.0 compatible controller: Intel Corporation 2nd Generation Core Process
or Family Integrated Graphics Controller (rev 09)
    DeviceName: Intel(R) Graphics 3000
    Subsystem: Hewlett-Packard Company 2nd Generation Core Processor Family
Integrated Graphics Controller

01:00.0 compatible controller: Advanced Micro Devices, Inc. [AMD/ATI] Thames
[Radeon HD 7500M/7600M Series] (rev ff)
    DeviceName: Radeon HD 7670M
    Kernel driver in use: radeon
sergey@Test-PC: ~$
```

Рис. 8.4. Информация о видеокарте

Последняя строка "Kernel driver in use: radeon" говорит о том, что используется стандартный драйвер ядра. Это означает, что в ОС не установлено никаких дополнительных видеодрайверов.

Команда `lspci -k| grep -EA2 'VGA|3D'` позволила узнать модель и производителя видеокарты. Это нам необходимо для установки нужного программного обеспечения, т. е. видеодрайвера.

Теперь, когда мы знаем, что у нас имеется видеокарта AMD/ATI, нам нужно установить соответствующий драйвер. Не закрывайте Терминал и выполните следующую команду (их будет три):

```
sudo add-apt-repository ppa:oibaf/graphics-drivers
```


Эта команда позволит подключить *персональный архив пакетов* (Personal Package Archives, PPA), с помощью которого нам представится возможность установить свободный видеодрайвер.

После ввода данной команды система запросит ваш пароль, заданный при установке ОС. Введите его и нажмите клавишу <Enter>. Терминал еще раз попросит нажать клавишу <Enter> для подтверждения ваших намерений. Нажмите <Enter>.

Поскольку первой командой мы внесли изменения в список локального репозитория, то нам как-то нужно сообщить об этом программе-установщику. Обновление списка репозитория выполняется следующей командой:

```
sudo apt-get update
```

ПРИМЕЧАНИЕ

О том, что такое *репозиторий*, мы поговорим немного позже.

В ответ на это обновится список пакетов, и теперь нужно обновить систему:

```
sudo apt-get dist-upgrade
```

По мере выполнения команды Терминал запросит подтверждения ваших намерений. Для подтверждения нажмите клавишу <y> (либо <д>, если у вас включена русскоязычная раскладка клавиатуры) и затем клавишу <Enter>.

Теперь осталось только перезагрузить компьютер, и ждущий режим будет работать должным образом.

ПРИМЕЧАНИЕ

Перейти в ждущий либо спящий режим (гибернация) можно следующими командами: `sudo pm-hibernate` для перевода компьютера в режим гибернации (спящий режим), `sudo pm-suspend` — ждущий режим.

Пустой рабочий стол

Иногда после ввода пароля учетной записи загружается пустой рабочий стол. Это случается по разным причинам, например, из-за неправильной конфигурации настроек дисплея, а также при повреждении профиля пользователя.

При этом на рабочем столе вызывается контекстное меню, и даже можно запустить Терминал. Если вы столкнулись с такой же проблемой, то вам помогут следующие команды:

```
sudo apt-get install dconf-tools
dconf reset -f /org/compiz/
setsid unity
```

Первая команда запрашивает установку `dconf-tools`. На предложение ввода пароля введите его и нажмите клавишу <Enter>. Начнется установка. На запрос подтверждения намерений введите `y` и нажмите клавишу <Enter>.

После этого выполните вторую команду (`dconf reset -f /org/compiz/`), а затем третью.

После этих манипуляций ваш профиль будет сброшен и рабочий стол успешно запустится.

Изменение порядка загрузки операционной системы

При адаптации к ОС Ubuntu многим не по душе установленный по умолчанию порядок загрузки операционных систем при запуске компьютера. Речь идет о следующем моменте (рис. 8.5).

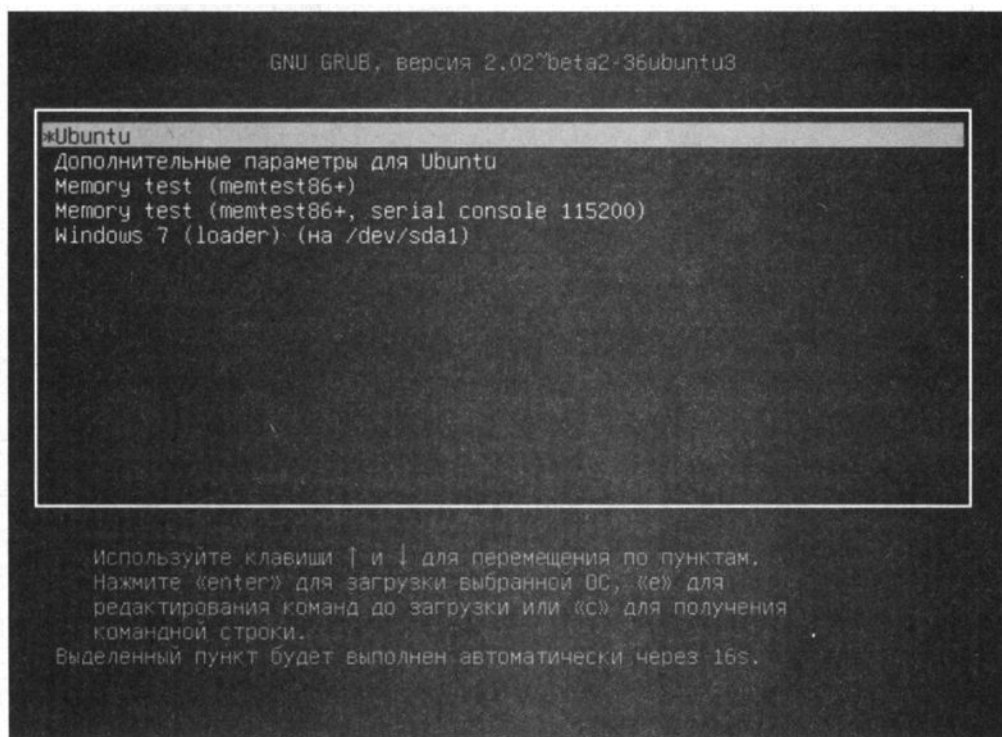


Рис. 8.5. Порядок загрузки ОС в загрузчике GRUB

Да, это уже знакомый нам загрузчик GRUB, который по умолчанию выделяет ОС Ubuntu первой в списке загрузки операционных систем. Но как быть, если мы хотим изменить этот порядок и сделать так, чтобы ОС Windows загружалась первой, а не ОС Ubuntu? Нужно отредактировать файл `/etc/default/grub` соответствующим образом.

Сейчас мы будем работать в текстовом редакторе Gedit. Он несколько удобнее, нежели Nano, с которым мы уже успели познакомиться чуть ранее.

Давайте запустим Терминал и выполним следующую команду:

```
sudo gedit /etc/default/grub
```

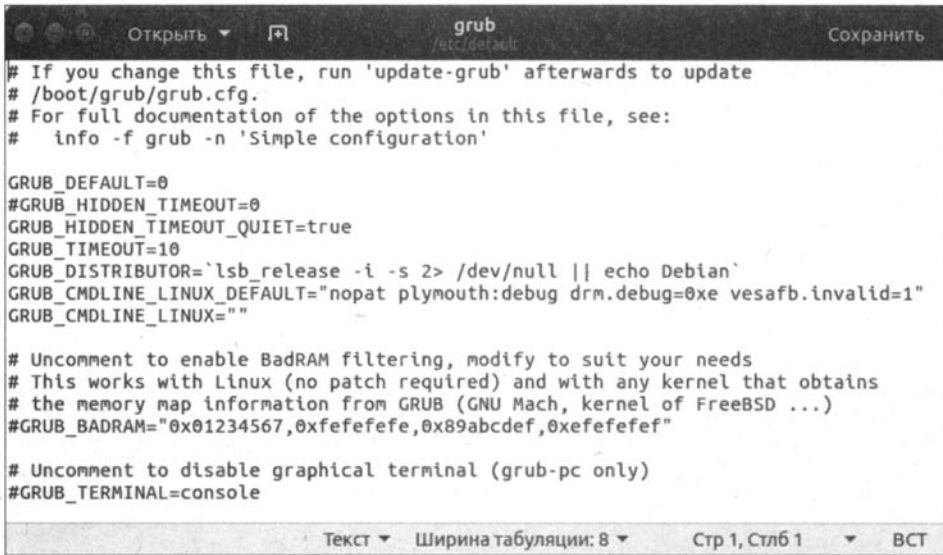


Рис. 8.6. Редактор Gedit открыл для редактирования файл /etc/default/grub

После ввода пароля откроется файл /etc/default/grub в текстовом редакторе Gedit (рис. 8.6).

Нас интересует строка `GRUB_DEFAULT=0`, которая и задает приоритет загрузки ОС. Цифра 0 означает, что будет использована первая запись в файле /boot/grub/grub.cfg.

Почему именно первая запись, если стоит 0? Дело в том, что нумерация пунктов загрузки начинается именно с нуля, а не с единицы, как мы привыкли.

Так как мы хотим по умолчанию загружать ОС Windows, то присвоим параметру `GRUB_DEFAULT` значение 4. Цифра 4 указывает на пятый пункт меню в загрузчике GRUB, а пятой в списке стоит именно ОС Windows (см. рис. 8.5). Соответственно, делаем поправку на ноль и получаем 4.

Если вам хочется изменить время отображения списка приоритетов загрузки ОС, то для этого служит параметр `GRUB_TIMEOUT`, который отвечает за время отображения этого меню. Например, если вы хотите, чтобы меню отображалось в течение полу-минуты, то присвойте параметру `GRUB_TIMEOUT` значение равное 30. Если указать значение равное -1, то список меню загрузки будет отображаться до тех пор, пока вы явно не укажете, какую ОС загружать.

После редактирования необходимо сохранить изменения в файле /etc/default/grub, нажав соответствующую кнопку в редакторе либо сочетание клавиш `<Ctrl>+<S>`. Закройте окно текстового редактора Gedit и в Терминале выполните следующую команду:

```
sudo update-grub
```

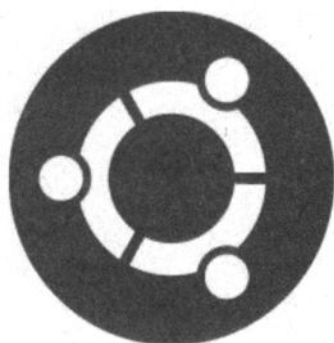
Данная команда обновит конфигурационный файл grub, и после перезагрузки компьютера вы можете увидеть сделанные вами изменения.

Выводы

Мы научились производить первоначальную настройку системы, используя оснастку **Параметры системы**, с помощью которой нам удалось сменить "горячие" клавиши раскладки клавиатуры.

Мы решили проблему с изменением системного времени в разных ОС: Windows и Ubuntu. Мы также настроили приоритет загрузки операционных систем, используя Терминал и текстовый редактор Gedit.

Немаловажным знанием и полезной настройкой, несомненно, будет возможность использования ждущего режима, который отказывался правильно работать без установленных свободных драйверов. Эта задача была решена, и проблема была устранена простой установкой видеодрайвера.

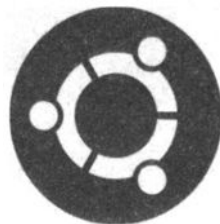


ЧАСТЬ II

Ubuntu для пользователя

Глава 9.	Структура файловой системы
Глава 10.	Типы файлов в ОС Ubuntu
Глава 11.	Администратор и суперпользователь root
Глава 12.	Знакомство с Консолью и Терминалом
Глава 13.	Полезные команды Терминала
Глава 14.	Работа с привилегиями root
Глава 15.	Установка драйверов в ОС Ubuntu
Глава 16.	Установка принтера в ОС Ubuntu
Глава 17.	Установка и удаление программ в ОС Ubuntu
Глава 18.	Установка дополнительного программного обеспечения
Глава 19.	Запуск Windows-приложений в ОС Linux
Глава 20.	Работа с архивами в ОС Ubuntu

ГЛАВА 9



Структура файловой системы

Файловый менеджер Nautilus

Nautilus — это графическая программа в ОС Ubuntu, предназначенная для комфортной работы с файловой системой в этой операционной системе, т. к. работать в графическом режиме намного удобнее, чем консольном.

Nautilus можно сравнить с Проводником Windows в семействе ОС Windows. Nautilus является стандартным файловым менеджером, который по умолчанию включен в дистрибутив ОС Ubuntu. Однако вы можете установить другой файловый менеджер, который вам больше импонирует. Например, для ОС Ubuntu можно установить такие файловые менеджеры, как MidnightCommander, Thunar, PCManFM, Dolphin, Krusader, XFE и т. п. Все зависит от ваших личных предпочтений и привычек.

Сейчас мы не будем рассматривать эти файловые менеджеры, а обсудим только стандартный файловый менеджер Nautilus.

Запустить Nautilus можно следующими способами:

- ◆ щелкнуть мышью на панели запуска по соответствующему значку;
- ◆ набрать в главном меню слово `Nautilus` и запустить найденную программу;
- ◆ воспользоваться командой `nautilus` Терминала.

Наиболее быстрый способ запуска файлового менеджера Nautilus — первый указанный вариант (рис. 9.1).



Рис. 9.1. Значок программы Nautilus

Запустится файловый менеджер Nautilus (рис. 9.2).

По умолчанию открывается домашний каталог пользователя. Обратите внимание, что скрытые файлы не отображаются. Для того чтобы включить их отображение,

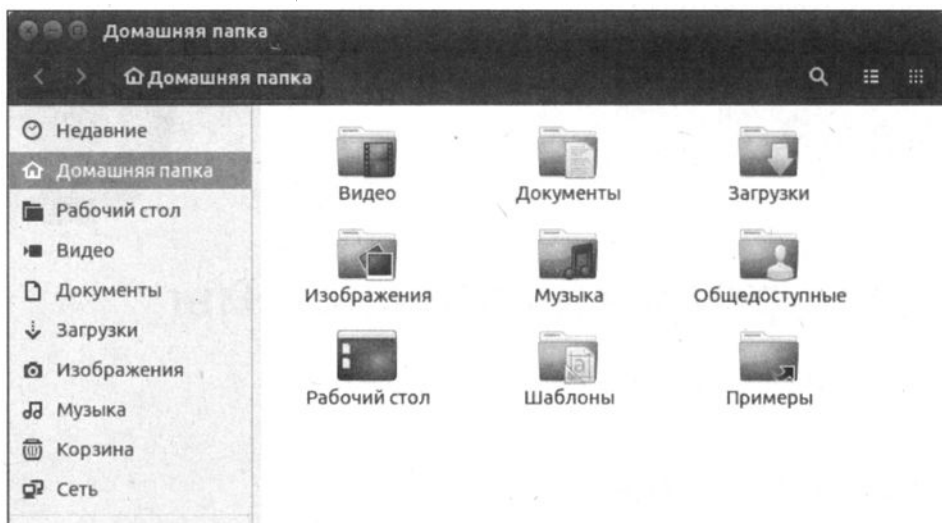


Рис. 9.2. Файловый менеджер Nautilus

в меню **Вид** выберите команду **Показывать скрытые файлы**. После вы увидите все скрытые файлы.

Имейте в виду, что скрытые файлы видны лишь до закрытия файлового менеджера Nautilus. Для того чтобы включить их постоянное отображение, выберите пункт меню **Правка**, а затем **Параметры** и в открывшемся окне настроек установите флажок **Показывать скрытые файлы и резервные копии** (рис. 9.3).

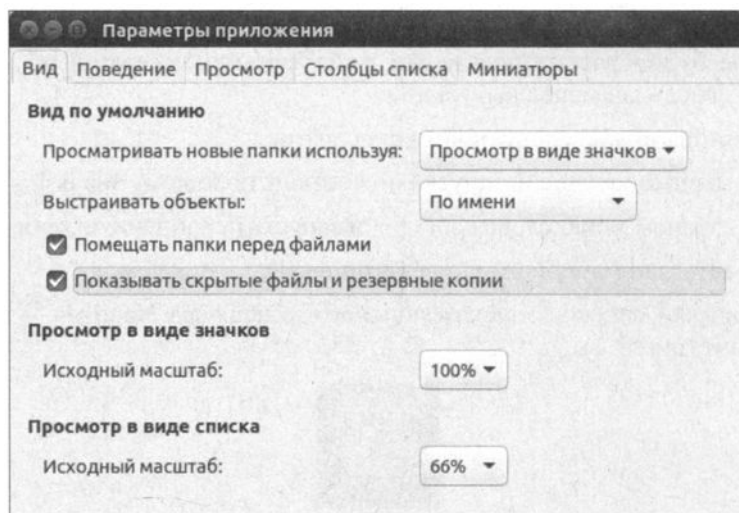


Рис. 9.3. Включение отображения скрытых файлов

После включения этого параметра скрытые файлы всегда будут отображаться.

Интерфейс файлового менеджера Nautilus ничем не отличается от привычных программ. Так, Nautilus содержит панель меню, панель инструментов, левую боковую

панель и, собственно, саму рабочую область. По сути, ничего сложного и не требующего дополнительного описания.

Имена устройств

Наверняка вы обратили внимание, что ОС Linux по-иному присваивает имена дисковым устройствам, нежели это делает ОС Windows.

В ОС Linux нет такого понятия, как *диск* или *устройство*. Вместо них создается специальный файл, через который операционная система уже "общается" с устройством, т. е. диском. Так, в ОС Linux-подобные файлы создаются для каждого жесткого диска и раздела на нем. Именно эти файлы используются при работе с жестким диском или устройством. В качестве названий устройств используются имена этих файлов. Вы их могли видеть в программе разметки жесткого диска GParted (рис. 9.4).

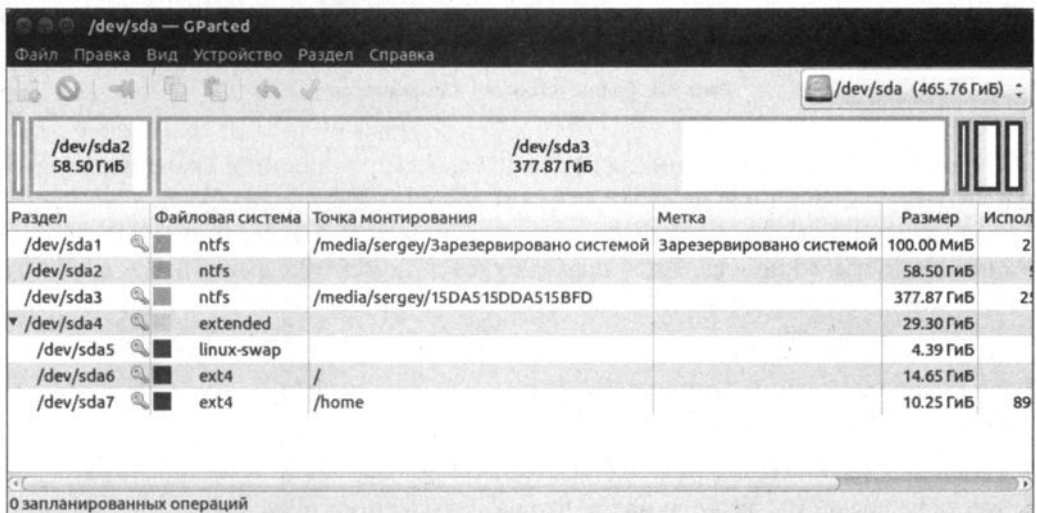


Рис. 9.4. Название устройств в программе GParted

Как видим, программа GParted сообщила нам о том, что в каталоге `/dev/` есть файлы устройств `sda1`, `sda2`, `sda3` и т. д. Сейчас у вас возникнет вопрос: можно ли воочию увидеть эти файлы? Конечно можно. Откройте файловый менеджер Nautilus и перейдите в каталог `/dev/`. Для этого в левой части окна Nautilus выберите пункт **Компьютер**, откройте требуемый каталог и найдите файлы жесткого диска (рис. 9.5).

На рис. 9.5 выделен файл жесткого диска, и помимо него есть еще 7 файлов с названиями `sda1`, ..., `sda7`, которые обозначают устройства, созданные нами при разметке жесткого диска во время установки ОС Ubuntu. Например, файл `sda7` — это раздел с домашним каталогом `/home/`, а `sda5` — это файл подкачки.

Теперь разберемся с тем, как создаются эти устройства, точнее, по какому принципу присваиваются названия устройствам.

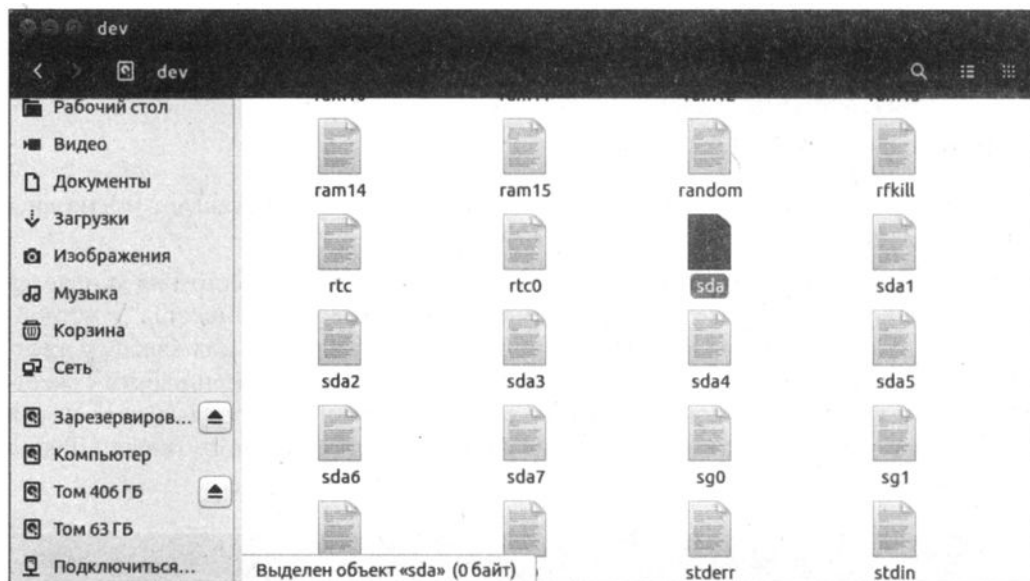


Рис. 9.5. Файлы устройств в каталоге /dev/

ОС Ubuntu Linux всем жестким дискам, вернее, всем устройствам хранения данных присваивает названия в виде `sdN`, где буква *N* — это следующая буква в алфавите. Так, когда в компьютере установлен только один жесткий диск, то он получает название `sda`, т. е. к `sd` прибавляется первая буква латинского алфавита — *a*. В случае если в компьютере установлено два физических диска, то второй жесткий диск получит название `sdb`.

Если вы подключите к компьютеру еще и флешку, то этот съемный накопитель получит название `sdс`. И так далее по аналогии.

С названием накопителей разобрались, теперь разберемся с нумерацией разделов на жестком диске. Вы же не думаете, что разделы жесткого диска будут именоваться по буквам алфавита? Нет, конечно, здесь действует несколько иной принцип.

Названия разделов на жестком диске состоят из наименования жесткого диска, например `sda`, к которому прибавляется номер раздела. В общем виде все это выглядит вот так: `sdaN`, где `sda` — это название жесткого диска, а буква *N* — это порядковый номер раздела жесткого диска. Так, если на жестком диске `sda` создано два раздела, то они получают названия в виде `sda1` и `sda2`. Третий и четвертый разделы получают наименования `sda3` и `sda4` соответственно.

Однако тут есть хитрая особенность. Первые четыре цифры зарезервированы для основных разделов, т. е. для первичных дисков, а их, как мы знаем, может быть не более 4 либо три первичных раздела и один расширенный. Поэтому название `sda5` получит только расширенный (логический) раздел. Даже если у вас на жестком диске созданы один первичный раздел и один расширенный, то таблица разделов в любом случае будет такой, как на рис. 9.6.

sda	— основной жесткий диск
sda1	— основной раздел
sda2	— расширенный раздел
sda5	— логический раздел
sda6	— логический раздел

Рис. 9.6. Таблица разделов жесткого диска

Таким образом, если вы впоследствии создадите еще один основной раздел, то он примет название sda3, а не sda7 и т. п.

ПРИМЕЧАНИЕ

Расширенный раздел — это всего лишь своеобразный контейнер для логических разделов. На него невозможно записать данные, т. к. он содержит все логические разделы.

Итак, мы разобрались с принципом именования жестких дисков и разделов в ОС Ubuntu Linux и уже можем перейти непосредственно к рассмотрению файловой системы.

Структура файловой системы

Современные операционные системы используют *древовидную систему* организации файлов, так называемое *дерево каталогов*. "Что это еще за деревья?" — спросите вы.

Если обычные файлы представить в виде листьев, а каталоги ветками, то такая система организации напоминает дерево. Немного странное сравнение, но тем не менее оно очень часто используется при описании организации современных файловых систем.

На рис. 9.7 схематически показана структура древовидной файловой системы в ОС Linux.

В ОС Ubuntu Linux существует такое понятие, как корень файловой системы.

Корень файловой системы — это точка отсчета в файловой структуре, от которой принято вести "отсчет" пути к файлу или какому-либо каталогу. Другими словами, корень файловой системы — это корневой каталог, содержащий всю файловую иерархию.

В ОС Windows вам не раз приходилось работать с дисками C:, D: и т. д. Именно они и являются корнями файловой системы — сколько дисков, столько и корней. В ОС Ubuntu Linux корнем файловой системы является прямой слеш /, и именно относительно этого корня ведется отсчет пути к файлу.

В ОС Ubuntu Linux корневая файловая система часто называется — *root*, т. е. *корень* по-английски.

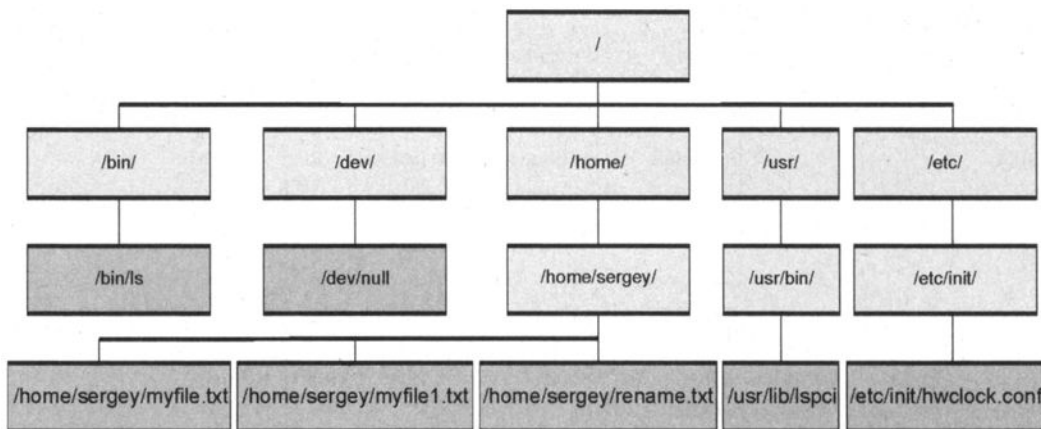


Рис. 9.7. Древовидная организация файловой системы

Корневая файловая система организована в виде файловой иерархии. Такая система организации структуры файловой системы основывается на стандарте Filesystem Hierarchy Standard (FHS — стандарт иерархии файловой системы), который описывает, какая информация должна находиться в том или ином каталоге. Однако в силу особенностей разработки ОС часто приходится отступать от этого стандарта.

Стоит отметить, что любой объект в файловой системе ОС Ubuntu Linux считается *файлом*. Это означает, что привычный для нас каталог, который в ОС Windows называется *папкой*, является всего лишь файлом, т. е. на самом деле это последовательность байтов на физическом диске. Не пугайтесь, когда в технической литературе вместо слова "каталог" употребляют слово "файл".

Давайте теперь более пристально рассмотрим организационную структуру файловой системы ОС Ubuntu.

Общее описание структуры каталогов

Если мы развернем корень файловой системы, то сможем наблюдать следующую картину (рис. 9.8).

ПРИМЕЧАНИЕ

Корень файловой системы можно развернуть следующим способом: запустите файловый менеджер Nautilus и в левой части окна щелкните на пункте **Компьютер**. Вы перейдете в корень файловой системы.

Среди каталогов, находящихся в корне файловой системы, есть уже знакомые нам каталоги `/home/`, `/dev/` и `/etc/`, с которыми мы успели немного познакомиться.

В общем виде, в ОС Ubuntu присутствуют следующие каталоги (табл. 9.1).

Это только основные каталоги в ОС Ubuntu, на деле их конечно же гораздо больше, и нет смысла описывать все, т. к. их количество варьирует от установленных пакетов. Описывать каждый в отдельности не имеет смысла.

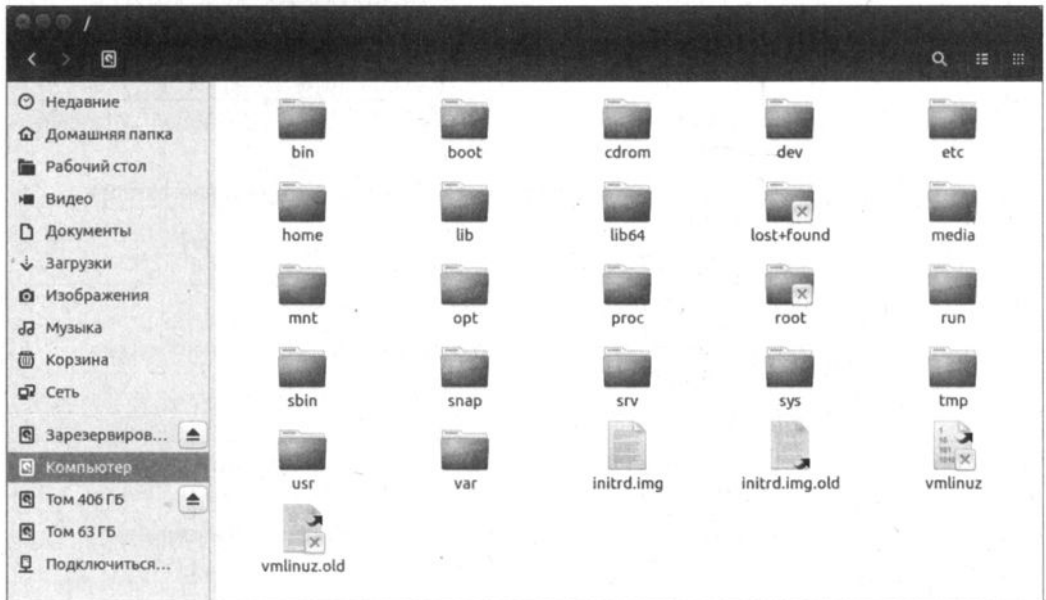


Рис. 9.8. Корень файловой системы

Таблица 9.1. Структура файловой системы

Каталог	Описание
/	Корневой каталог, содержит всю файловую иерархию
~	Сокращенное обозначение домашнего каталога пользователя
/bin/	Каталог с основными пользовательскими утилитами, которые необходимы для работы с ОС
/boot/	Каталог с загрузочными файлами ОС, необходимыми для запуска операционной системы
/cdrom/	Содержит устройство CD-привода
/dev/	Каталог с основными файлами устройств, такими как жесткие диски и т. п.
/etc/	Каталог с основными конфигурационными файлами ОС и установленными программами
/etc/X11/	Каталог с файлами конфигурации X Window System версии 11
/etc/apt/	Каталог с файлами конфигурации пакетного менеджера apt
/etc/dpkg/	Каталог с файлами конфигурации пакетного менеджера dpkg
/home/	Содержит домашние каталоги пользователей, а также персональные настройки и данные
/lib/	Каталог с необходимыми библиотеками для запуска программ из каталогов /bin/ и /sbin/
/lib64/	Каталог с необходимыми библиотеками для 64-битных программ

Таблица 9.1 (продолжение)

Каталог	Описание
/lost+found/	Каталог с файлами, на которые нет ссылок, например, с файлами, удаление которых было аварийно завершено
/media/	Точки монтирования съемных носителей, например флеш-накопителей, DVD-привода и т. п.
/mnt/	Содержит пользовательские точки монтирования
/opt/	Содержит дополнительное программное обеспечение
/proc/	Содержит виртуальную файловую систему procfs, которая необходима для работы модулей ядра
/root/	Домашний каталог пользователя root
/run/	Централизованное хранилище временных файлов, которые необходимы для запуска служб на ранних стадиях загрузки ОС
/sbin/	Каталог с основными системными программами, предназначенными для администрирования и управления ОС
/snap/	Каталог с файлами Snap-пакетов
/srv/	Каталог с данными, специфичными для окружения операционной системы
/sys/	Содержит монтированную виртуальную файловую систему sysfs
/tmp/	Каталог с временными файлами
/usr/	Каталог с установленными пакетами программ, сопровождающей документацией, а также исходными кодами самих программ
/usr/bin/	Каталог с дополнительными программами для всех учетных записей
/usr/sbin/	Каталог с дополнительными системными программами, например службой сетевых сервисов
/usr/include/	Каталог со стандартными заголовочными файлами
/usr/games/	Каталог с пользовательскими играми
/usr/lib/	Каталог с библиотеками программ для каталогов /usr/bin/ и /usr/sbin/
/usr/bin/ /usr/sbin/	Каталог с дополнительными системными программами
/usr/src/	Содержит исходные коды ядра
/usr/local/	Каталог с установленными программами, специфичными для данного хоста
/var/	Каталог с изменяемыми файлами, такими как log-файлы, временные файлы очереди печати и т. п.
/var/cache/	Каталог с данными кэша приложений, которые необходимы для установки того или иного приложения
/var/lib/	Каталог с данными, изменяемыми программами в процессе работы
/var/lock/	Каталог с файлами, которые недоступны для изменения в текущий момент времени
/var/log/	Каталог с файлами журнала регистраций различных приложений

Таблица 9.1 (окончание)

Каталог	Описание
/var/run/	Каталог с информацией о запущенных программах
/var/spool/	Каталог с файлами очереди печати
/var/tmp/	Каталог с временными файлами, которые должны быть сохранены между перезагрузками ОС

Корневая файловая система

Как вы уже знаете, в ОС Ubuntu Linux корневая файловая система обозначается прямым слешем /. Именно от этого слеша и ведется отсчет по пути к файлу или каталогу (рис. 9.9).

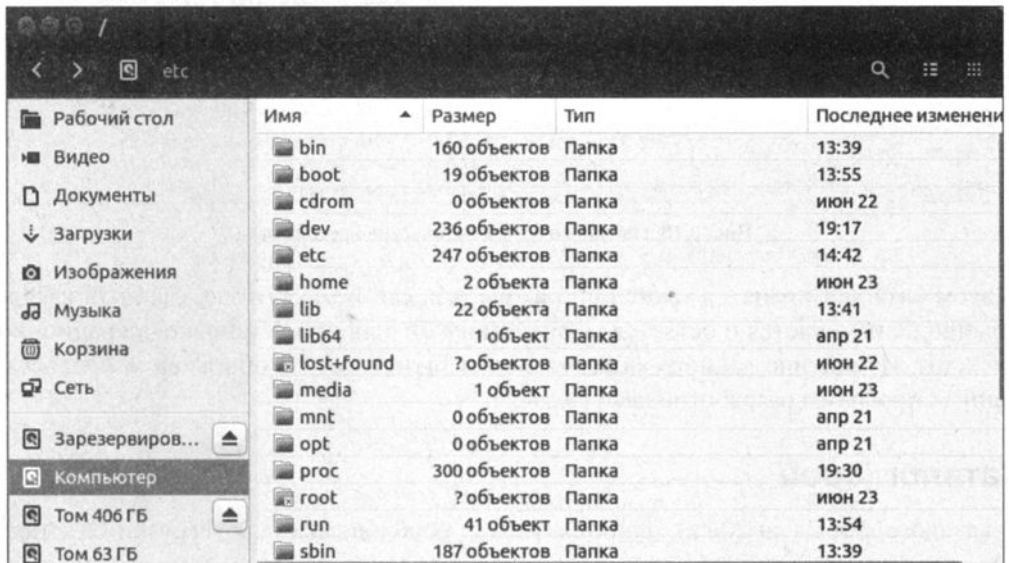


Рис. 9.9. Открыта корневая файловая система

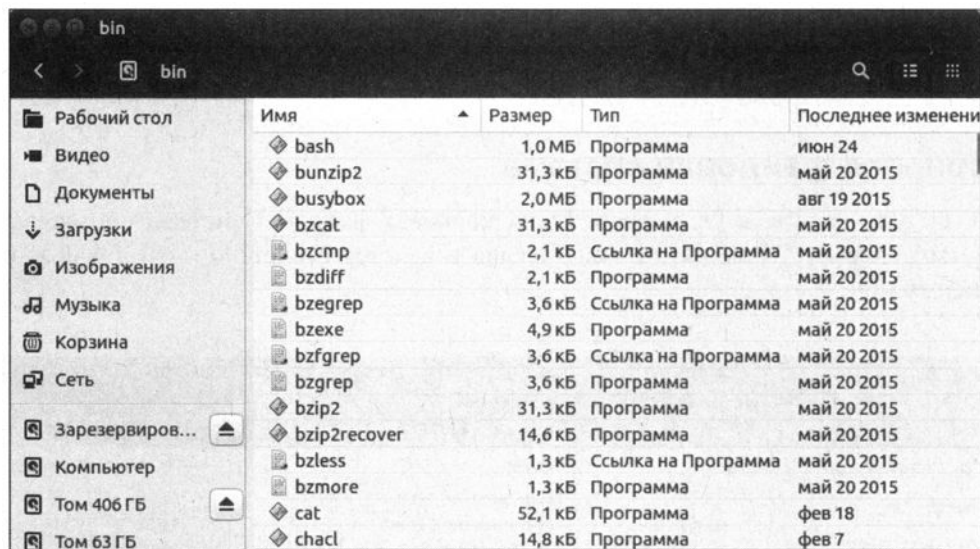
Корневая файловая система является отправной точкой и содержит все остальные подкаталоги и файлы, такие как /home/, /boot/, /bin/ и т. д.

Домашний каталог пользователя

Тильдой ~ обозначается домашний каталог пользователя. Таким образом, если вы в пути к файлу или каталогу видите символ ~, то это означает, что данный файл или каталог находится в домашнем каталоге пользователя. Например, путь к файлу ~/Sample/text.txt будет равносильен длинной записи /home/Sample/text.txt.

Каталог /bin/

В каталоге /bin/ (от англ. *binaries* — двоичные файлы) содержатся основные пользовательские программы, т. е. основные команды, которые необходимы пользователю в процессе работы с операционной системой (рис. 9.10).



Имя	Размер	Тип	Последнее изменение
bash	1,0 МБ	Программа	июн 24
bunzip2	31,3 кБ	Программа	май 20 2015
busybox	2,0 МБ	Программа	авг 19 2015
bzcat	31,3 кБ	Программа	май 20 2015
bzcmp	2,1 кБ	Ссылка на Программа	май 20 2015
bzdiff	2,1 кБ	Программа	май 20 2015
bzegrep	3,6 кБ	Ссылка на Программа	май 20 2015
bzexe	4,9 кБ	Программа	май 20 2015
bzfgrep	3,6 кБ	Ссылка на Программа	май 20 2015
bzgrep	3,6 кБ	Программа	май 20 2015
bzip2	31,3 кБ	Программа	май 20 2015
bzip2recover	14,6 кБ	Программа	май 20 2015
bzless	1,3 кБ	Ссылка на Программа	май 20 2015
bzmore	1,3 кБ	Программа	май 20 2015
cat	52,1 кБ	Программа	фев 18
chacl	14,8 кБ	Программа	фев 7

Рис. 9.10. Основные пользовательские программы

В этом каталоге хранятся такие программы, как `cat`, `ls`, `echo` и т. д. Данный каталог обычно не изменяется и остается таким, каким он был при установке операционной системы. Изменение данного каталога может быть в случае обновления предоставленных программ разработчиками ОС.

Каталог /boot/

В каталоге /boot/ (загрузка) хранятся файлы, необходимые для загрузки ОС, такие как загрузчик `grub`, модули ядра, модуль тестирования памяти и т. п. Каталог практически никогда не изменяется в процессе работы с ОС Ubuntu.

Каталог /cdrom/

Каталог /cdrom/ является исторически выбранной точкой монтирования для CD-дисков и, возможно, вскоре будет изъят разработчиками за ненадобностью, т. к. все подключаемые устройства подобного типа монтируются в каталоге /media/.

Каталог /dev/

В каталоге /dev/ (от англ. *device* — устройство) размещены файлы устройств, которые присутствуют в системе. Для того чтобы существовала возможность получения

доступа к определенному устройству, необходимо наличие соответствующего файла в каталоге `/dev/`.

В каталоге `/dev/` можно обнаружить файл жесткого диска, который называется `sda`, а также файлы CD- и DVD-приводов и т. п.

Кроме файлов физических устройств, в этом каталоге также хранятся файлы псевдоустройств, т. е. файлы виртуальных устройств. Для таких виртуальных устройств не существует реального физического устройства. Например, файл `/dev/null` является виртуальным, и для него не существует реального физического устройства. Данный файл является специальным устройством, при обращении к которому удаляются все отправленные ему данные.

Еще одним ярким примером виртуального устройства служит файл `/dev/random`, который генерирует случайные числа. В компьютере нет такого специального физического устройства, задачей которого являлась бы генерация случайных чисел. Подобное устройство просто бессмысленно, когда ту же задачу может выполнить обычная программа. Такого рода программой и является виртуальное устройство `/dev/random`.

Каталог `/etc/`

В каталоге `/etc/` (от лат. *et cet̃era* — и другие) хранятся общесистемные конфигурационные файлы. Так, в данном каталоге хранятся конфигурационные файлы менеджеров пакетов `apt` и `dpkg`. Кроме того, в данном каталоге хранятся файлы конфигурации X Window System версии 11, настройки графической оболочки среды рабочего стола GNOME и множество других общесистемных конфигурационных файлов.

ПРИМЕЧАНИЕ

Пользовательские конфигурационные файлы хранятся в домашнем каталоге каждого пользователя.

Каталог `/home/`

В каталоге `/home/` хранятся домашние каталоги каждого пользователя ОС Ubuntu Linux. Например, если в вашей ОС заведена только одна учетная запись, то в этом каталоге будет как минимум два каталога: `lost+found` (для хранения файлов, на которые нет ссылок) и каталог с именем вашей учетной записи (рис. 9.11).

В самом каталоге с именем вашей учетной записи (или какой-либо другой учетной записи) хранятся файлы с пользовательскими данными, а также конфигурационные файлы конкретной учетной записи.

Таким образом, в данном каталоге помимо ваших личных файлов и данных хранятся файлы конфигурации вашей учетной записи. Среди этих файлов можно найти файлы конфигураций установленных вами программ, а также файлы персональной настройки системы, которые вы когда-либо производили.

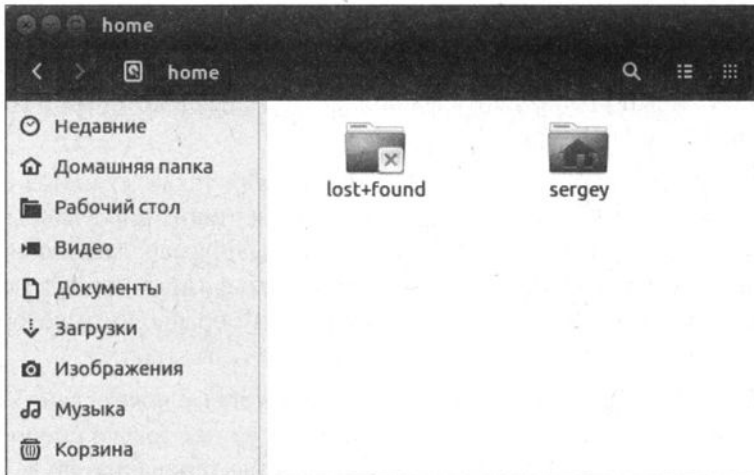


Рис. 9.11. Домашний каталог пользователя

Каждый пользователь имеет право записи только в свой каталог. Право изменять и читать каталоги других учетных записей имеется только у пользователя root.

Каталоги `/lib/` и `/lib64/`

В каталогах `/lib/` и `/lib64/` (от англ. *library* — библиотека) находятся системные библиотеки (32- и 64-битные соответственно), необходимые для запуска двоичных файлов, которые хранятся в каталогах `/bin/` и `/sbin/`.

Кроме этого, в каталоге `/lib/` хранятся модули ядра операционной системы.

Каталог `/lost+found/`

В каталоге `/lost+found/` хранятся файлы, на которые нет ссылок. Такое часто бывает, когда при удалении файла произошел сбой в работе компьютера, например, отключилось питание компьютера. При следующем включении компьютера происходит проверка файловых систем на наличие ошибок, а так как ошибка была найдена, то запускается программа `fsck`. После этого поврежденные и восстановленные программой `fsck` файлы будут помещены в каталог `/lost+found/`, зайдя в который, вы можете самостоятельно их просмотреть и при необходимости переместить их в нужное вам место.

Каталог `/media/`

В каталоге `/media/` находятся подкаталоги съемных носителей, которые подключаются к компьютеру. Например, когда вы вставляете DVD-диск, то в этом каталоге автоматически будет создан соответствующий подкаталог. После этого вы можете зайти в каталог `/media/` и начать работу с подключенным носителем информации.

Каталог /mnt/

Каталог /mnt/ (от англ. *mount* — монтировать) предназначен для ручного подключения устройств. Он, по сути, играет роль временной точки монтирования подключаемых устройств, таких как жесткие диски, дискеты и т. п.

Каталог /opt/

В каталоге /opt/ (от англ. *optional* — опциональный) размещаются дополнительные пакеты программ, установленные пользователем в ОС. Как правило, данные пакеты программного обеспечения предоставлены сторонними разработчиками, а не разработчиками дистрибутива.

Подкаталоги в данном каталоге имеют собственную структуру, и зачастую для того чтобы удалить установленный пакет, достаточно просто удалить одноименный каталог из /opt/.

ПРИМЕЧАНИЕ

Удаление пакетов путем простого удаления каталогов является неправильным и рекомендуется к использованию только в самых крайних случаях.

Каталог /proc/

Каталог /proc/ предназначен для нужд псевдофайловой системы procfs. Почему псевдофайловой? Дело в том, что на жестком диске нет такой файловой системы, как procfs, т. к. она расположена в виртуальной памяти и монтируется во время работы операционной системы.

В данном каталоге хранятся файлы, в которых находится информация о системе и процессах. Так, в файле /proc/modules хранится информация о загруженных модулях ядра, а в файле /proc/cpuinfo — информация о процессоре вашего компьютера. В файле /proc/uptime хранится время, в течение которого система находится в рабочем состоянии. Это далеко не весь список файлов данного каталога.

Каталог /root/

Каталог /root/ предназначен для пользователя с именем root. Данный каталог специально был помещен в корневой раздел файловой системы, а не в каталог /home/. Сделано это было для того, чтобы предотвратить возможные проблемы с доступом к каталогу /home/, т. к. в случае невозможности получить доступ к каталогу /home/ остается возможность получить доступ к каталогу /root/. Например, данное решение позволит загрузить ОС, когда домашний каталог пользователя вынесен на другой жесткий диск, который по той или иной причине отказал. У администратора операционной системы остается возможность загрузиться с другой учетной записью, которая находится на том же диске, что и корневой каталог /.

Каталог `/run/`

Каталог `/run/` в дистрибутивах ОС на базе Linux появился сравнительно недавно, и его появление обусловлено тем, что приложениям при запуске теперь предоставляется возможность сохранять в нем вспомогательные файлы, которые им понадобятся для дальнейшей работы. Логично, что хранить такие файлы в каталоге `/tmp/` нецелесообразно, т. к. они могут быть удалены системой еще до того, как приложение ими успеет воспользоваться.

Каталог `/sbin/`

Каталог `/sbin/` предназначен для хранения двоичных файлов, которые служат для управления и администрирования ОС. Данный каталог отличается от каталога `/bin/` тем, что в последнем хранятся программы, не предназначенные для выполнения административных задач.

Каталог `/snap/`

Каталог `/snap/` в ОС Ubuntu появился сравнительно недавно, а именно в версии дистрибутива 16.04. Данный каталог предназначен для установки программ, которые не имеют зависимостей. Например, в подобной программе, точнее, в Snap-пакете, уже содержатся все необходимые библиотеки и для запуска такой программы. Таким образом, появляется возможность установки программ, а затем последующего их обновления без обновления всей ОС.

Так, у вас появляется возможность в случае сбоя или ошибки Snap-пакета откатиться на предыдущую версию пакета, не влияя на работу самой операционной системы.

Каталог `/srv/`

Каталог `/srv/` предназначен для хранения данных сервисных служб, представляемых операционной системой.

Изначально предполагалось, что в этом каталоге будут храниться данные, с которыми работает та или иная служба. Например, намечалось, что в случае использования сервера Apache HTTP все файлы сайта, обслуживаемые данной службой, будут храниться в каталоге `/srv/`. Однако исторически сложилось так, что для данной службы вместо каталога `/srv/` стал использоваться каталог `/var/www/`. По этой причине каталог `/srv/` пустует.

Каталог `/sys/`

Каталог `/sys/` содержит виртуальную файловую систему `sysfs`, которая добавляет в пространство пользователя служебную информацию ядра, например информацию о драйверах и т. п.

Каталог /tmp/

Каталог /tmp/ (от англ. *temporary* — временный) предназначен для хранения приложениями временных данных, которые нужны им в процессе работы. Обычно по завершении работы приложения все временные данные удаляются, а сам каталог /tmp/ очищается при загрузке операционной системы. Очищать данный каталог можно и вручную, однако стоит иметь в виду, что некоторые файлы могут использоваться в тот или иной момент времени, а их несвоевременное удаление может привести к ошибке в работе программы. Разумеется, хранить свои личные данные в этом каталоге — весьма глупое занятие.

Каталог /usr/

Каталог /usr/ (от англ. *user* — пользователь) предназначен для хранения установленных программ, документации, исходного кода ядра и т. п. К данному каталогу разрешен доступ только для чтения, кроме пользователя root.

Структура подкаталогов /usr/ несколько напоминает корневой раздел файловой системы, т. к. в каталоге /usr/ присутствуют подкаталоги /bin/, /sbin/, /lib/ и /src/. Назначение этих каталогов совпадает с назначением корневых каталогов, за исключением того, что корневые каталоги предназначены для нужд системных программ, а те же каталоги в /usr/ — для нужд прикладных программ.

Каталог /usr/ является одним из самых больших в ОС Ubuntu Linux.

Каталог /var/

Каталог /var/ (от англ. *variable* — переменный) используется для хранения часто меняющихся данных, таких как журналы работы программ, cache-файлы, файлы почтового и Web-серверов, файлы заданий очереди печати.

Так, файлы журнала различных программ хранятся в каталоге /var/log/, а в каталоге /var/run/ находятся файлы запущенных программ. В каталоге /var/spool/ размещаются сведения о заданиях для печати, а также информация работы с электронной почтой.

Выводы

Мы рассмотрели принцип, по которому формируются наименования жестких дисков в ОС Ubuntu Linux, а также принцип именования разделов жесткого диска и узнали, что в ОС Ubuntu Linux номер раздела sda5 зарезервирован для расширенного раздела, а первые четыре — только для основных разделов.

Мы познакомились со структурой файловой системы в ОС Ubuntu Linux и с такими понятиями, как иерархическое дерево и корень файловой системы. Мы узнали о назначении каталогов, которыми располагает файловая система в ОС Ubuntu.

ГЛАВА 10



Типы файлов в ОС Ubuntu

О файлах и каталогах

Разделителем в каталогах путей до файлов является прямой слеш /. Именно он используется в операционной системе семейства Linux, а не обратный слеш \, как в ОС Windows.

При описании назначения каталогов файловой системы в конце каждого названия каталога используется *прямой слеш*. Многие могут возразить, якобы писать его совсем необязательно. Однако на это замечание есть весомый аргумент: в ОС Ubuntu Linux у файлов зачастую отсутствует расширение, поэтому если опустить в наименовании каталога прямой слеш, то с первого взгляда будет сложно понять, каталог это или файл.

Так, если в названии каталога /var/log/ опустить прямой слеш в конце, то получится, что речь идет не о каталоге, а о файле /var/log, т. е. о файле log в каталоге /var/. Разумеется, что такого файла в корне каталога /var/ не должно быть, но все же лучше прямо указывать, где каталог, а где файл.

Раз уж речь пошла о слешах, то стоит упомянуть, что в названии файлов или каталога вы вольны использовать любые символы, кроме прямого слеша /. Обратный слеш \ разрешается использовать.

В ОС Windows вы могли заметить, что данная операционная система не позволяла создавать файлы, в названии которых встречалось двоеточие :. ОС Ubuntu Linux разрешает эту вольность, поэтому вы можете использовать символ двоеточия там, где вам это необходимо.

И последняя маленькая особенность файловой системы в ОС семейства Linux — *регистрозависимость*. Это означает, что Sample.txt и sample.txt — два совершенно разных файла, которые вы можете спокойно создать в одном каталоге. Собственно говоря, двумя файлами это действие не ограничивается, т. к. можно создать, например, файлы sAmple.txt, SAMPLE.txt, SAPle.txt, sampLE.txt, и это будут разные файлы.

Содержимое и расширение файлов

Когда вы работали в ОС Windows, то могли заметить, что в этой операционной системе тип файла определяется *его расширением*, и уже исходя из расширения происходит ассоциация с программой, которая умеет работать с файлами этого типа.

В данном случае, расширение — это последняя часть имени файла после точки, например, у файла Sample.txt расширением будут являться последние три буквы: txt. Так ОС Windows сопоставляет тип файла с программой, которая работает с файлами этого типа.

ОС Ubuntu в большинстве случаев определяет тип файла *по его содержимому*. Это означает, что тип файла Sample.txt будет определен не по его расширению, а по тому, что находится внутри файла. Давайте на примере рассмотрим этот момент.

Создайте обычный текстовый документ в домашнем каталоге. Для этого щелкните в нем правой кнопкой мыши и выберите пункт меню **Создать документ**, а затем выберите **Создать пустой документ**. Присвойте этому документу имя Sample.txt.

После этого аналогичным образом создайте документ с названием Music_1.mp3, а потом скопируйте в этот каталог музыкальную композицию и переименуйте ее в Music_2.mp3. Теперь переключитесь на режим отображения списком, щелкнув мышью по кнопке **Показывать объекты в виде списка** в правом верхнем углу окна файлового менеджера Nautilus (рис. 10.1).

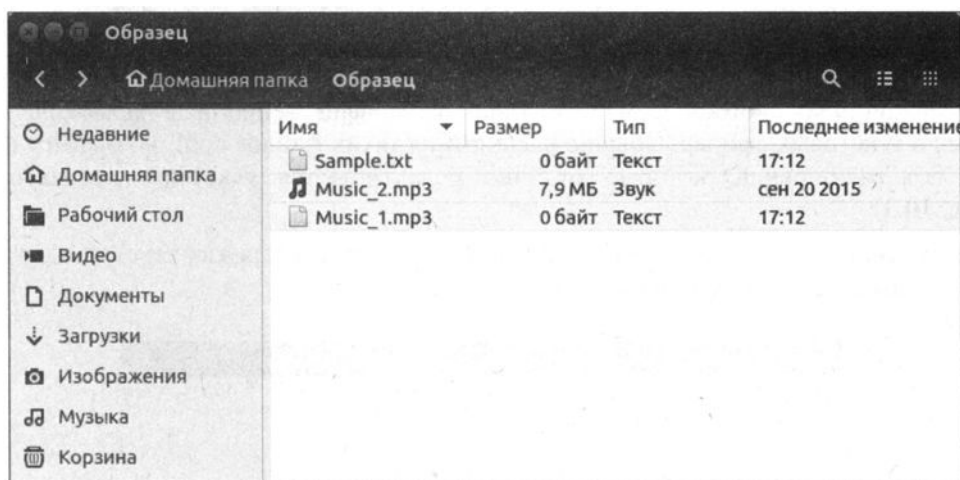


Рис. 10.1. Тип документа в файловом менеджере

Теперь обратите внимание на тип файла в колонке **Тип**. Созданные нами документы Sample.txt и Music_1.mp3 определяются как текстовые, а вот документ Music_2.mp3 значится как **Звук**.

Давайте теперь переименуем Music_2.mp3 в Text.txt и посмотрим, что получится (рис. 10.2).

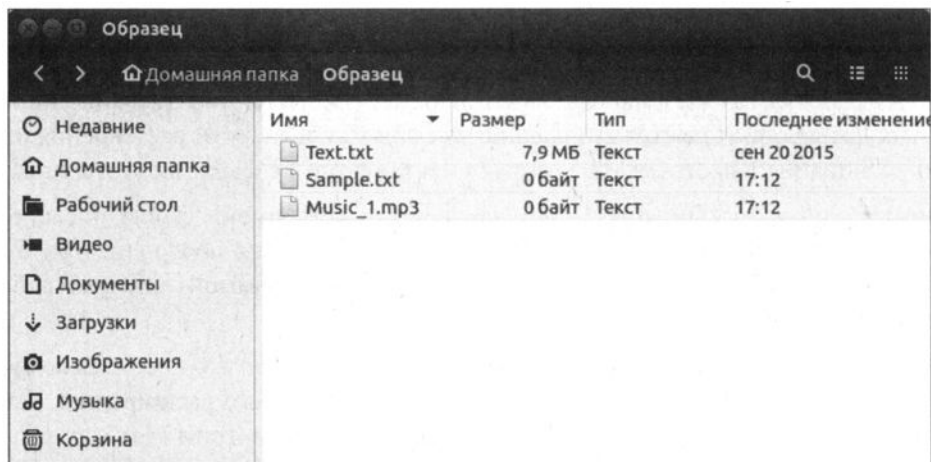


Рис. 10.2. Файловый менеджер определяет файл Text.txt как текстовый

Теперь наш аудиофайл Text.txt определяется как текстовый. Почему же так произошло? Это всего лишь особенность файлового менеджера Nautilus, который пытается сопоставить тип файла с его расширением, т. к. на считывание содержимого файла требуется некоторое время, а это затрачивает ресурсы процессора и прямо влияет на производительность. В других файловых менеджерах такой трюк может не пройти, и файл будет определен именно по его содержимому, а не по расширению.

Текстовые файлы

В операционных системах семейства Linux все конфигурационные параметры хранятся в текстовых файлах. Обычно расширения таких файлов `conf`, но бывают файлы без расширения, которые отлично редактируются текстовым редактором (рис. 10.3).

Как вы видите, текстовый файл `hosts` успешно определяется как текстовый, о чем свидетельствует строка **Тип** окна свойств файла.

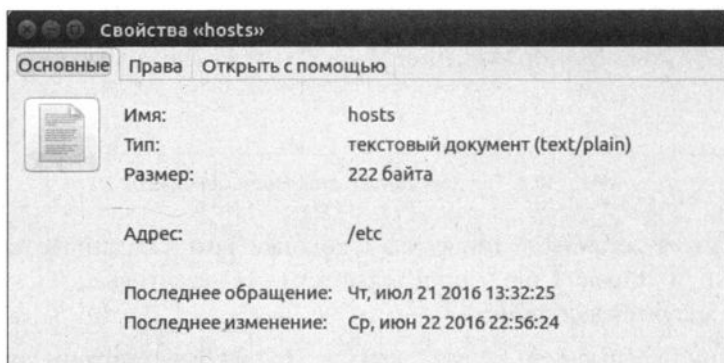


Рис. 10.3. Файл hosts определен как текстовый

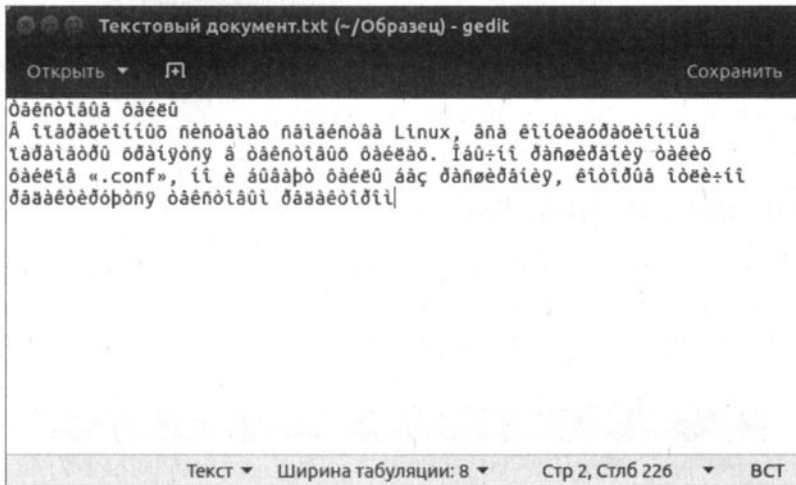


Рис. 10.4. Кракозябры вместо текста

Одна интересная особенность, которая кроется при работе с текстовыми файлами, созданными в ОС Windows, — это наличие так называемых кракозябр вместо текста (рис. 10.4).

Случается это по той причине, что документы, созданные в ОС Windows, по умолчанию сохраняются в кодировке Windows-1251 вместо более распространенной UTF-8.

Для корректного открытия документа, созданного в ОС Windows, нужно при открытии файла в текстовом редакторе Gedit прямо указать ту кодировку, в которой был создан документ. Для этого на панели меню текстового редактора Gedit выберите пункт **Открыть** и в окне выбора файла укажите кодировку **Кириллица (WINDOWS-1251)**, а затем нажмите кнопку **Открыть** (рис. 10.5).

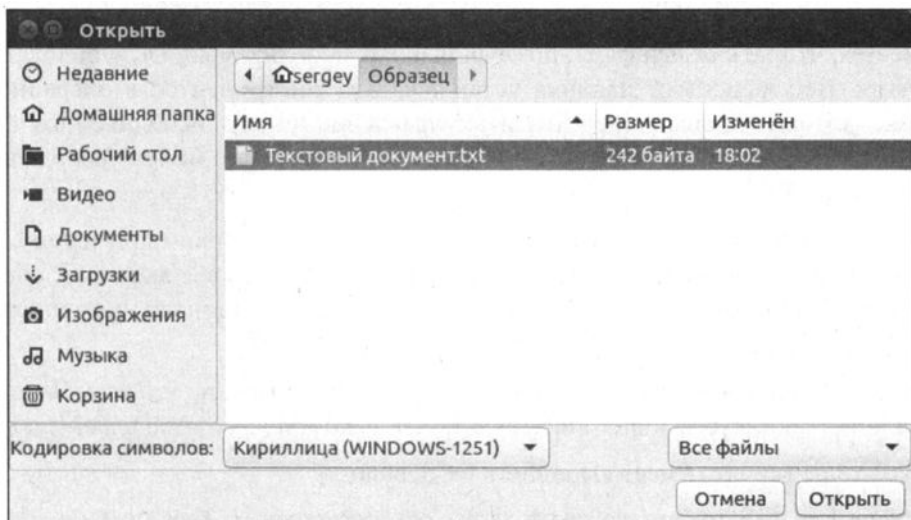


Рис. 10.5. Выбор кодировки для документов, созданных в ОС Windows

ПРИМЕЧАНИЕ

Если у вас документ повторно открылся с кракозябрами, попробуйте закрыть его в текстовом редакторе и только после открывайте его снова, т. к. редактор Gedit не открывает еще одну копию уже открытого документа.

Исполняемые файлы

В отличие от ОС Windows, в ОС Ubuntu Linux исполняемым файлом является любой файл, который помечен как *исполняемый*. Это означает, что любой файл, помеченный таким образом, может быть выполнен (запущен) в ОС Ubuntu (рис. 10.6).

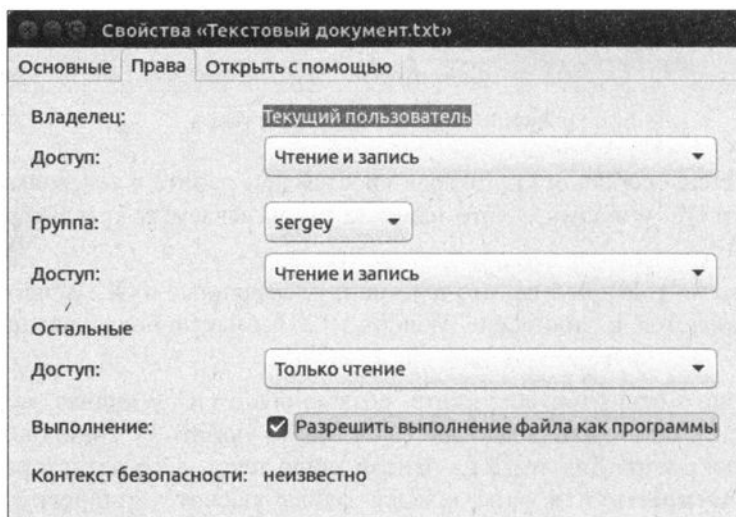


Рис. 10.6. Текстовый документ помечен как исполняемый

Разумеется, что не каждый файл, помеченный как исполняемый, ОС Ubuntu сможет запустить. Все зависит от наличия установленных инструментов в операционной системе, которые должны перехватить управление такого исполняемого файла. Если подходящего инструмента не будет, то соответственно ничего необыкновенного не произойдет.

Исполняемым файлом может быть любой текстовый файл. Ярким примером служат файлы языка программирования PHP, которые по сути своей являются обычными текстовыми файлами, но содержат инструкции командного интерпретатора (рис. 10.7).

Такие файлы по умолчанию получают право на выполнение, но это возможно лишь при наличии соответствующего инструмента, т. е. *командного интерпретатора*.

Однако это не все, что умеет выполнять ОС Ubuntu.

Исполняемым файлом может быть *файл без расширения*. Как правило, именно файлы без расширения являются какой-либо программой в семействе ОС Linux.

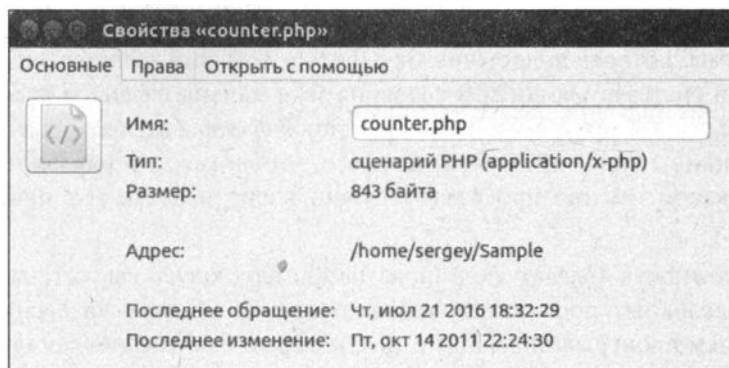


Рис. 10.7. Файл с инструкциями языка программирования PHP

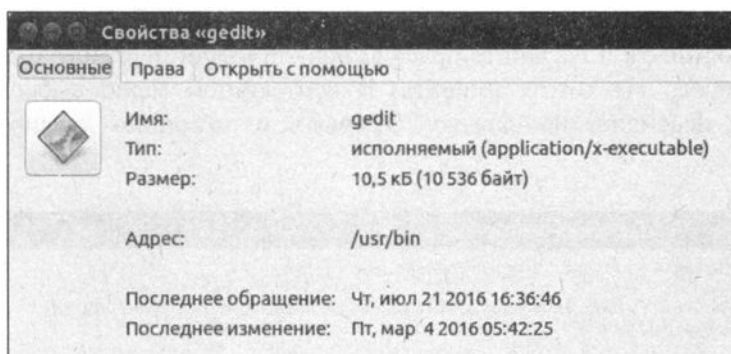


Рис. 10.8. Исполняемый файл текстового редактора Gedit

Например, все тот же текстовый редактор Gedit вовсе не имеет расширения, но является исполняемым (рис. 10.8).

Таким образом, практически за любым файлом может быть закреплено свойство выполнения, что является одним из свойств прав доступа владельца файла, но об этом чуть позже.

Прочие файлы

Все остальные файлы, будь то графические или мультимедийные, также успешно распознаются в ОС Ubuntu, а в случае если тот или иной файл по какой-то причине не может быть открыт, то следует определиться с типом файла и поискать соответствующую программу для работы с файлом этого типа.

Изменение ассоциации файлов

Как вы уже могли догадаться, ассоциация файлов — это вовсе не какая-то определенная организация группы файлов, которая имеет свой устав и проталкивает свои взгляды в массы. Ну да ладно, шутки в сторону, и поговорим о том, как работать с ассоциацией файлов в ОС Ubuntu.

Ассоциация файлов — это назначение тому или иному типу файла соответствующей программы, которая поддерживает работу с файлами данного типа. Например, операционная система для воспроизведения музыкальных файлов ассоциирует эти файлы с конкретной программой, которая должна обрабатывать, т. е. воспроизводить файл данного типа. Это нужно для того, чтобы каждый раз не указывать ОС, с помощью какой именно программы нужно воспроизвести тот или иной музыкальный файл.

Данная возможность бывает полезной, например, когда вы устанавливаете несколько музыкальных проигрывателей и хотите, чтобы музыка воспроизводилась именно в вашем проигрывателе, а не в том, который был установлен по умолчанию. То же самое применимо и к другим типам файлов, например графическим, текстовым, видеофайлам и т. п.

Давайте на примере изменим ассоциацию у аудиофайла. Для этого перейдите в каталог с аудиофайлом и щелкните правой кнопкой мыши по файлу того типа, ассоциацию которого вы хотите сменить. В контекстном меню выберите команду **Свойства** и перейдите на вкладку **Открыть с помощью** появившегося окна (рис. 10.9).

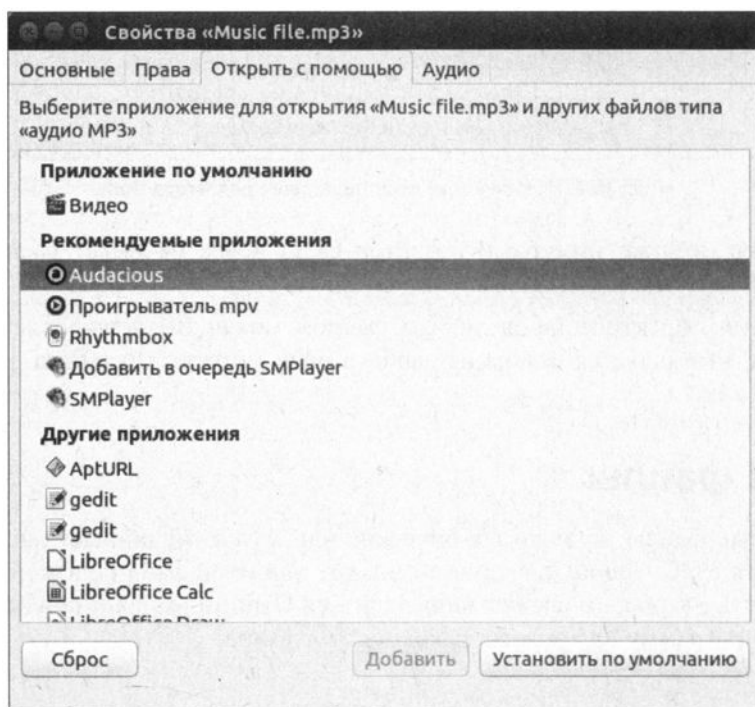


Рис. 10.9. Смена ассоциации аудиофайла

В этом окне выделите то приложение, с помощью которого вы хотите проигрывать файл данного типа, и нажмите кнопку **Установить по умолчанию**. Выбранная программа будет установлена по умолчанию для данного типа файла (рис. 10.10).

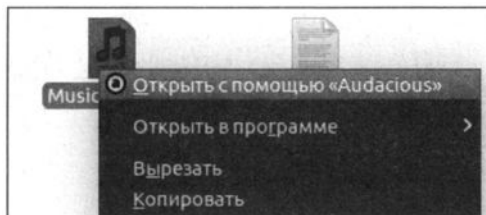


Рис. 10.10. Выбранная программа теперь ассоциируется с данным типом файла

Если вы хотите сбросить сохраненную ассоциацию файлов, нажмите кнопку **Сброс**, и проделанные вами изменения будут отменены.

Кнопка **Добавить** добавляет выбранный пункт в раздел рекомендуемых приложений.

В окне смены ассоциации файла перечисляются лишь те программы, которые уже установлены в ОС. Если вы хотите выбрать другое приложение, но не знаете, с чего начать поиск, щелкните правой кнопкой мыши по требуемому файлу и в контекстном меню выберите пункт **Открыть в программе**, а затем **Другое приложение....** Откроется окно, в котором можно либо просмотреть все установленные приложения, либо найти приложения в Менеджере приложений Ubuntu.

Выводы

Мы познакомились с правилами именования файлов в операционной системе Ubuntu Linux, а также узнали о том, что эта операционная система является регистронезависимой, что означает возможность создания нескольких файлов с одинаковыми названиями, но в разном регистре.

Помимо этого мы узнали о типах файлов и их расширениях и о том, как ОС Ubuntu Linux определяет тип файла.

ГЛАВА 11



Администратор и суперпользователь root

Привилегированный пользователь root

В любой операционной системе семейства Linux всегда присутствует привилегированная учетная запись `root`. Ее часто называют просто *суперпользователь root*. И это на самом деле так. Скоро вы поймете, почему именно суперпользователь.

Данная учетная запись сравнима с учетной записью **Администратор** в ОС Windows, только у учетной записи `root` намного больше привилегий в системе, нежели чем у аналогичной учетной записи в ОС Windows.

Эта учетная запись имеет неограниченные привилегии в операционной системе, в том числе на простые операции типа удаления системных файлов и изменения параметров системы. При неосторожном обращении пользователь, наделенный правами `root`, может полностью уничтожить операционную систему, восстановить работоспособность которой будет довольно проблематично.

По этой и многим другим причинам в дистрибутивах на основе Debian вход в учетную запись `root` был заблокирован. Это означает, что вы не можете зарегистрироваться в системе, используя логин `root` и его пароль. Наверняка при установке ОС Ubuntu вы обратили внимание, что программа-установщик ОС не запросила у нас создание пароля для этой учетной записи. Учетная запись `root` просто-напросто отключена, т. е. она присутствует в системе, но вы не можете в нее попасть, потому что для этого ее нужно предварительно включить.

"Зачем такие сложности, — спросите вы, — ведь можно было не отключать учетную запись `root`, а дать возможность с ней работать как с обычной учетной записью?" На самом деле, работать за учетной записью `root` — большой соблазн, который может закончиться плачевно. Человек всегда пытается облегчить себе жизнь, и если дать ему полные и постоянные полномочия (привилегии) в операционной системе, то он так и будет продолжать работать с этими расширенными привилегиями, которые зачастую ему просто не нужны.

Согласитесь, ведь нам не так часто приходится править конфигурационные файлы операционной системы, устанавливать программы или производить подобные дей-

ствия. Такая необходимость возникает не часто, пусть даже раз в неделю, а то и реже. Да, раз в неделю можно зайти под учетной записью root, произвести все нужные действия и зайти под обычной пользовательской записью, но тут появляется соблазн продолжить работу под учетной записью root, а на компьютерах с важными данными это губительно. Почему? Представьте ситуацию, когда пользователь поработал под учетной записью root и не вышел из системы, а через несколько минут его отвлекли, и за его компьютер сел человек, который захотел похитить важные данные с компьютера или уничтожить систему. Злоумышленнику будет достаточно выполнить пару простых команд, и все данные окажутся уничтоженными. А все потому, что пользователь забыл выйти из-под учетной записи root.

Теперь вам понятен смысл названия "*суперпользователь root*"?

Разработчики операционной системы Ubuntu приняли правильное решение, отключив учетную запись суперпользователя. Стоп, а как же тогда устанавливать программы и редактировать необходимые системные файлы, если эта учетная запись отключена, а наша учетная запись наделена обычными правами? Хороший вопрос, и сейчас мы постараемся на него ответить.

Обычная учетная запись и запись администратора

Как и ОС Windows, операционная система Ubuntu Linux позволяет создавать учетные записи двух типов:

- ◆ учетная запись администратора;
- ◆ обычная учетная запись.

Давайте сначала поговорим про *обычную учетную запись*. Наверняка вы обращали внимание, что в ОС Windows можно создать учетную запись без каких-либо привилегий? Это так называемая *гостевая учетная запись*. Так вот, ОС Ubuntu позволяет создавать похожую учетную запись, которая практически не наделена никакими полномочиями. Данное решение вполне разумно, когда за одним компьютером работает несколько человек, и некоторым из них нежелательно или не нужно давать возможность внесения изменения в работу ОС. Именно для этих целей предназначена обычная учетная запись, работая под которой, можно не бояться нанести тот или иной ущерб системе. По сути, это гостевая запись, лишенная всяческих привилегий и предназначенная для неопытных пользователей.

Теперь настало время разобраться с административной учетной записью, т. е. поговорить о том, что такое *учетная запись администратора*. Данная учетная запись создается еще на этапе установки операционной системы. Помните, мы создавали имя пользователя и указывали пароль учетной записи при установке ОС Ubuntu (рис. 11.1)?

Вспомнили? А теперь вспомните, от какой учетной записи мы вводили пароль, когда ОС запрашивала нас при вводе команды `sudo` в Терминале. Правильно! Мы вводили пароль именно от этой учетной записи, которую создали еще на этапе установки операционной системы.

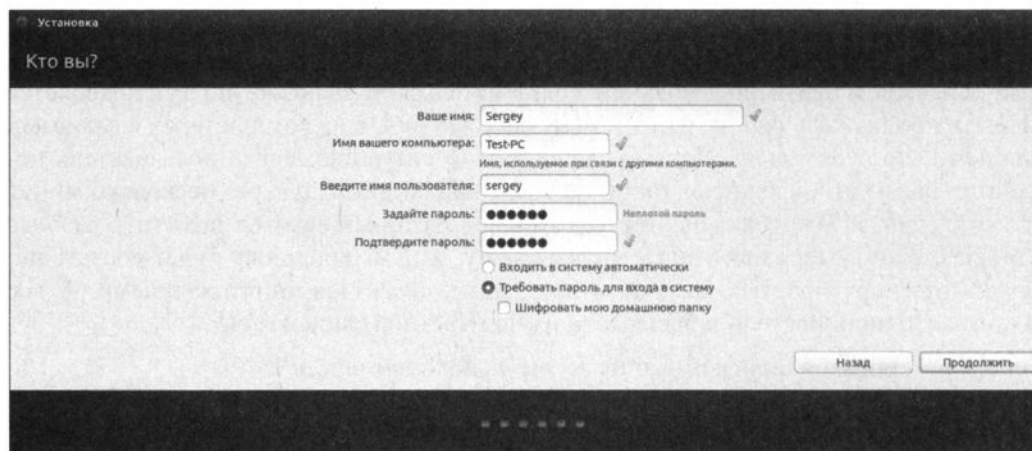


Рис. 11.1. Создание учетной записи администратора

Это и есть учетная запись администратора. Однако она не наделена такими же правами, как учетная запись суперпользователя — root. Главным отличием этой учетной записи является то, что для осуществления какого-либо вмешательства в работу ОС, например установку или удаление программ, от вас потребуются ввод пароля, что свидетельствует о потенциально опасном действии, т. е. о вмешательстве в работу самой операционной системы.

Учетная запись администратора может напоминать обычную учетную запись, однако последняя при попытке вмешаться в работу ОС просто-напросто отклонит эту попытку, а при работе с административной учетной записью ОС попросит ввести ваш пароль (рис. 11.2).

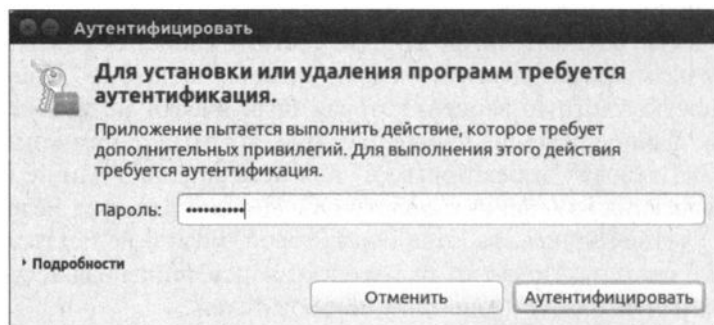


Рис. 11.2. ОС запрашивает пароль учетной записи администратора

Таким образом, при работе с административной учетной записью можно не беспокоиться о том, что кто-то может выполнить от вашего имени потенциально опасные действия и злонамеренно нарушить работу вашей ОС, т. к. ввод пароля будет запрашиваться каждый раз при выполнении таких манипуляций.

В случае если вы ошибочно введете пароль, система сообщит вам об этом (рис. 11.3), а если пароль был введен неправильно три раза, то окно аутентификации будет закрыто.

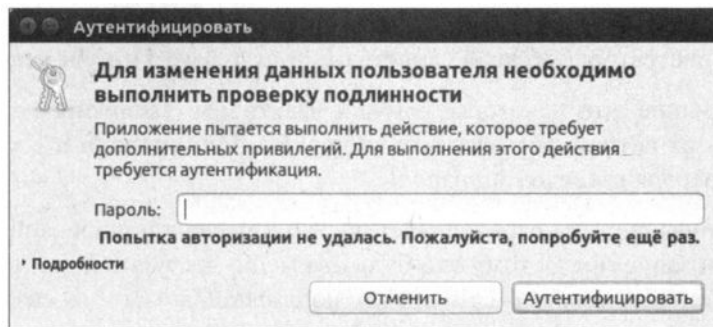


Рис. 11.3. Неправильный ввод пароля

Пугаться этого не стоит, потому что вы можете повторить попытку ввода пароля.

Теперь все встало на свои места, и мы знаем, что когда нам нужно внести изменения в работу ОС, то от нас потребуется ввод пароля административной учетной записи.

Привилегии администратора

Давайте на конкретном примере разберемся, для чего нам нужны привилегии администратора.

Попробуем добавить нового пользователя в систему. Для этого нажмите на значок системного меню в области уведомлений и выберите пункт меню **Параметры системы...** Откроется меню параметров системы, в котором найдите пункт **Учетные записи** и щелкните по нему. Откроется окно управления учетными записями (рис. 11.4).

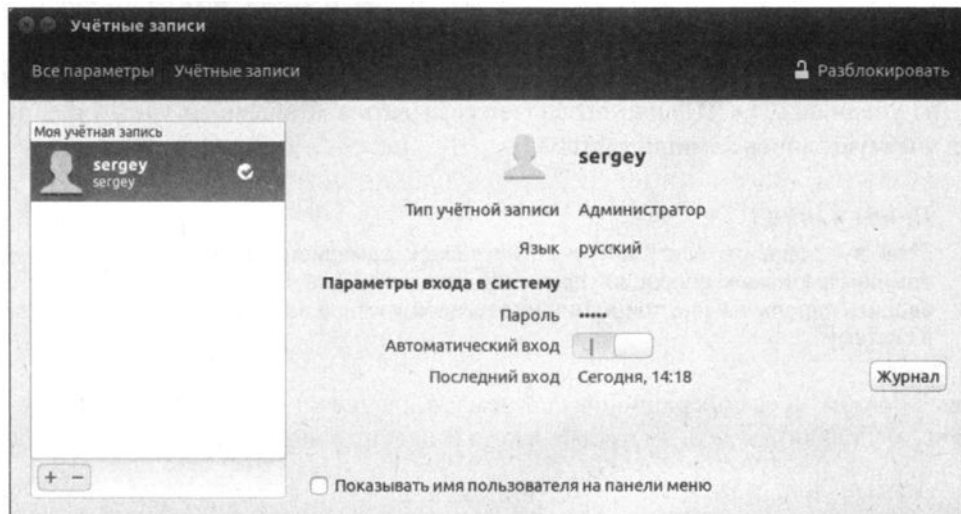


Рис. 11.4. Управление учетными записями

В этом окне мы видим, что у нас создана одна учетная запись, которая наделена правами администратора. Об этом свидетельствует строка **Тип учётной записи**.

Обратите внимание, что некоторые кнопки неактивны (заблокированы). Для того чтобы сделать их активными, нажмите кнопку **Разблокировать** и в уже знакомом окне запроса пароля введите ваш пароль.

Так операционная система ограничивает доступ к несанкционированному управлению учетными записями, потому что будь это меню доступным для изменения без аутентификации пользователя, то любой пользователь мог бы создать нужную учетную запись и уже в новой учетной записи нанести урон системе.

Обратите внимание, что после того как вы разблокировали возможность управления учетными записями, кнопки управления в этом окне стали активными.

Давайте посмотрим, какие типы учетных записей можно создать в ОС Ubuntu. Для этого в левой части окна нажмите кнопку с изображением символа "плюс" и в открывшемся окне добавления учетной записи щелкните на раскрывающемся списке **Тип учётной записи** (рис. 11.5).

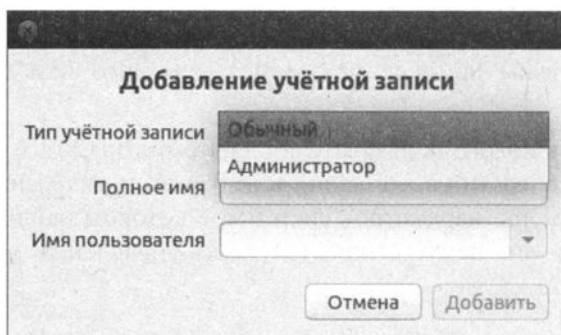


Рис. 11.5. Окно добавления учетной записи

Как мы убедились, ОС Ubuntu позволяет создавать как обычную учетную запись, так и учетную запись администратора.

ПРИМЕЧАНИЕ

Если вы добавите еще одну учетную запись администратора, то при выполнении административных операций, например при установке программ, вам нужно будет вводить пароль именно той административной учетной записи, под которой вы вошли в систему.

Таким образом, когда операционная система запрашивает от вас административный пароль, выполняются административные задачи в операционной системе Ubuntu.

Административные задачи в текстовом режиме

С тем, как выполнять административные задачи в графическом режиме, мы разобрались, но как быть, когда нужно выполнить другие административные задачи в текстовом режиме, т. е. в Терминале?

Для того чтобы выполнять команды, требующие более высоких привилегий, в ОС Ubuntu существует утилита `sudo`, которая предназначена специально для таких целей. С ней мы уже немного познакомились, когда запускали команды, требующие прав администратора.

ПРИМЕЧАНИЕ

Более детально команда `sudo` будет рассмотрена чуть позже, когда речь пойдет о работе в Терминале.

`Sudo` — это утилита, предоставляющая административные привилегии (привилегии `root`), когда это необходимо, т. е. по первому запросу пользователя.

При выполнении команды `sudo` пользователю, выполнившему ее, доступны те же полномочия, что и пользователю `root`.

Не забывайте о том, что первый пользователь, учетную запись которого вы создали при установке операционной системы, обладает теми же полномочиями, что и пользователь `root`.

Выводы

Мы рассмотрели несколько запутанную тему, связанную с учетными записями в операционной системе Ubuntu Linux. Мы узнали о том, что в данной ОС учетная запись пользователя `root` отключена по умолчанию, и вместо нее используется команда `sudo`, позволяющая выполнять операции, которые требуют административных полномочий.

Помимо этого мы узнали, что кроме административных учетных записей ОС Ubuntu Linux позволяет создавать и обычные, гостевые учетные записи. Благодаря гостевым учетным записям пользователь не сможет нарушить работоспособность операционной системы.

ГЛАВА 12



Знакомство с Консолью и Терминалом

Что такое Терминал и Консоль

До этого момента нам не редко приходилось открывать Терминал и выполнять какие-либо команды, прежде не сталкиваясь с ним.

В этой главе речь пойдет именно о работе с Терминалом и Консолью, и мы наконец-то поближе познакомимся с этим мощным инструментом, с которым должен уметь работать любой пользователь ОС Ubuntu Linux.

Начинающие пользователи Ubuntu Linux часто путают Терминал с Консолью, считая, что оба названия равнозначны. Настало время разобраться в этих понятиях, а также поближе познакомиться, без преувеличения сказать, с основным инструментом работы в ОС Ubuntu Linux.

Терминал — это название программы в ОС Ubuntu Linux, которая позволяет обрабатывать направляемые этой программе команды и направлять их соответствующим приложениям.

Терминал выступает неким посредником между пользователем и операционной системой. Запуская Терминал, мы получаем доступ к Консоли текущей либо удаленной машины. Таким образом, сам Терминал является надстройкой Консоли. Другими словами, Терминал лишь эмулирует работу Консоли.

Консоль — это программное обеспечение, организующее взаимодействие человека и компьютера. По своей сути, является устройством ввода-вывода.

Так, при запуске ОС Ubuntu Linux запускается сразу 7 виртуальных консолей, одна из которых стартует в графическом режиме. По умолчанию загружается сразу седьмая консоль, т.е. то, что мы привыкли называть графическим режимом. Остальные 6 виртуальных консолей запускаются в текстовом режиме.

ПРИМЕЧАНИЕ

Про виртуальные консоли в ОС Ubuntu Linux мы поговорим несколько позже.

Таким образом, Консоль является "основой" для Терминала, который наделен дополнительным функционалом, присущим для работы в графическом режиме.

Консоль и Терминал обрабатывают вводимые пользователем команды с помощью программной оболочки. В ОС Ubuntu по умолчанию используется командная оболочка `bash`, которая распознает команды в привычном для нее языке `bash`.

Пользователь вправе изменить стандартную командную оболочку на любую другую, но при этом стоит учитывать особенности новой оболочки, такие как различное написание типичных команд и т. п.

Запуск Консоли

Консоль можно запустить прямо из графического режима, нажав одно из сочетаний клавиш:

- ◆ `<Ctrl>+<Alt>+<F1>` — для запуска первой виртуальной консоли;
- ◆ `<Ctrl>+<Alt>+<F2>` — для запуска второй виртуальной консоли;
- ◆ `<Ctrl>+<Alt>+<F3>` — для запуска третьей виртуальной консоли;
- ◆ `<Ctrl>+<Alt>+<F4>` — для запуска четвертой виртуальной консоли;
- ◆ `<Ctrl>+<Alt>+<F5>` — для запуска пятой виртуальной консоли;
- ◆ `<Ctrl>+<Alt>+<F6>` — для запуска шестой виртуальной консоли;
- ◆ `<Ctrl>+<Alt>+<F7>` — для запуска седьмой виртуальной консоли и возврата в графический режим.

По нажатию этих клавиш вызовется соответствующая консоль (рис. 12.1).

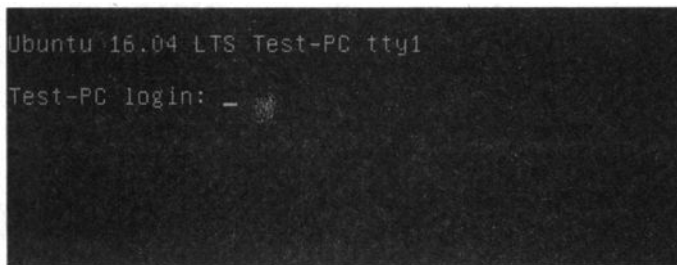


Рис. 12.1. Запущена первая виртуальная консоль `tty1`

В нашем случае была вызвана первая виртуальная консоль `tty1` (сочетание клавиш `<Ctrl>+<Alt>+<F1>`). При запуске виртуальной консоли от вас требуется ввести свой логин и пароль. Введите свой логин, в моем случае это `sergey`, т. е. имя, которое было задано при создании учетной записи еще на этапе установки операционной системы. После этого, Консоль запросит ввод пароля. Введите его от учетной записи, под именем которой вы пытаетесь зарегистрироваться.

ПРИМЕЧАНИЕ

При вводе пароля на экране не отображаются никакие символы, даже символ звездочки (*). Это нормальная ситуация, и данное решение позволяет скрыть от посторонних глаз не только сам пароль, но и его длину.

После ввода пароля появится приветствие (рис. 12.2).

Для возврата в графический режим нажмите сочетание клавиш `<Ctrl>+<Alt>+<F7>`, но не забудьте перед этим выйти из Консоли, выполнив команду `exit`.

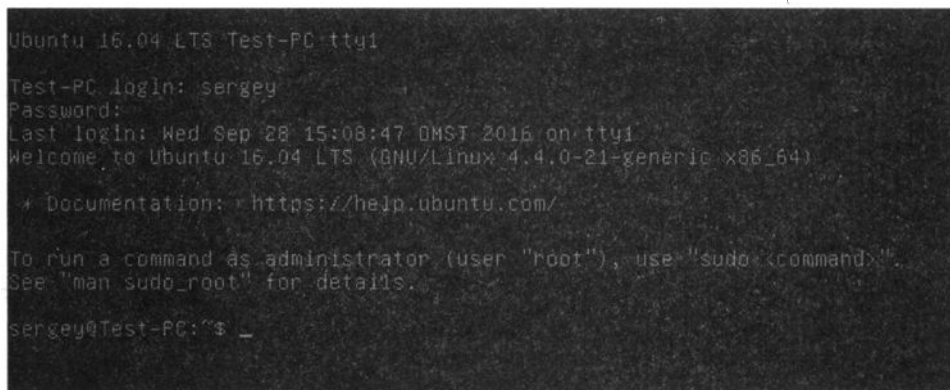


Рис. 12.2. Приветствие Консоли

ПРИМЕЧАНИЕ

Всегда выходите из Консоли, когда вы ею не пользуетесь, т. к. при запущенной Консоли доступны все те команды, выполнив которые злоумышленник может нанести значительный урон как самой операционной системе, так и данным, хранящимся на вашем компьютере.

Довольно часто у пользователей возникает вопрос: как осуществлять постраничную прокрутку текстовой информации в Консоли? Прокрутка выполняется довольно просто: нажмите сочетание клавиш `<Shift>+<Page Up>` для прокрутки страницы вверх или `<Shift>+<Page Down>`, чтобы пролистать вывод Консоли вниз.

Более детально узнать обо всех командах, доступных в Консоли, можно, выполнив команду `help`.

Запуск Терминала

Запустить Терминал можно одним из следующих способов:

- ◆ нажать сочетание клавиш `<Ctrl>+<Alt>+<T>`;
- ◆ нажать кнопку главного меню, в строке поиска ввести слово `Терминал` и запустить требуемую программу.

Первый способ наиболее быстрый и удобный, и, вероятно, он придется вам по душе. Запустите Терминал понравившимся вам способом, и вы увидите стандартное окно с текстом приветствия (рис. 12.3).

Текст приветствия выглядит следующим образом:

```
sergey@Test-PC:~$
```

Давайте разберем этот текст приветствия на составляющие (табл. 12.1).

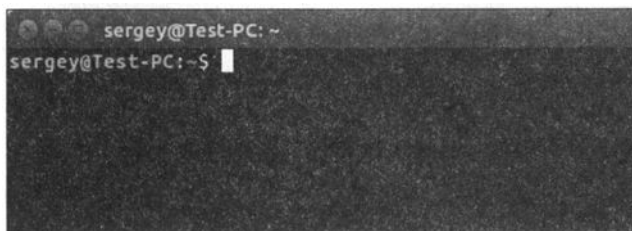


Рис. 12.3. Открыто окно Терминала

Таблица 12.1. Описание составляющих текста приветствия

Параметр	Описание
sergey	Имя учетной записи пользователя
@	Символ разделителя
Test-PC	Имя компьютера
:	Символ разделителя
~	Каталог выполнения команды. В нашем случае команды выполняются в домашнем каталоге
\$	Приглашение к выполнению команды с правами простого пользователя

Перед тем как перейти к дальнейшему рассмотрению Терминала, приведем список наиболее часто используемых "горячих" клавиш этой программы (табл. 12.2).

Таблица 12.2. Список "горячих" клавиш Терминала

"Горячие" клавиши	Описание
<Ctrl>+<Shift>+<C>	Копирование в буфера обмена выделенного текста
<Ctrl>+<Shift>+<V>	Вставка текста в Терминал из буфера обмена
<↑> либо <Ctrl>+<P>	Прокрутка недавно вводившихся команд вверх
<↓> либо <Ctrl>+<N>	Прокрутка недавно вводившихся команд вниз
<Ctrl>+<A>	Перемещение курсора в начало строки
<Ctrl>+<E>	Перемещение курсора в конец строки
<Ctrl>+<U>	Удаление текущей строки
<Ctrl>+<K>	Удаление текста с текущей позиции и до конца строки
<Ctrl>+<W>	Удаление слова, расположенного перед курсором
<Ctrl>+<R>	Поиск ранее введенных команд
<Ctrl>+<C>	Посылает сигнал завершения режима ввода
<Ctrl>+<D>	Закрывает текущий Терминал

Это краткий список "горячих" клавиш, которыми вам придется пользоваться довольно часто.

Синтаксис команд в Терминале

Команды Терминала, как правило, состоят из названия программы, ключа и значения и в общем виде выглядят так:

название_программы [-ключ] [значение]

- ◆ *название_программы* — это имя исполняемого файла из каталогов, записанных в переменную `$PATH` (`/bin`, `/sbin`, `/usr/bin`, `/usr/sbin`, `/usr/local/bin`, `/usr/local/sbin` и др.);
- ◆ [-ключ] — опции программы, которые может принимать выполняемая программа;
- ◆ [значение] — данный параметр может принимать в качестве аргумента цифры, текст, специальные символы и даже переменные.

Типичным примером данного синтаксиса может служить команда `ls -l ~`, которая отображает содержимое домашнего каталога пользователя.

Выполнение команд осуществляется по нажатию клавиши <Enter>. Сами команды могут быть введены вручную либо быть скопированными, но в любом случае выполнение команд осуществляется по нажатию клавиши <Enter>.

ПРИМЕЧАНИЕ

ОС Ubuntu Linux чувствительна к регистру, и команда `ls -l ~` не будет равносильна команде `Ls -l ~`. При попытке выполнения команды в неправильном регистре будет выдано сообщение о том, что команда не найдена.

Команда *echo*

Практически в каждой книге по программированию изучение языка начинают с фразы "Hello, World!". Эта фраза уже практически стала стандартной во всех пособиях по языкам программирования. Давайте и мы не будем отступать от традиции и "заставим" Терминал поприветствовать мир. Однако для начала стоит сказать, что в командном интерпретаторе `bash` для вывода какой-либо строки предусмотрена команда `echo` (от англ. *echo* — эхо).

Итак, для того чтобы "заставить" Терминал "сказать нужную фразу", выполните следующую команду:

```
echo 'Hello, World!'
```

В ответ на это Терминал выведет строку "Hello, World!".

Команда `echo` не несет никакой функциональной нагрузки, но бывает полезной для системных администраторов, которые автоматизируют свою работу с помощью сценариев командного интерпретатора `bash`.

Просмотр версии ядра

Часто возникают ситуации, когда нужно узнать текущую версию ядра ОС Ubuntu. Для этих целей предназначена следующая команда:

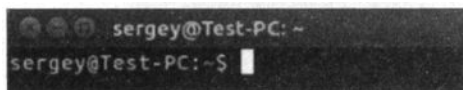
```
uname -r
```

Данная команда выведет номер выпуска операционной системы. Если запустить команду `uname` с ключом `-a`, то будет выведена вся доступная информация об используемой операционной системе.

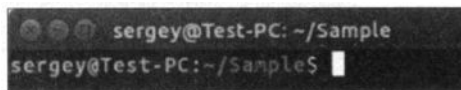
Установка текущего каталога

При запуске Терминала текущим является домашний каталог пользователя, обозначаемый тильдой (~). Разумеется, вы вправе сменить домашний каталог на какой-нибудь другой.

Узнать, в каком каталоге вы сейчас находитесь, очень просто: достаточно посмотреть на приглашение Терминала, т. е. на то, что находится между символом разделителя `:` и символом приглашения к вводу команды `$` (рис. 12.4).



а



б

Рис. 12.4. Текущая директория для разных сеансов Терминала:

а — домашний каталог — `~/` (`/home/sergey/`); б — домашний каталог — `~/Sample/` (`/home/sergey/Sample/`)

Как видим, на рис. 12.4 показано два текущих каталога: в первом случае текущим каталогом является домашний каталог пользователя (символ ~), а во втором — каталог `Sample`, который находится в том же домашнем каталоге пользователя.

Для того чтобы сменить текущий каталог на другой, служит команда `cd` (от англ. *change directory* — изменить каталог). Для смены текущего каталога выполните следующую команду:

```
cd /home/sergey/myfolder/
```

ПРИМЕЧАНИЕ

Чтобы не писать полный путь к домашнему каталогу пользователя, можно прибегнуть к короткому варианту: `cd ~/myfolder/`.

Сейчас возникнет резонный вопрос: как быть, если в названии каталога используется символ пробела или другие символы?

Давайте на практике убедимся, как на пробел отреагирует Терминал, выполнив команду `cd ~/new folder/` (рис. 12.5).

Для того чтобы не возникало таких ситуаций, когда Терминал не распознает каталоги, в имени которых есть специальные символы, разработчики предусмотрели так называемое экранирование специальных символов.

A terminal window with a dark background. The prompt is 'sergey@Test-PC: ~'. The user enters 'cd ~/new folder/'. The terminal output is 'bash: cd: /home/sergey/new: Нет такого файла или каталога'. The prompt returns to 'sergey@Test-PC: ~\$'.

Рис. 12.5. Терминал "ругается" на символы пробела в пути

Экранирование — это постановка символа обратного слеша \, в пути к файлу или каталогу.

Так, чтобы Терминал правильно отреагировал на специальные символы, давайте выполним экранирование пробелов в имени каталога, чтобы получилась следующая команда:

```
cd ~/new\ folder/
```

Теперь Терминал правильно обработал специальные символы, т. е. символ пробела, и сменил на тот, который мы указали (рис. 12.6).

A terminal window with a dark background. The prompt is 'sergey@Test-PC: ~/new folder'. The user enters 'cd ~/new folder/'. The terminal output is 'bash: cd: /home/sergey/new: Нет такого файла или каталога'. The user then enters 'cd ~/new\ folder/'. The terminal output is 'sergey@Test-PC: ~/new folder\$'. The prompt has changed to reflect the current directory.

Рис. 12.6. Терминал корректно обработал экранированный символ пробела

Как видите, Терминал больше не "ругается" на наличие пробела в имени каталога.

Для того чтобы вернуться в домашний каталог, совсем не обязательно набирать путь, достаточно просто выполнить команду `cd` без аргументов.

Вернуться в ранее посещенный каталог можно, выполнив команду:

```
cd -
```

Чтобы переместиться на каталог выше, достаточно передать команде `cd` две точки:

```
cd ..
```

Просмотр текущего каталога

При отображении Терминалом большого объема информации некоторые пользователи начинают путаться и теряться и забывают, в каком каталоге они сейчас находятся. Листать вывод Терминала не всегда удобно, но разработчики позаботились об этом моменте и предусмотрели команду `pwd` (от англ. *present working directory* — текущий рабочий каталог).

Выполните эту команду, и Терминал сообщит, в каком каталоге вы сейчас находитесь.

Просмотр содержимого каталога

Для просмотра содержимого каталога предназначена команда `ls` (от англ. *list* — список).

Синтаксис команды следующий:

```
ls [ключи]... [файл]...
```

Если выполнить команду `ls` без передачи ей каких-либо параметров, то будет выведен список содержимого текущего каталога (рис. 12.7).

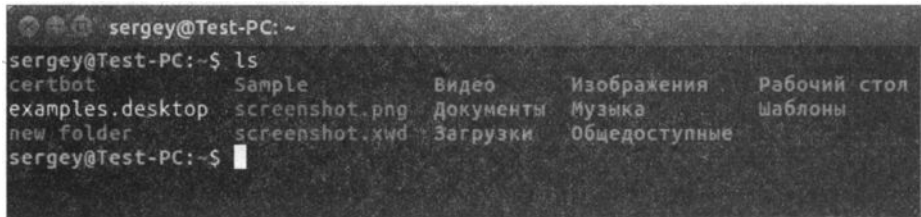


Рис. 12.7. Список содержимого текущей директории

Однако если передать команде параметр `-l`, то будет выведен подробный отчет о содержимом каталога.

Ключ `-l` позволяет отобразить имена только тех каталогов, которые не являются системными или скрытыми. Для того чтобы вывести все файлы и каталоги, включая скрытые, следует добавить ключ `a`. Команда должна получиться следующей: `ls -la` (рис. 12.8).

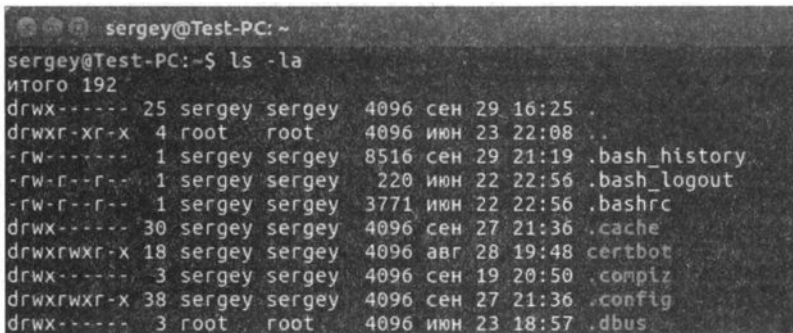


Рис. 12.8. Подробный вывод содержимого директории, включая скрытые и системные файлы

По умолчанию результат выводится в девять колонок. Сейчас разберем, что обозначают эти непонятные "надписи" на примере первой строки (табл. 12.3).

Для того чтобы просмотреть содержимое какого-то определенного каталога, выполните следующую команду:

```
ls -al /etc/
```

Таблица 12.3. Описание содержимого вывода команды `ls -la`

Номер столбца	Содержимое столбца	Описание
1	drwx-----	Права доступа на файл. В нашем случае права доступа на каталог, т. к. первый символ d обозначает, что перед нами каталог
2	25	Количество жестких ссылок в каталоге
3	sergey	Имя пользователя, являющегося владельцем файла
4	sergey	Группа пользователя, которой принадлежит файл или каталог
5	4096	Размер файла. В случае если перед нами каталог, то значение 4096, т. е. 4 Кбайт, равно блоку памяти, выделяемому для работы с данными
6	сен	Месяц последнего изменения файла
7	29	День последнего изменения файла
8	16:26	Время последнего изменения файла
9	.	Имя файла. В нашем случае точка означает ссылку на текущий каталог

Однако на этом не заканчиваются возможности команды `ls`. Полный список параметров данной команды можно узнать, выполнив команду `man ls`. По ее выполнению, вам будет представлен весь функционал программы `ls`.

Просмотр справки по командам

Теперь, когда мы поближе познакомимся с Терминалом, стоит сказать несколько слов о справочной системе команд этой программы, так называемым "мануалом". "Мануалом" часто любят пугать неопытных пользователей в русскоязычном сегменте Интернета. Сам "мануал" — это всего лишь справочная система по той или иной команде Терминала. Само слово берет начало от англ. *manual*, что в переводе означает "руководство".

В Терминале для удобства набора команд это слово было сокращено до трех букв — `man`.

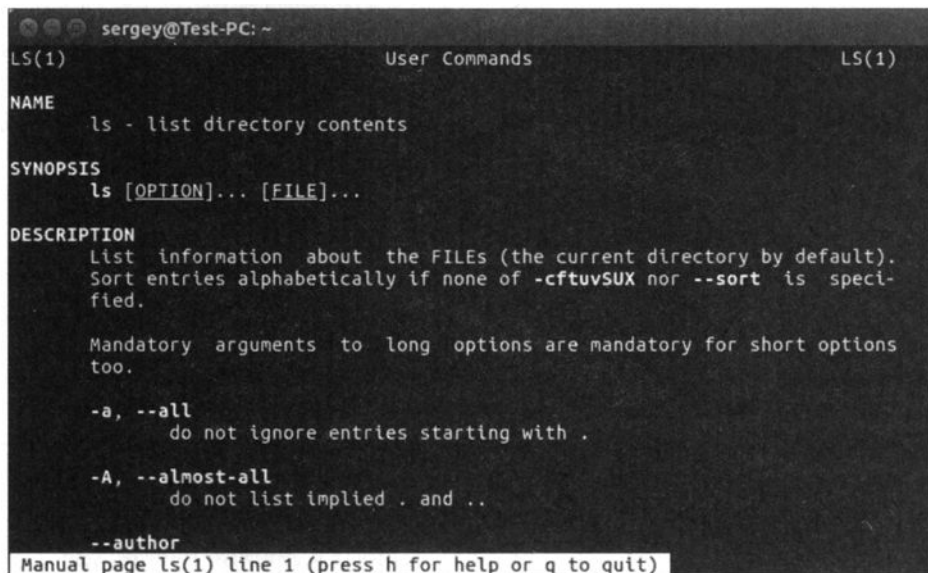
Синтаксис команды следующий:

```
man [имя_команды]
```

Параметр `[имя_команды]` принимает имя самой команды утилиты или функции, справочную информацию о которой вам необходимо получить.

Для того чтобы выполнить запрос справки по нужной команде, например по команде `ls`, выполните в Терминале следующую команду:

```
man ls
```



```
sergey@Test-PC: ~
LS(1)                                User Commands                                LS(1)

NAME
  ls - list directory contents

SYNOPSIS
  ls [OPTION]... [FILE]...

DESCRIPTION
  List information about the FILES (the current directory by default).
  Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
  fied.

  Mandatory arguments to long options are mandatory for short options
  too.

  -a, --all
        do not ignore entries starting with .

  -A, --almost-all
        do not list implied . and ..

  --author

Manual page ls(1) line 1 (press h for help or q to quit)
```

Рис. 12.9. Вызов справки по команде `ls`

В ответ на это вам будет представлен справочный материал о запрашиваемой команде (рис. 12.9).

Справочная информация по той или иной команде в большинстве случаев выдается на русском языке. При отсутствии локализованной версии справки вам будет предложена англоязычная версия.

Для постраничной навигации в справочной системе служат клавиши `<↑>` и `<↓>`. Также можно использовать клавиши `<Page Up>` для прокрутки страницы вверх или `<Page Down>` для прокрутки страницы вниз.

Справочная система имеет возможность постраничной прокрутки текста колесиком мыши, и этот способ просмотра справочных сведений покажется вам наиболее удобным после перехода в мир Ubuntu Linux после работы с ОС Windows.

Для того чтобы выйти из справочной системы и продолжить пользоваться Терминалом, необходимо нажать клавишу `<Q>`. При этом работа Терминала не будет прекращена. Если же нажать на значок закрытия окна Терминала, то появится предупреждение о том, что в Терминале запущен процесс и его закрытие приведет к его принудительному завершению.

Для получения более детальной информации по использованию справочной системы предназначена следующая команда:

```
man man
```

Да, вы не ошиблись: для справочной системы существует своя справочная система. Вот такая тавтология.

Помимо команды `man`, у многих утилит есть встроенная справочная система, которая вызывается следующим образом:

```
[имя_программы] --help
```

Давайте запустим встроенную справку для программы `ls`:

```
ls --help
```

На этот раз запустится встроенная в программу `ls` справочная система, которая уже будет локализована.

Подсказки Терминала

Мозг человека имеет свойство забывать ту или иную информацию, и часто это "свойство" мозга проявляется в самый неподходящий момент. Например, вам срочно необходимо установить какую-либо программу, но вы забыли, как она правильно пишется, либо она слишком длинная и вам намного проще ее откуда-то скопировать, чем печатать ее имя вручную. Что же делать в этом случае? Разработчики как всегда позаботились о нас и предусмотрели возможность так называемого автодополнения, другими словами — подсказки Терминала.

Продemonстрируем возможности автодополнения на практике. Начните вводить в Терминале символы `apt-g` и нажмите клавишу `<Tab>`. По нажатию клавиши `<Tab>` Терминал автоматически дополнит набранное слово, точнее, команду.

Это одна ситуация. А что, если вы не помните полное название команды, но помните только, что она начинается со слова `apt`. Ну что ж, введите в Терминале слово `apt` и дважды нажмите клавишу `<Tab>` (рис. 12.10).

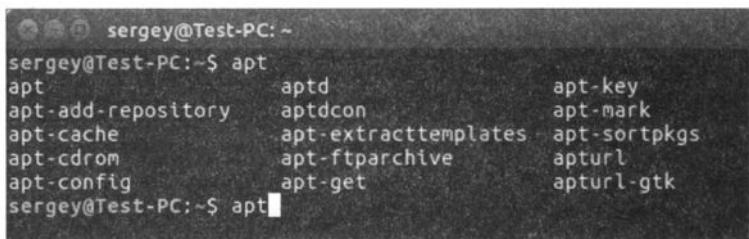


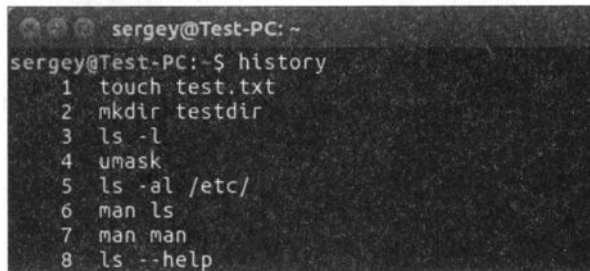
Рис. 12.10. Терминал вывел список всех команд, начинающихся с искомого слова

Как видите, Терминал вместо автодополнения вывел список всех команд, которые начинаются с искомого слова. Просмотрев этот список, вы без труда найдете именно ту команду, которая вам требуется. Очень полезная функциональность Терминала, не так ли?

История введенных ранее команд

Терминал имеет очень полезную функцию сохранения ранее введенных команд в файл `.bash_history`, который расположен в домашнем каталоге пользовательской учетной записи. Таким образом, когда загружается Терминал, то вместе с ним в память загружается и сам файл `.bash_history`.

По умолчанию в этот файл записываются до 1000 введенных ранее команд.



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ history  
1 touch test.txt  
2 mkdir testdir  
3 ls -l  
4 umask  
5 ls -al /etc/  
6 man ls  
7 man man  
8 ls --help
```

Рис. 12.11. Вывод списка ранее введенных команд

Для просмотра ранее введенных команд служит команда `history`, которая выведет список ранее выполненных команд (рис. 12.11).

Таким образом, вы можете просмотреть введенные ранее команды, а также повторно выполнить уже отработанную команду, введя только ее номер. Так, если вы хотите выполнить команду под номером 5, то введите в Терминале символ восклицательного знака и номер команды:

```
!5
```

Будет выполнена команда `ls -al /etc/`, и вам не потребуется полностью вводить ее.

Еще полезным способом поиска ранее введенных команд является нажатие сочетания клавиш `<Ctrl>+<R>`. Нажмите их, начните вводить искомую команду, и Терминал автоматически будет подставлять варианты, исходя из тех команд, что были введены вами ранее.

Очистка окна Терминала

Очень часто после выполнения нескольких команд окно Терминала становится слишком переполненным уже неактуальной информацией, и неопытные пользователи просто закрывают и потом заново открывают Терминал, чтобы очистить его окно. Однако этот радикальный способ не является самым удобным, уже не говоря о его банальности. Как раз для таких целей, когда требуется очистить окно Терминала, не прибегая к его закрытию, предназначена команда `clear` (от англ. *clear* — очистить). Выполните ее, и окно Терминала будет очищено от вывода предыдущих команд.

Остановка выполнения команды

Случаются ситуации, когда нужно немедленно прервать выполнение команды, не дожидаясь ее завершения.

Например, вами по невнимательности была запущена команда принудительного удаления содержимого какого-либо каталога, и вы в определенный момент начинаете осознавать, что нужно немедленно прервать выполнение этой команды.

Разумеется, отключать питание компьютера — не самый удачный вариант, поэтому намного удобнее и быстрее нажать сочетание клавиш `<Ctrl>+<C>`.

Нажав это сочетание клавиш, вы прервете выполнение операции. Конечно же, нажатие клавиш `<Ctrl>+<C>` уже не возвратит удаленные файлы, но, тем не менее, позволит хотя бы сохранить те из них, которые попадали под удаление по невнимательности.

Выход из Терминала

Пользуясь компьютером на работе или в каком-либо другом месте, всегда следует помнить об информационной безопасности. По этой причине не стоит оставлять ни Консоль, ни Терминал запущенными. Помните, что оставив без присмотра какую-либо запущенную программу от имени администратора, вы рискуете потерять контроль над вашей операционной системой, т. к. проходящий мимо вашего компьютера посторонний человек может выполнить любую команду, которая способна повлечь непредсказуемые последствия, например удаление критически важных документов. Согласитесь, такой инцидент доставит немало проблем, а в системных журналах все изменения будут числиться под вашей учетной записью, и вам будет проблематично доказать, что в тот или иной момент времени вы отсутствовали за компьютером.

Для того чтобы никто не воспользовался Терминалом или Консолью от имени вашей учетной записи без вашего ведома, всегда по окончании работы выполняйте команду `exit` для выхода из Терминала или Консоли.

Поверьте, лучше лишний раз ввести команду `exit`, чем потом нести ответственность за чьи-то недобросовестные действия, выполненные на вашем компьютере в ваше отсутствие.

Также для закрытия Терминала можно воспользоваться "горячими" клавишами: `<Alt>+<F4>` либо `<Ctrl>+<D>`.

Выводы

Мы познакомились с Консолью и Терминалом, навыки использования которых просто необходимы для полноценного использования операционной системы Ubuntu Linux.

Мы познакомились с основными "горячими" клавишами, которые используются в Консоли и Терминале, а также узнали, что их применение может быть полезным в случае ошибочного выполнения команд либо просто для удобства выполнения операций в Терминале.

Мы на практике освоили основные команды по управлению файлами и каталогами, а также научились запрашивать справочную информацию по нужной нам команде.

Полученные знания и навыки мы будем применять на протяжении всего процесса изучения, а также использования ОС Ubuntu Linux.

ГЛАВА 13



Полезные команды Терминала

О пользе работы в Терминале

Возможности Терминала практически неограниченны. Все упирается только в ваши практические навыки работы в этой программе.

Дело в том, что в ОС Ubuntu Linux вовсе не обязательно должна присутствовать графическая оболочка. Вся работа в операционной системе может сводиться лишь к использованию Консоли.

ПРИМЕЧАНИЕ

Как мы помним, Терминал является лишь своеобразной надстройкой над Консолью.

Все команды, которые выподняются в Терминале, будут также работать и в Консоли. Именно поэтому знание основных команд Терминала вам может пригодиться в дальнейшем. Например, в случае возникновения каких-либо непредвиденных проблем в графической среде операционной системы у вас всегда будет возможность восстановить функционирование системы в командной строке, т. е. в Консоли. При этом вам не придется прибегать к переустановке операционной системы, как это было бы в случае использования ОС Windows.

Это лишь только верхушка тех возможностей, что может вам предоставить знание команд Терминала. Однако на практике вам, скорее всего, не придется использовать все возможности, предоставляемые нам Терминалом, поэтому мы рассмотрим лишь самые необходимые команды, которые с большей долей вероятности вам пригодятся для повседневного использования ОС Ubuntu Linux.

Команда *whoami*, или "Кто я?"

Команда *whoami* (от англ. *who am i* — кто я) предназначена для получения сведений об имени пользователя, под учетной записью которого выполняется данная команда.

Выполнив команду *whoami*, вы получите имя пользователя. На этом функционал программы заканчивается.

Команда *who*

Команда *who* (от англ. *who* — кто) применяется для вывода более подробной информации о пользователях, которые вошли в систему.

Синтаксис команды *who* следующий:

who [*ключ*]... [*файл* | *аргумент1* *аргумент2*]

Параметр [*ключ*] принимает следующие значения (табл. 13.1).

Таблица 13.1. Параметры команды *who*

Сокращенный вариант	Полный вариант	Описание
-a	--all	Эквивалент выполнения команды с ключами: -b -d -l -p -r -t -T -u
-b	--boot	Выводит время последней загрузки системы
-d	--dead	Выводит информацию обо всех мертвых процессах, которые были завершены и не были заново открыты процессом <i>init</i> . У мертвых процессов есть поле <i>EXIT</i> , в котором перечисляются значения системного и пользовательского кодов завершения и могут оказаться полезными для выяснения причин аварийного завершения этого процесса
-H	--heading	Выводит запрашиваемую строку с заголовками столбцов
	--ips	Выводит имя пользователя, номер виртуальной консоли, дату и время запроса, имя терминальной линии
-l	--login	Выводит информацию о процессе входа в систему
	--lookup	Выводит ту же самую информацию, что и параметр <i>--ips</i> , только при этом происходит попытка разрешения имен через DNS
-m		Выводит информацию об имени узла и пользователе, которые связаны со стандартным вводом
-p	--process	Выводит информацию об активных процессах, которые были вызваны процессом <i>init</i>
-q	--count	Выводит информацию обо всех именах и количестве подключенных пользователей
-r	--runlevel	Выводит информацию о текущем уровне выполнения
-s	--short	Выводит краткую информацию: имя пользователя, имя терминальной линии и времени начала сеанса
-t	--time	Выводит информацию о последнем изменении системного времени
-T, -w	--mesg	Выводит информацию о состоянии терминальной линии. В случае если терминальной линии можно передавать сообщения, то статус приема сообщений на эту терминальную линию будет помечен знаком +, в противном случае линия будет помечена знаком -. Пользователь <i>root</i> может передавать сообщения во все линии, независимо от знака + или -. При обнаружении неисправностей терминальная линия будет помечена знаком ?

Таблица 13.1 (окончание)

Сокращенный вариант	Полный вариант	Описание
-u	--users	Выводит информацию о подключенных пользователях
	--message, --writable	Эквиваленты ключа -T
	--help	Выводит справочную информацию
	--version	Выводит информацию о версии

Обычно формат вывода информации имеет следующий вид:

имя [*состояние*] *терминал* *время* [*ожидание*] [*pid*] [*комментарий*] [*статус_выхода*]

В общем виде данный формат обозначает следующее (табл. 13.2).

Таблица 13.2. Описание формата вывода команды *who*

Формат вывода	Описание
<i>имя</i>	Регистрационное имя пользователя
[<i>состояние</i>]	Возможность записи на терминальную линию
<i>терминал</i>	Имя терминальной линии
<i>время</i>	Время, прошедшее с момента регистрации пользователя в системе
[<i>ожидание</i>]	Время, прошедшее с момента выполнения пользователем последнего действия
[<i>pid</i>]	Идентификатор процесса начального командного интерпретатора пользователя
[<i>комментарий</i>]	Строка комментария из файла <i>/etc/inittab</i>
[<i>статус_выхода</i>]	Статус выхода для мертвых процессов

По умолчанию информация об имени пользователя, терминальной линии, астрономическом времени начала сеанса, продолжительности бездействия терминальной линии с момента последнего обмена и идентификатор процесса интерпретатора команд (в нашем случае Терминала) берется из файла */etc/utmp*.

В случае если команде *who*, помимо параметров, передать имя файла в параметре [*файл* | *аргумент1* *аргумент2*], например файл */etc/wtmp*, то можно узнать время начала всех сеансов с момента последнего создания самого файла.

Создание файлов

Теперь, когда мы уже столько знаем о Терминале, вам не покажется странным, что в ОС Ubuntu Linux создавать и удалять файлы принято в Терминале. Разумеется, вы можете создать файл в графической оболочке, но гораздо полезнее знать, как это делается средствами командного интерпретатора *bash*.

Итак, для создания файла, в ОС Ubuntu Linux существует команда `touch` (от англ. *touch* — прикоснуться, прикасаться), которая в то же время наделена функциями установки времени последнего изменения файла.

Прежде всего команда `touch` предназначена для установки времени последнего изменения файла или времени последнего доступа к нему. Если файл с заданным именем не существует, то команда `touch` создает файл с указанным именем и устанавливает время последнего изменения и последнего доступа, равным значению, которое было передано в качестве аргумента. В том случае если аргумент времени не задан, то будет использоваться текущее время.

Синтаксис команды следующий:

```
touch [ключ]... файл...
```

Параметр `[ключ]` принимает значения, перечисленные в табл. 13.3.

Таблица 13.3. Параметры команды `touch`

Сокращенный вариант	Полный вариант	Описание
-a		Позволяет изменять только время доступа к файлу
-c	--no-create	Позволяет не создавать файл, если он существует
-d	--date=строка	Позволяет проанализировать переданную в качестве аргумента строку и использовать ее вместо текущего времени
-f		Игнорируется командным интерпретатором. Ранее использование этого параметра позволяло обновить информацию о времени, даже когда права доступа к файлу запрещали это
-h	--no-dereference	Позволяет изменять символьные ссылки, а не сами файлы, на которые они ссылаются
-m		Позволяет устанавливать только время последнего изменения файла
-r	--reference=файл	Позволяет использовать время файла, переданного в ключе <code>--reference=файл</code>
-t время		<p>Позволяет установить время последнего изменения и доступа в соответствии с форматом <i>время</i>. Формат времени, переданный в аргументе <i>время</i>, имеет следующий шаблон:</p> <pre>[[BB] ГГ] ММ ДД чч мм [. cc]</pre> <p>где BB — первые две цифры года (век); ГГ — последние две цифры года; ММ — двузначный номер месяца; ДД — двузначный номер дня; чч — значение часов даты; мм — значение минут даты; cc — значение секунд даты.</p> <p>Если параметр BB не задан и значение ГГ находится в пределах 69 и 99, тогда BB устанавливается равным 19, иначе будет использоваться значение 20</p>

Таблица 13.3 (окончание)

Сокращенный вариант	Полный вариант	Описание
	<code>--time=слово</code>	Позволяет задать время, заданное с переданным форматом времени в аргументе <i>слово</i> . Формат времени принимает следующие значения: <code>access</code> , <code>atime</code> , <code>use</code> , <code>mtime</code> , <code>modify</code>
	<code>--help</code>	Выводит справочную информацию
	<code>--version</code>	Выводит информацию о версии

Давайте создадим файл `myfile.txt` с помощью команды:

```
touch myfile.txt
```

По выполнении данной команды в домашнем каталоге вашей учетной записи будет создан соответствующий файл, если до этого он не существовал. В случае существования заданного файла команда только обновит время последнего изменения и доступа к файлу, а его содержимое останется нетронутым.

Теперь в качестве примера установим нашему файлу `myfile.txt` дату последнего изменения (параметр `modify`), равную *05:15:27 5 ноября 2011 года*:

```
touch -t 201111050515.27 --time=modify myfile.txt
```

Теперь в свойствах нашего файла `myfile.txt` значится вот такая интересная дата (рис. 13.1).

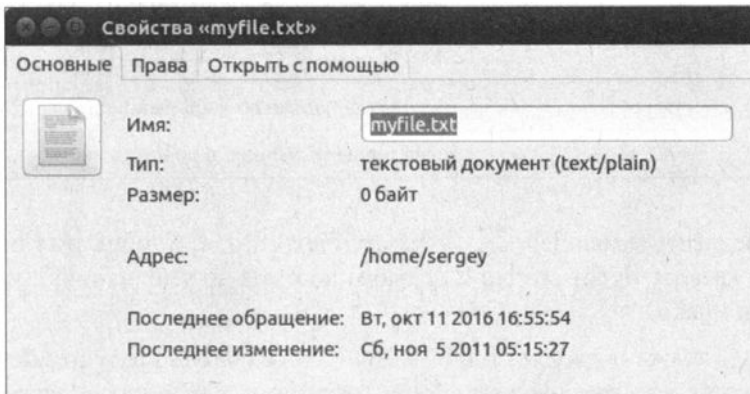


Рис. 13.1. Файл с модифицированной датой последнего изменения

Обратите внимание, что, применив параметр `--time` и передав ему значение `modify`, мы тем самым указали команде `touch`, что требуется изменить только дату последнего изменения файла, а не время последнего обращения к файлу и время его изменения.

Создание каталогов

Для создания каталога в ОС Ubuntu Linux служит команда `mkdir` (от англ. *make directory* — создать каталог). Синтаксис команды следующий:

```
mkdir [ключ]... каталог...
```

Параметр [ключ] принимает значения, перечисленные в табл. 13.4.

Таблица 13.4. Параметры команды `mkdir`

Сокращенный вариант	Полный вариант	Описание
<code>-m</code>	<code>--mode=режим</code>	Позволяет установить права доступа к файлу, как в команде <code>chmod</code> . При этом при установке прав не учитывается пользовательская маска, т. е. права устанавливаются без учета <code>umask</code>
<code>-p</code>	<code>--parents</code>	Позволяет не выводить ошибку, если каталог существует, а также создавать недостающие родительские каталоги для каждого указанного каталога
<code>-v</code>	<code>--verbose</code>	Указывает на необходимость вывода сообщения о каждом созданном каталоге, а также выводить сообщение в случае, если создаваемый каталог уже существует
<code>-Z</code>		Позволяет устанавливать контекст безопасности SELinux для каждого создаваемого каталога, равным типу по умолчанию
	<code>--context[=CTX]</code>	Аналогичен параметру <code>-Z</code> , но если указан CTX, то устанавливается контекст безопасности SELinux или SMACK равным CTX
	<code>--help</code>	Выводит справочную информацию
	<code>--version</code>	Выводит информацию о версии

Если не передавать команде `mkdir` никаких параметров, а лишь имя создаваемого каталога, то каталог будет создан с правами доступа по умолчанию с учетом пользовательской маски.

Команда `mkdir` также позволяет одной командой создавать сразу несколько каталогов, для этого просто перечислите имена создаваемых каталогов, разделяя их пробелами так, как это показано в примере:

```
mkdir folder1 folder2 folder3
```

После выполнения данной команды в текущем каталоге будут созданы три каталога: `folder1`, `folder2` и `folder3`.

При попытке создать каталог с уже существующим именем будет показано сообщение о невозможности выполнения данной операции.

ПРИМЕЧАНИЕ

Напомним, что в ОС Ubuntu Linux любой объект, в том числе и каталог, является *файлом*.

Однако если добавить параметр `-p`, то ошибок не будет выведено даже в том случае, когда каталог существует (рис. 13.2).

```

sergey@Test-PC: ~
sergey@Test-PC:~$ mkdir folder1 folder2 folder3
sergey@Test-PC:~$ mkdir folder3
mkdir: невозможно создать каталог «folder3»: Файл существует
sergey@Test-PC:~$ mkdir folder3 -p
sergey@Test-PC:~$
  
```

Рис. 13.2. Ключ `-p` команды `mkdir` подавил вывод ошибок

Как видим, добавление параметра `-p`, "подавило" вывод ошибок. При использовании этого параметра стоит учитывать, что каталог все равно создан не будет, потому что уже существует.

Удаление каталогов

Создавать каталоги с помощью Терминала мы уже научились, но нужно еще и уметь их удалять. Для удаления пустых каталогов предназначена команда `rmdir`, синтаксис которой выглядит следующим образом:

`rmdir [ключ]... каталог...`

Параметр `[ключ]` принимает значения, перечисленные в табл. 13.5.

Таблица 13.5. Параметры команды `rmdir`

Сокращенный вариант	Полный вариант	Описание
	<code>--ignore-fail-on-non-empty</code>	Позволяет проигнорировать все ошибки, которые будут возникать при удалении непустых каталогов
<code>-p</code>	<code>--parents</code>	Позволяет удалить каталог и его потомков, т. е. все вложенные каталоги в этом каталоге
<code>-v</code>	<code>--verbose</code>	Позволяет выводить сообщение для каждого обработанного каталога
	<code>--help</code>	Выводит справочную информацию
	<code>--version</code>	Выводит информацию о версии

Перечисленные ключи являются необязательными, т. к. команда может принимать в качестве аргумента только имя удаляемого каталога.

Данная команда идеально подходит в том случае, если необходимо удалить один или несколько пустых каталогов. Однако она не сработает, если один или несколько из удаляемых каталогов не пусты.

Для удаления непустых каталогов предназначена команда `rm`, о которой сейчас и пойдет речь.

Удаление каталогов и файлов

Для удаления каталогов существует опасная команда `rm` (от англ. *remove* — удалить). Чем же она так опасна, мы узнаем чуть позже, а пока взглянем на ее синтаксис:

```
rm [ключ]... [файл]...
```

Параметр `[ключ]` принимает значения, перечисленные в табл. 13.6.

Таблица 13.6. Параметры команды `rm`

Сокращенный вариант	Полный вариант	Описание
<code>-f</code>	<code>--force</code>	Позволяет игнорировать ошибки выполнения и не запрашивать подтверждения выполнения операций
<code>-i</code>		Позволяет выводить запрос на подтверждение перед каждым удалением
<code>-I</code>		Установка данного параметра позволяет выдавать запрос на подтверждение в том случае, когда удаляется сразу более 3 файлов или перед рекурсивным удалением
	<code>--interactive[=<i>когда</i>]</code>	Установка данного параметра позволяет выводить запрос на подтверждение в соответствии с переданным аргументом ключу <code>--interactive</code> . Передаваемые аргументы могут быть следующими: <code>never</code> — никогда; <code>once</code> — один раз; <code>always</code> — всегда. Если передаваемый аргумент не задан, подтверждение будет выводиться перед каждой операцией удаления
	<code>--one-file-system</code>	Установка данного параметра позволяет при рекурсивном удалении иерархии пропускать все те каталоги, которые находятся на другой файловой системе, что и соответствующий аргумент командной строки
	<code>--no-preserve-root</code>	Позволяет не обрабатывать корневую файловую систему особым образом
	<code>--preserve-root</code>	Предотвращает удаление корневой файловой системы

Таблица 13.6 (окончание)

Сокращенный вариант	Полный вариант	Описание
-r, -R	--recursive	Позволяет рекурсивно удалять каталоги и их содержимое
-d	--dir	Позволяет задать удаление пустых каталогов
-v	--verbose	Позволяет сопровождать все производимые действия соответствующими пояснениями
	--help	Выводит справочную информацию
	--version	Выводит информацию о версии

Вероятно, вы сейчас кинулись выполнять операцию удаления каталога следующей командой:

```
rm folder3
```

Но как же так, команда ведь не работает и выдает сообщение о том, что невозможно удалить данный каталог?

Дело в том, что по умолчанию команда `rm` удаляет *не каталоги*, а только *файлы*. Для того чтобы удалить каталог, нужно команде `rm` передать ключ `-r`, который позволит рекурсивно удалить каталог и его содержимое.

А теперь узнаем, почему же команда `rm` иногда бывает так опасна.

Опасность выполнения команды `rm -rf /`

Опасность в работе с командой кроется лишь в неопытности пользователей, которые, выполняя команду, не знают, что именно она делает. Так, нередко случаи, когда пользователи запускали опасную команду `rm -rf /`, и им оставалось лишь наблюдать, как медленно, но верно, операционная система самоуничтожается. Да, именно самоуничтожается, потому что команда `rm -rf /` запускает процесс *удаления содержимого корневой файловой системы*. При выполнении этой команды не запрашивается никаких подтверждений на удаление благодаря ключу `-f`.

Очень часто данную команду предлагают выполнить тем неопытным пользователям, которые обращаются за советом на какой-либо форум. Причем не всегда это делается из злого умысла, а ради шутки. И таким "шутникам" вовсе нет дела до того, что кто-то лишается всей важной для него информации, а зачастую этот нерадостный опыт получают студенты, которые решили освоить новую для них операционную систему.

Да что там говорить, у каждого из нас на компьютере есть хоть какая-то часть информации, которая ценна для нас, и ее потеря крайне нежелательна.

Разумеется, такие шутники никогда не переведутся и будут также предлагать выполнить "команду самоуничтожения" операционной системы.

Примечательно то, что на такую "шутку" попадают не только неопытные пользователи, но и даже продвинутые. Так, на одном популярном форуме, посвященном ОС Linux, пользователь опубликовал сообщение, в котором он просил подсказать ему, почему его код не работает. Код выглядел следующим образом (листинг 13.1).

Листинг 13.1. Код удаления файловой системы

```
cat "test... test... test..." | perl -e '$??s;;s;s;;$?:;s;;=]>%-{<-|}<&|`
(;;y; -/:-@[~`{-);`-{" -;;s;;$_;see'
```

Не спешите его набирать у себя в Консоли, т. к. этот код на языке Perl полностью идентичен команде `rm -rf /`. Да, этот код непросто понять, и многие сразу же попытались запустить его, чем и поплатились по доброте потерей личных файлов.

После большого ажиотажа вокруг команды `rm -rf /` многие производители операционных системы стали реализовывать в своих продуктах защиту от самоуничтожения системы и для выполнения этой команды начали запрашиваться права root.

Позже разработчики UNIX-подобных операционных систем сделали так, что команда `rm -rf /` перестала выполняться без указания специального ключа. Таким образом, только при указании ключа `--no-preserve-root` применительно к команде `rm -rf /` система сможет выполнить самоуничтожение.

Если вы вдруг по невнимательности выполнили данную команду, то *как можно скорее* нажимайте сочетание клавиш `<Ctrl>+<C>`. Этим вы все же сможете прервать удаление важных файлов и хоть что-то спасти от удаления.

Копирование файлов и каталогов

Для копирования файлов и каталогов в ОС Linux предназначена команда `cp` (от англ. *copy* — копировать).

По умолчанию команда копирует только файлы, но если дополнительно указать соответствующий ключ, то будет произведено копирование каталогов. Команда позволяет копировать один файл в другой файл, а также несколько файлов в заданный каталог.

В общем виде команда `cp` имеет следующий синтаксис:

```
cp [ключ]... источник каталог_ (назначение)
```

Параметр `[ключ]` принимает значения, перечисленные в табл. 13.7.

Таблица 13.7. Параметры команды `cp`

Сокращенный вариант	Полный вариант	Описание
-a	--archive	Идентичен ключам <code>-dPR</code> и <code>--preserve=all</code>
	--attributes-only	Позволяет скопировать только атрибуты файла, при этом данные файла не копируются

Таблица 13.7 (продолжение)

Сокращенный вариант	Полный вариант	Описание
	<code>--backup [=когда]</code>	Позволяет создавать резервную копию каждого целевого файла. По умолчанию суффикс для запасных копий — <code>~</code> , если только не установлены переменная окружения <code>SIMPLE_BACKUP_SUFFIX</code> или ключ <code>--suffix</code> . Способ контроля версий может быть установлен при помощи ключа <code>--backup</code> или переменной окружения <code>VERSION_CONTROL</code> . Допустимые значения: <code>none</code> , <code>off</code> — никогда не создавать резервных копий, даже если указан ключ <code>--backup</code> ; <code>numbered</code> , <code>t</code> — создавать нумерованные копии; <code>existing</code> , <code>nil</code> — если существуют нумерованные копии, то создавать нумерованные, в противном случае создавать простые копии; <code>simple</code> , <code>never</code> — всегда создавать простые копии
<code>-b</code>		Равнозначен ключу <code>--backup</code> , но без аргумента
<code>-d</code>		Равнозначен ключам <code>--no-dereference</code> и <code>--preserve=link</code>
<code>-f</code>	<code>--force</code>	При невозможности открыть существующий файл будет произведена попытка удалить его и произведена повторная попытка его копирования. Этот параметр игнорируется, если также указан ключ <code>-n</code>
<code>-i</code>	<code>--interactive</code>	Позволяет выводить запрос перед тем, как переписывать данные. Этот параметр отменяет ранее указанный ключ <code>-n</code>
<code>-H</code>		Позволяет следовать по символьным ссылкам, указанным в источнике
<code>-l</code>	<code>--link</code>	Вместо копирования позволяет создавать жесткие ссылки
<code>-L</code>	<code>--dereference</code>	Использование данного параметра позволяет всегда следовать символьным ссылкам в источнике
<code>-n</code>	<code>--no-clobber</code>	Позволяет не перезаписывать существующий файл. При этом данный параметр отменяет ранее указанный ключ <code>-i</code>
<code>-P</code>	<code>--no-dereference</code>	Позволяет не следовать по символьным ссылкам, указанным в источнике
<code>-p</code>		Равнозначен ключу <code>--preserve=mode,ownership,timestamps</code>
	<code>--preserve [=список_атрибутов]</code>	Позволяет сохранять указанные атрибуты. По умолчанию атрибуты следующие: <code>mode,ownership,timestamps</code> . Если возможно, производится попытка сохранить дополнительные атрибуты: <code>context, links, xattr, all</code>

Таблица 13.7 (продолжение)

Сокращенный вариант	Полный вариант	Описание
	--parents	Позволяет добавить исходный путь к <i>каталогу_ (назначения)</i>
-R, -r	--recursive	Задаёт рекурсивное копирование каталогов
	--reflink[= <i>когда</i>]	Позволяет контролировать создаваемые копии. Если указан параметр --reflink[=always], выполняется облегченное копирование, при котором блоки данных копируются только при изменении. Если это невозможно или если указан параметр --reflink=auto, то используется обычное копирование
	--remove-destination	Позволяет удалять каждый файл назначения перед попыткой его открыть
	--sparse= <i>когда</i>	Позволяет управлять созданием разреженных файлов. С ключом --sparse=always всегда создается разреженный файл <i>каталога_ (назначения)</i> вне зависимости от того, содержит ли <i>источник</i> достаточно длинные последовательности нулевых байтов. Ключ --sparse=never используется для запрета создания разреженных файлов. По умолчанию используется значение --sparse=auto, что означает автоматическое создание разреженных файлов <i>каталога_ (назначения)</i>
	--strip-trailing-slashes	Позволяет удалять все конечные косые черты из каждого аргумента <i>источник</i>
-s	--symbolic-link	Позволяет создавать символичные ссылки вместо копирования
-S	--suffix= <i>суффикс</i>	Позволяет использовать для запасных копий заданный <i>суффикс</i>
	--target-directory= <i>каталог</i>	Позволяет скопировать все <i>источники</i> в <i>каталог_ (назначение)</i>
-T	--no-target-directory	Позволяет считать <i>каталог_ (назначение)</i> обычным файлом
-u	--update	Позволяет копировать только тогда, когда исходный файл новее, чем файл назначения, или когда файл назначения отсутствует
-v	--verbose	Позволяет сопровождать все производимые действия соответствующими пояснениями
-x	--one-file-system	Предписывает оставаться только в пределах одной файловой системы
-Z		Позволяет установить контекст безопасности SELinux

Таблица 13.7 (окончание)

Сокращенный вариант	Полный вариант	Описание
	<code>--context [=CTX]</code>	Аналогичен параметру <code>-Z</code> . Если указан CTX, то устанавливается контекст безопасности SELinux или SMACK равным CTX
	<code>--help</code>	Выводит справочную информацию
	<code>--version</code>	Выводит информацию о версии

Теперь на конкретных примерах научимся работать с командой `cp`. Для начала скопируем содержимое одного файла в другой. Пусть у нас будет файл `Test.txt` с записанными в него данными. Этот файл нужно скопировать в файл `Test1.txt`. Файл `Test1.txt` может даже отсутствовать на диске либо уже иметь какие-либо данные. В обоих случаях конечный файл будет содержать данные из исходного файла `Test.txt`.

Итак, выполните следующую команду:

```
cp Test.txt Text1.txt
```

В домашнем каталоге пользователя будет создан файл `Test1.txt`, который по содержанию будет полностью идентичен файлу `Test.txt`.

Все это замечательно, но в Терминале ничего не отображается, и сложно проконтролировать правильность выполнения этой команды. Как же быть в этой ситуации? Все просто! Добавьте к команде копирования ключ `-v`, и все ваши действия будут сопровождаться соответствующими пояснениями (рис. 13.3):

```
cp -v Test.txt Text1.txt
```

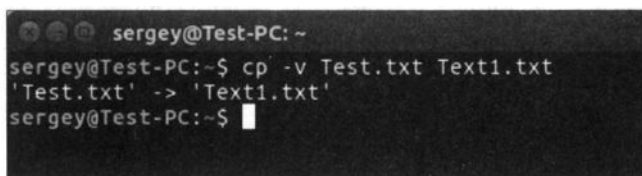


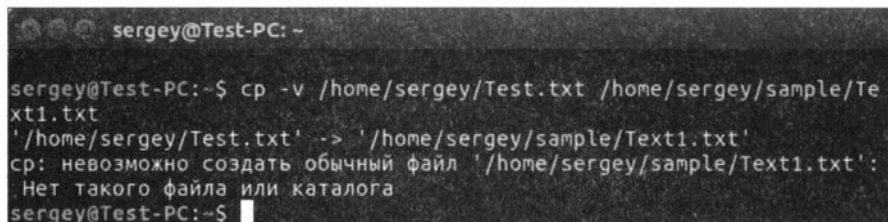
Рис. 13.3. При добавлении ключа `-v` все действия сопровождаются пояснениями

Конструкция `->` в выводе пояснения при выполнении команды означает, что содержимое файла `Test.txt` было скопировано в файл `Text1.txt`. С данной конструкцией мы встретимся несколько позже, а пока продолжим изучение работы команды `cp`.

Теперь нам необходимо скопировать наш файл `Test.txt` в каталог `Sample`, который также находится в домашнем каталоге. Давайте для наглядности пропишем полные пути к каталогам:

```
cp -v /home/sergey/Test.txt /home/sergey/Sample/Text1.txt
```

По выполнении данной команды в каталог Sample будет скопирован файл Text1.txt. Не забываем про регистр в именах файлов и каталогов. Если написать имя в другом регистре, то будет выведена ошибка, что невозможно создать файл, т. к. нет такого файла или каталога (рис. 13.4).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ cp -v /home/sergey/Test.txt /home/sergey/sample/Text1.txt  
cp: невозможно создать обычный файл '/home/sergey/sample/Text1.txt':  
Нет такого файла или каталога  
sergey@Test-PC:~$
```

Рис. 13.4. Чувствительность к регистру в названии каталога выдало ошибку копирования файла

Иногда бывают задачи, когда требуется скопировать сразу несколько файлов в другой каталог. Как быть в этой ситуации: копировать каждый файл отдельной командой или все же можно скопировать их с помощью одной строчки? Выход из этой ситуации довольно прост: необходимо перечислить все копируемые файлы, а затем указать каталог назначения:

```
cp -v /home/sergey/Test.txt /home/sergey/Test1.txt /home/sergey/Sample/
```

Копировать несколько файлов можно также и из разных каталогов: просто укажите полный путь к файлам, а в последнем параметре задайте каталог назначения копируемых файлов.

Сразу возникает логичный вопрос: а если нужно скопировать большое количество файлов из одного каталога в другой, то неужели придется перечислять все файлы? Нет, достаточно только указать в первом параметре символ `*` вместо файлов и каталог назначения:

```
cp -v /home/sergey/* /home/sergey/Sample/
```

После выполнения данной команды все файлы, находящиеся в домашнем каталоге, будут скопированы в каталог Sample.

Можно скопировать только часть файлов, например, лишь те, чьи имена начинаются с буквы `t`:

```
cp -v /home/sergey/t* /home/sergey/Sample/
```

Таким образом, файлы, чьи имена начинаются на другие буквы, скопированы не будут.

Это лишь основные операции, которые выполняет команда `cp`. Список возможностей команды достаточно большой, и сейчас нет смысла рассматривать каждый параметр, перечисленный в табл. 13.7, т. к. при необходимости вы сами сможете разобраться с ними.

Перемещение файлов и каталогов

Для перемещения файлов и каталогов средствами Терминала в ОС Linux существует команда `mv` (от англ. *move* — перемещать). Она позволяет перемещать любой файл или каталог в нужный каталог, а также производить переименование файлов и каталогов.

Общий синтаксис команды следующий:

```
mv [ключ]... источник... каталог
```

Параметр `[ключ]` принимает значения, перечисленные в табл. 13.8.

Таблица 13.8. Параметры команды `mv`

Сокращенный вариант	Полный вариант	Описание
	<code>--backup[=контроль]</code>	Позволяет создавать резервные копии целевых файлов
<code>-b</code>		Идентичен параметру <code>--backup</code> , но без аргумента
<code>-f</code>	<code>--force</code>	Позволяет переписывать существующие файлы, не спрашивая при этом подтверждения со стороны пользователя
<code>-i</code>	<code>--interactive</code>	Позволяет спрашивать подтверждение со стороны пользователя перед тем, как переписывать файл или каталог
<code>-n</code>	<code>--no-clobber</code>	Позволяет не переписывать существующий файл. Если указано более одного ключа <code>-i</code> , <code>-f</code> , <code>-n</code> , то будет использован только последний
	<code>--strip-trailing-slashes</code>	Позволяет удалить все конечные косые черты из каждого аргумента <i>источника</i>
<code>-S</code>	<code>--suffix=суффикс</code>	Позволяет перекрыть обычный суффикс для резервных копий
<code>-t</code>	<code>--target-directory=каталог</code>	Позволяет переместить все <i>источники</i> в <i>каталог</i>
<code>-T</code>	<code>--no-target-directory</code>	Позволяет считать параметр <i>каталог</i> обычным файлом
<code>-u</code>	<code>--update</code>	Позволяет перемещать, только если <i>источник</i> новее, чем файл назначения, или когда файл назначения отсутствует
<code>-v</code>	<code>--verbose</code>	Позволяет сопровождать все производимые действия соответствующими пояснениями
<code>-Z</code>	<code>--context</code>	Позволяет задать контекст безопасности SELinux файлу назначения как у типа по умолчанию
	<code>--help</code>	Выводит справочную информацию
	<code>--version</code>	Выводит информацию о версии

Наверняка сейчас вы задались вопросом: почему же в ОС Ubuntu Linux переименование и перемещение совмещены в одной команде? Дело в том, что ОС Ubuntu Linux считает переименование и перемещение эквивалентными операциями. Если вы переименовываете файл, то операционная система трактует это действие как перенос файла с новым именем на новое место.

В качестве примера рассмотрим переименование файла. Итак, у нас имеется файл `Test.txt`. Нам нужно его переименовать в `new_file.txt`. Для этого выполним следующую команду:

```
mv Test.txt new_file.txt
```

Как вы уже поняли из примера, первым аргументом команды `mv` идет старое имя файла, вторым аргументом вводится новое имя файла, которое мы хотим присвоить нашему файлу `Test.txt`.

Теперь нам потребовалось переместить файл `new_file.txt` в каталог `Sample`, который пока находится в домашнем каталоге. Для этого нужно выполнить следующую команду:

```
mv -v new_file.txt /home/sergey/Sample/
```

Таким образом, наш файл будет перемещен в каталог `Sample`. Обратите внимание, что для визуального контроля над перемещением мы к команде перемещения добавили параметр `-v`.

Воспользуемся еще одним интересным приемом — одновременным перемещением и переименованием файла:

```
mv -v /home/sergey/Sample/new_file.txt rename.txt
```

Здесь мы перемещаем файл `new_file.txt` из каталога `/home/sergey/Sample/` в домашний каталог, при этом сразу переименовываем его в `rename.txt`. Все очень просто и понятно.

Рассмотрим еще один интересный пример перемещения каталогов. Для начала выполним следующую команду:

```
mv -v /home/sergey/Sample/ /home/sergey/sample/
```

На первый взгляд все может показаться очень непонятным, но это только на первый взгляд.

В этой команде мы переименовываем каталог `Sample` в `sample`. Согласитесь, не всегда удобно каждый раз вводить название каталога с заглавной буквы, ведь иногда можно ошибиться. Пусть лучше имя каталога для удобства будет в нижнем регистре.

Так как в домашнем каталоге не существует каталога `sample`, то перемещение не происходит, а выполняется лишь переименование конечного каталога. Ну что ж, с перемещением и переименованием мы разобрались.

Теперь переместим все файлы с расширением `txt` из домашнего каталога в каталог `sample`:

```
mv -v *.txt /home/sergey/sample/
```

После выполнения команды все файлы, имеющие указанное расширение, будут перемещены в каталог `/home/sergey/sample/`. Подобным образом производится перемещение файлов с разными расширениями, которые перечисляются в первом аргументе через пробел.

Вывод содержимого файла

Для вывода на экран содержимого файла предназначена команда `cat` (от англ. *concatenate* — слияние). Данная команда позволяет отправить содержимое файла на стандартный вывод, т. е. на монитор, а также создавать файлы. Для начала посмотрим на синтаксис команды:

```
cat [ключ]... [файл]...
```

Параметр `[ключ]` принимает значения, перечисленные в табл. 13.9.

Таблица 13.9. Параметры команды `cat`

Сокращенный вариант	Полный вариант	Описание
-A	--show-all	Аналогичен ключу -vET
-b	--number-nonblank	Позволяет нумеровать непустые строки при выводе
-e		Подавляет указанный ключ -n. Аналогичен ключу -vE
-E	--show-ends	Позволяет показывать символ \$ в конце каждой строки
-n	--number	Позволяет нумеровать все строки при выводе
-s	--squeeze-blank	Позволяет выводить не более одной пустой строки при повторе
-t		Параметр аналогичен ключу -vT
-T	--show-tabs	Позволяет показывать символы табуляции в виде символов ^I
-u		В настоящее время данный параметр игнорируется
-v	--show-nonprinting	Позволяет использовать запись с символами ^ и M-, за исключением символов перевода строки и табуляции
	--help	Выводит справочную информацию
	--version	Выводит информацию о версии

Если команде `cat` не передавать никаких параметров или просто передать символ `-`, то команда будет копировать стандартный ввод на стандартный вывод. Это означает, что все, что будет вводиться в Терминале или Консоли, будет напоминать обычный ввод текста в текстовом редакторе. Даже команды будут восприняты как

обычный текст. Для того чтобы остановить ввод, нажмите сочетание клавиш `<Ctrl>+<C>` или `<Ctrl>+<D>`. Ввод текста будет отменен, и вы можете продолжить использование Терминала.

В качестве примера рассмотрим уже знакомый файл `myfile.txt`, находящийся в домашнем каталоге пользователя. Предположим, что файл имеет некоторый текст, и этот текст нам нужно вывести в Терминал. Для этого выполним такую команду:

```
cat myfile.txt
```

Терминал отобразит содержимое файла `myfile.txt` (рис. 13.5).

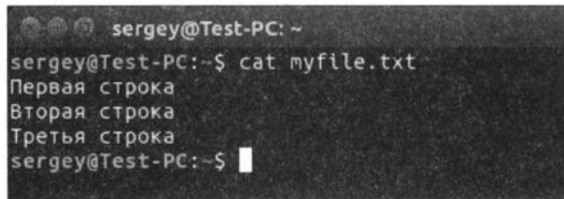


Рис. 13.5. Вывод содержимого файла в Терминал

Если нам требуется получить копию содержимого файла `myfile.txt`, то придется выполнить следующую команду:

```
cat myfile.txt > copy.txt
```

Обратите внимание, что первым аргументом указывается исходный файл, затем следует символ `>`, предписывающий перенаправить стандартный ввод в текстовый файл `copy.txt`, который автоматически будет создан командой `cat`.

Рассмотрим еще один интересный пример:

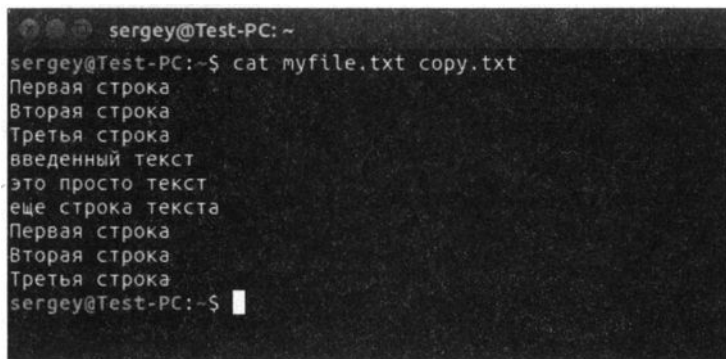
```
cat > copy.txt
```

Теперь попробуйте вводить различный текст в Терминале. Как видите, данная конструкция перенаправила стандартный ввод в Терминал, а вывод в файл `copy.txt`. Теперь, чтобы завершить работу нашей команды, нажмите сочетание клавиш `<Ctrl>+<C>` или `<Ctrl>+<D>`, и весь введенный вами текст сохранится в файле `copy.txt`.

С этим мы разобрались, а как быть с самим названием команды, ведь ее название звучит как *concatenate*, что означает слияние? Все верно, команда позволяет производить конкатенацию, т. е. слияние файлов. Для начала возьмем наши два файла: `myfile.txt` и `copy.txt`. В файл `copy.txt` записан текст, который мы вводили в примере, когда перенаправляли стандартный ввод на Терминал. Итак, для слияния содержимого двух файлов и отображения их в Терминале выполним следующую команду:

```
cat myfile.txt copy.txt
```

В качестве первого аргумента мы передаем команде `cat` первый файл `myfile.txt`, а вторым аргументом выступает файл `copy.txt`. По выполнении данной команды Терминал отобразит содержимое наших двух файлов (рис. 13.6).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ cat myfile.txt copy.txt  
Первая строка  
Вторая строка  
Третья строка  
введенный текст  
это просто текст  
еще строка текста  
Первая строка  
Вторая строка  
Третья строка  
sergey@Test-PC:~$
```

Рис. 13.6. Терминал отобразил содержимое файлов myfile.txt и copy.txt

Чтобы сохранить результат слияния (конкатенации) этих двух файлов в какой-либо файл, нужно выполнить следующую команду:

```
cat myfile.txt copy.txt > cat_file.txt
```

Содержимое двух файлов будет записано в файл cat_file.txt и именно в том порядке, в котором идет перечисление файлов.

Теперь рассмотрим пример добавления содержимого первого файла во второй:

```
cat myfile.txt >> copy.txt
```

Обратите внимание, что в команде используются уже две угловые скобки >> вместо одной. Данная конструкция позволяет добавить содержимое первого файла в конец файла, который передается вторым аргументом. Таким образом, содержимое файла myfile.txt было добавлено в конец файла copy.txt.

С перенаправлением ввода и конкатенацией все понятно. Теперь пронумеруем выводимые строки (ключ -n) из файла myfile.txt и добавим символ конца строки \$ (ключ -E):

```
cat -En myfile.txt
```

После выполнения этой команды каждая строка будет пронумерована и окончание каждой строки будет отмечено символом \$.

Вывод содержимого файла в обратной последовательности

В ОС Ubuntu Linux существует команда под названием tac, которая является практически идентичной команде cat. Команда tac делает все то же самое, что и команда cat, за исключением того, что tac выводит содержимое файла в обратной последовательности. Обратите внимание, она даже пишется наоборот.

Сразу возникают резонные вопросы: зачем вообще нужна эта "дублирующая" команда, и кому может понадобиться выводить строки из файла в обратной последовательности? На самом деле, эта команда очень важна, и ее часто приходится

использовать системным администраторам, когда они просматривают файлы журналов записей системных событий, а они, как нам известно, дописываются именно в конец. По этой причине было бы совершенно бессмысленным загружать файл с первой строчки, чтобы потом его приходилось "отматывать" в самый конец в поиске последнего события. Не проще ли сразу вывести содержимое такого файла событий в обратной последовательности? Логично, не так ли?

Именно для этих целей служит команда `tac`, которая имеет следующий синтаксис:

```
tac [ключ]... [файл]...
```

Параметр `[ключ]` принимает значения, перечисленные в табл. 13.10.

Таблица 13.10. Параметры команды `tac`

Сокращенный вариант	Полный вариант	Описание
<code>-b</code>	<code>--before</code>	Позволяет присоединять разделитель к началу, а не к концу строки
<code>-r</code>	<code>--regex</code>	Позволяет воспринимать разделитель как регулярное выражение
<code>-s</code>	<code>--separator=строка</code>	Позволяет использовать как разделитель строку, а не символ новой строки

В качестве примера выведем содержимое файла `myfile.txt` в обратном порядке:

```
tac myfile.txt
```

После выполнения команды строки из файла `myfile.txt` будут выведены в обратной последовательности (рис. 13.7).

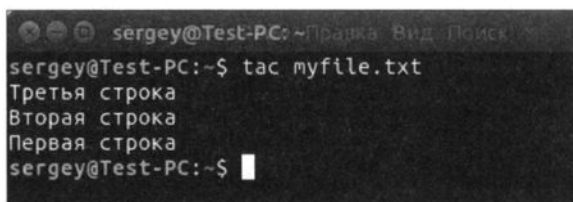


Рис. 13.7. Команда `tac` вывела строки из файла в обратной последовательности

В данном примере наш файл содержит всего три строки, а что если строк 30 или даже 300? Как раз для таких больших файлов существуют специальные команды страничного просмотра содержимого файлов. О них и пойдет речь далее.

Команды *more* и *less* для просмотра содержимого файлов

Изначально, для просмотра содержимого объемных файлов была разработана программа *more* (от англ. *more* — больше). Она обладает функциями вывода содержимого файла на экран отдельными страницами, т. е. по размеру окна.

Синтаксис этой команды прост:

```
more [опции] <файл>...
```

При передаче команде *more* имени файла она постранично выводит содержимое файла в окно Терминала. Чтобы увидеть следующую страницу, нужно нажать клавишу <Пробел>, а для прокрутки текста на одну строку служит клавиша <Enter>. Чтобы окончить просмотр файла, нужно нажать клавишу <Q> либо сочетание клавиш <Ctrl>+<C>.

Сейчас мы не будем детально рассматривать команду *more*, т. к. ей на смену пришла более функциональная команда, точнее, программа *less*.

ПРИМЕЧАНИЕ

С командой *more* вы можете более подробно ознакомиться, выполнив запрос справки командой *man more*.

Команда *less* (от англ. *less* — меньше) предназначена для просмотра содержимого текстовых файлов. В отличие от своего предшественника (программы *more*) команда *less* содержит огромный функционал. Она также "понимает" параметры команды *more*.

Сейчас мы рассмотрим весь функционал команды *less*, который довольно объемный в отличие от команды *more*.

Синтаксис команды *less* следующий:

```
less [ключ] [файл...]
```

Параметр [ключ] принимает значения, перечисленные в табл. 13.11.

Таблица 13.11. Параметры команды *less*

Ключи (короткий и/или полный вариант)	Описание
-a --search-skip-screen	Позволяет при поиске пропускать текущий экран (страницу)
-A --SEARCH-SKIP-SCREEN	Позволяет начать поиск с текущей позиции, т. е. с начала текущей страницы
-b [N] --buffers=[N]	Позволяет установить размер буфера для чтения. По умолчанию размер буфера равен 64 Кбайт
-B --auto-buffers	Позволяет отменить автоматическое выделение размера буфера для блоков стандартного вывода

Таблица 13.11 (продолжение)

Ключи (короткий и/или полный вариант)	Описание
-c --clear-screen	Позволяет очищать экран вместо прокрутки данных
-d --dumb	Позволяет создать дамп Терминала
-D [xn.n] --color=xn.n	Позволяет сменить цвет стандартного устройства вывода. Данный параметр поддерживается только если в качестве стандартного устройства вывода используется клиентская система MS-DOS
-e, -E --quit-at-eof --QUIT-AT-EOF	Позволяет выходить из программы при достижении конца файла
-f --force	Позволяет принудительно открывать нерегулярные файлы, т. е. файлы с нетекстовой информацией
-F --quit-if-one-screen	Позволяет выйти из программы в том случае, если содержимое файла умещается на первом экране
-g --hilite-search	Позволяет выделить только последний результат при поиске определенного значения
-G --HILITE-SEARCH	Предписывает не выделять совпадения, даже если таковые были найдены
-h [N] --max-back-scroll=[N]	Позволяет установить ограничения на обратную прокрутку, т. е. ограничить объем тех данных, которые будут доступны при обратной прокрутке
-i --ignore-case	Позволяет игнорировать регистр при поиске, даже если искомое значение не в верхнем регистре
-I --IGNORE-CASE	Позволяет полностью игнорировать регистр независимо от регистра искомой фразы
-j [N] --jump-target=[N]	Позволяет отобразить текущую позицию в окне вывода
-J --status-column	Позволяет отобразить столбец статуса слева от экрана вывода
-k [file] --lesskey-file=[файл]	Позволяет использовать специальный файл lesskey, который описывает привязку клавиш и команды, которые будут выполняться при их нажатии
-K --quit-on-intr	Позволяет использовать сочетание клавиш <Ctrl>+<C> для выхода из режима вывода
-L --no-lessopen	Позволяет игнорировать переменную окружения LESSOPEN
-m, -M --long-prompt --LONG-PROMPT	Позволяет отобразить информацию об объеме просмотренного в процентах

Таблица 13.11 (продолжение)

Ключи (короткий и/или полный вариант)	Описание
-n, -N --line-numbers --LINE-NUMBERS	Позволяет включить либо отключить нумерацию строк при выводе на экран
-o [файл] --log-file=[файл]	Позволяет скопировать вывод в log-файл при стандартном способе ввода
-O [файл] --LOG-FILE=[файл]	Позволяет скопировать вывод в log-файл, при этом прежнее содержимое файла будет перезаписано
-p [шаблон] --pattern=[шаблон]	Позволяет выполнить шаблон из командной строки
-P {prompt} --prompt={prompt}	Позволяет установить новый текст строки приветствия, которое отображается в нижней части экрана
-q, -Q --quiet --QUIET --silent --SILENT	Позволяет установить "тихий режим", при котором на стандартном устройстве вывода не будет отображено никаких ошибок
-r, -R --raw-control-chars --RAW-CONTROL-CHARS	Позволяет установить так называемый RAW-режим, при котором будут отображены все управляющие символы
-s --squeeze-blank-lines	Установка данного параметра позволяет опускать несколько пустых строк, идущих подряд
-S --chop-long-lines	Позволяет обрезать слишком длинные строки
-t [ter] --tag=[ter]	Позволяет найти нужный тег
-T [tagsfile] --tag-file=[tagsfile]	Позволяет использовать теги из указанного в параметре файла
-u, -U --underline-special --UNDERLINE-SPECIAL	Добавляет вывод специальных символов в местах окончания и переноса строк
-w --hilite-unread	Позволяет выделить первую строку на первом экране
-W --HILITE-UNREAD	Позволяет выделять первую строку после каждого пролистывания вперед
-x [N[,...]] --tabs=[N[,...]]	Позволяет установить табуляцию равной символам, указанным в передаваемом параметре
-X --no-init	Позволяет при выводе не использовать библиотеки и базы данных свойств терминала, а использовать дисплей компьютерного терминала аппаратно-независимо
-y [N] --max-forw-scroll=[N]	Позволяет установить лимит на прокрутку вперед через передаваемое в качестве параметра значение

Таблица 13.11 (окончание)

Ключи (короткий и/или полный вариант)	Описание
-z [N] --window=[N]	Позволяет установить размер окна через передаваемое в качестве параметра значение
-" [c[c]] --quotes=[c[c]]	Позволяет использовать имена файлов, в которых содержатся символы ". Кавычки экранируются символами "
-- --tilde	Отключает отображение тильды ~ в конце файла
-# [N] --shift=[N]	Позволяет установить количество горизонтальной прокрутки. Значение 0 равно половине ширины экрана
-V --version	Позволяет отобразить версию программы less
-? --help	Выводит справочную информацию

Как видите, возможности программы `less` велики, и для удобства использования этой утилиты разработчики внесли в ее функционал набор команд, которые выполняются по нажатию той или иной клавиши. Давайте посмотрим на основные команды (табл. 13.12).

Таблица 13.12. Функциональные клавиши программы `less`

Клавиша	Описание
<←>, <→>, <↑> и <↓>	Прокрутка текста влево, вправо, вверх и вниз соответственно
<Enter> (Return)	Прокрутка текста на одну строку вниз
<Пробел>	Прокрутка текста на одну страницу вниз
<Home>	Установка курсора в начало текста
<End>	Установка курсора в конец текста
<v>	Переход в режим редактирования отображаемого текста
<q>	Выход из режима вывода

Это лишь основные функциональные клавиши программы `less`. Если вам для работы с программой потребуется намного больше функционала, то получить информацию о нем можно, выполнив команду `man less`.

В качестве примера рассмотрим вывод на Терминал содержимого файла `/var/log/syslog`:

```
less -N /var/log/syslog
```

Данная команда позволит вывести на Терминал содержимое нужного нам файла, при этом каждая строка будет пронумерована для удобства восприятия информации. Перемещаться вы можете клавишами <Return>, <Пробел> или клавишами-стрелками, а выйти из режима отображения текста можно, нажав клавишу <q>.

Если вы хотите выходить из режима вывода текста по нажатию на клавиши <Ctrl>+<C>, то вам следует добавить параметр -к:

```
less -NK /var/log/syslog
```

Теперь при нажатии клавиш <Ctrl>+<C> программа будет завершать свою работу. А нажатие клавиши <q> будет проигнорировано.

Для выделения первой строки при каждом прокручивании текста следует использовать параметр -w:

```
less -W /var/log/syslog
```

Данный прием позволит при прокрутке текста клавишей <Пробел> выделять первую строку на текущей странице просмотра. Это иногда бывает удобным в случае просмотра больших объемов информации.

Команда поиска файлов

Всем нам уже не раз приходилось искать тот или иной файл на компьютере, и для этих целей мы прибегали к использованию удобного графического интерфейса программы поиска файлов. ОС Ubuntu Linux также располагает инструментами поиска файлов и каталогов, но помимо графического режима операционная система позволяет производить поиск средствами командной строки, а правильней будет сказать — посредством Консоли или Терминала.

Для поиска нужных объектов на жестком диске предназначена команда `find` (от англ. *find* — найти).

В общем виде синтаксис команды `find` выглядит следующим образом:

```
find [путь] [опции]
```

Список опций, принимаемых командой `find`, довольно огромный, и сейчас мы не будем приводить его целиком, а представим лишь наиболее часто используемые. Полный набор опций вы можете просмотреть, выполнив команду `man find`.

Итак, параметр [опции] принимает значения, перечисленные в табл. 13.13.

Таблица 13.13. Параметры команды `find`

Опции	Описание
-name	Позволяет искать объект по заданному имени, при этом учитывается регистр символов
-iname	Позволяет искать объект по заданному имени, при этом регистр символов не учитывается

Таблица 13.13 (окончание)

Опции	Описание
-type	Позволяет указать тип требуемого объекта. Может принимать следующие значения: <ul style="list-style-type: none"> • <code>f</code> — файл (file); • <code>d</code> — каталог (directory); • <code>l</code> — ссылка (link); • <code>p</code> — канал (pipe); • <code>s</code> — сокет (socket)
-size	Позволяет произвести поиск по заданному размеру искомого объекта
-atime	Позволяет произвести поиск с учетом последнего обращения к объекту. Время указывается в часах
-mtime	Позволяет произвести поиск с учетом последнего изменения объекта. Время указывается в днях
-mmin	Позволяет произвести поиск с учетом последнего изменения объекта. Время указывается в минутах
-print	Позволяет отобразить на экран результат поиска
-empty	Позволяет произвести поиск пустых каталогов

Приведенные в таблице параметры — это лишь наиболее часто используемые, т. е. те, с которыми вам скорее всего придется столкнуться. Помимо этих параметров существует еще большое количество возможных критериев поиска, однако сейчас мы не будем их рассматривать.

Попробуем найти файл `myfile.txt`, который расположен в домашнем каталоге пользователя:

```
find /home/ -name myfile.txt
```

В этом примере мы первым параметром передаем команде каталог, в котором необходимо произвести поиск файла (`/home/`), а вторым параметром указываем имя файла: `-name myfile.txt`.

Данную команду лучше выполнять с административными привилегиями. В противном случае окно поиска может быть "засорено" сообщениями с текстом "Отказано в доступе" (рис. 13.8).

```

sergey@Test-PC: ~
sergey@Test-PC:~$ find /home/ -name myfile.txt
/home/sergey/myfile.txt
find: «/home/sergey/.cache/dconf»: Отказано в доступе
find: «/home/sergey/.dbus»: Отказано в доступе
find: «/home/lost+found»: Отказано в доступе
sergey@Test-PC:~$

```

Рис. 13.8. Программе `find` не хватает административных привилегий

Для того чтобы не возникало таких ошибок, выполняйте команду с административными привилегиями:

```
sudo find /home/ -name myfile.txt
```

Для того чтобы выполнить поиск всех файлов в домашнем каталоге, которые были изменены более 5 дней назад, выполните следующую команду:

```
sudo find /home/ -mtime +5
```

Список окажется довольно огромным, поэтому попробуем уточнить запрос, указав команде, что размер файла больше 10 Мбайт:

```
sudo find /home/ -mtime +5 -size +10M
```

Результат будет уже намного короче. Таким образом, правильно составляя запрос поиска, можно добиться более точных результатов.

Отображение каталога программы

Нередко опытным пользователям приходится выяснять каталог, в котором расположена та или иная программа. Обращаться к Интернету за подсказкой не всегда разумно, т. к. в ОС Ubuntu Linux предусмотрена команда специально для этой цели.

Итак, команда `which` (от англ. *which* — какой) предназначена для отображения каталога, в котором находится требуемая программа.

Программа принимает в качестве аргумента имя программы, каталог которой вы хотите узнать. Так, для выяснения расположения программы `ls` необходимо выполнить следующую команду:

```
which ls
```

После выполнения команды будет отображен каталог, в котором находится программа `ls`.

Поиск текстового шаблона в файлах

Для поиска заданного текстового шаблона в текстовых файлах в ОС Linux существует команда `grep`. Само название является почти акронимом английской фразы "search globally for lines matching the regular expression, and print them", которое переводится как "искать везде строки, соответствующие регулярному выражению и вывести их".

Синтаксис команды `grep` следующий:

```
grep [параметр]... шаблон [файл]...
```

Список опций, принимаемых командой `grep`, довольно большой, и мы сейчас рассмотрим лишь самые необходимые из них.

Итак, команда `grep` принимает значения, перечисленные в табл. 13.14.

Таблица 13.14. Параметры команды *grep*

Опции	Описание
Параметры для регулярных выражений	
-E --extended-regexp	Позволяет рассматривать <шаблон>, как расширенное регулярное выражение
-F --fixed-regexp	Позволяет рассматривать <шаблон>, как список фиксированных регулярных выражений. Каждый образец ищется как строка, а не как регулярное выражение
-G --basic-regexp	Позволяет рассматривать <шаблон>, как базовое регулярное выражение. Это поведение характерно для команды <i>grep</i> по умолчанию
-P --perl-regexp	Позволяет рассматривать <шаблон>, как регулярное выражение языка Perl
-e --regexp=шаблон	Позволяет использовать <шаблон> для поиска
-f --file=файл	Позволяет использовать шаблон для поиска из <файла>
-i --ignore-case	Позволяет при поиске не учитывать регистр
-w --word-regexp	Указывает поиску, что <шаблон> должен подходить ко всем словам
-x --line-regexp	Указывает поиску, что <шаблон> должен подходить ко всем строкам в поиске
-z --null-data	Позволяет при поиске разделять строки байтом с нулевым значением, а не символом конца строки
Управление выводом результата	
-m --max-count=число	Позволяет остановить поиск после достижения указанного <числа> совпадений
-b --byte-offset	Позволяет вывести вместе с выходными строками смещение в байтах
-n --line-number	Позволяет печатать номер строки вместе с выводимыми данными
--line-buffered	Позволяет сбрасывать буфер после каждой строки
-H --with-filename	Позволяет печатать имя файла для каждого совпадения
-h --no-filename	Параметр позволяет не начинать вывод с имени файла
--label=метка	Использовать <метку> в качестве имени файла для стандартного ввода
-o --only-matching	Позволяет показывать только часть строки, совпадающей с заданным <шаблоном>

Таблица 13.14 (продолжение)

Опции	Описание
-q --quiet --silent	Использование данного параметра позволит вывести в выходной поток только сопоставившиеся строки. Так, когда одна из входных строк соответствует образцу, возвращается статус выхода "0"
--binary-files=тип	Позволяет указать тип двоичного файла. <тип> файла может быть следующим: binary — бинарный; text — текстовый; without-match — не подходящий ни под один из перечисленных типов
-a --text	Аналогичен параметру --binary-files=text
-I	Аналогичен параметру --binary-files=without-match
-d --directories=действие	Позволяет задать способ обработки каталогов в соответствии с переданным значением <действие>. Параметр <действие> может принимать следующие значения: read — читать содержимое каталога; recurse — рекурсивно обрабатывать каталог; skip — пропускать содержимое каталогов
-r --recursive	Аналогичен ключу --directories=recurse
-R --dereference-recursive	Позволяет рекурсивно обрабатывать содержимое каталогов, переходя по всем символьным ссылкам
--include=ф_шаблон	Позволяет обработать только файлы, подпадающие под <ф_шаблон>
--exclude=ф_шаблон	Позволяет пропустить файлы и каталоги, подпадающие под <ф_шаблон>
--exclude-from=файл	Позволяет пропустить файлы, подпадающие под шаблон файлов из <файла>
--exclude-dir=шаблон	Позволяет пропустить каталоги, подпадающие под <шаблон>
-L --files-without-match	Позволяет печатать только имена [файлов] без совпадений
-l --files-with-matches	Позволяет печатать только имена [файлов] с совпадениями
-c --count	Позволяет печатать только количество совпадающих строк на [файл]
-T --initial-tab	Установка данного параметра позволяет при необходимости выравнивать результат при помощи табуляции
-Z --null	Позволяет печатать байт "0" после имени [файла]
Управление контекстом	
-B --before-context=число	Позволяет печатать [число] строк предшествующего контекста

Таблица 13.14 (окончание)

Опции	Описание
-A --after-context= <i>число</i>	Позволяет печатать [<i>число</i>] строк последующего контекста
-C --context[= <i>число</i>]	Позволяет печатать [<i>число</i>] строк контекста
--color[= <i>когда</i>] --colour[= <i>когда</i>]	Позволяет использовать маркеры для различия совпадающих строк. Параметр [<i>когда</i>] может принимать следующие значения: always — всегда; never — никогда; auto — автоматически
-U --binary	Позволяет не удалять символы CR (символ возврата каретки) в конце строки. Использование данного параметра бывает целесообразно при поиске в файлах, которые были созданы в ОС Windows
-u --unix-byte-offsets	Позволяет выдавать смещения без учета символов CR (символ возврата каретки)
Прочие параметры	
-s --no-messages	Позволяет не выводить сообщения об ошибках
-v --invert-match	Позволяет выбирать неподходящие строки, т. е. происходит инвертированный поиск

Теперь разберем служебные символы, которые используются при составлении шаблона для поиска (табл. 13.15).

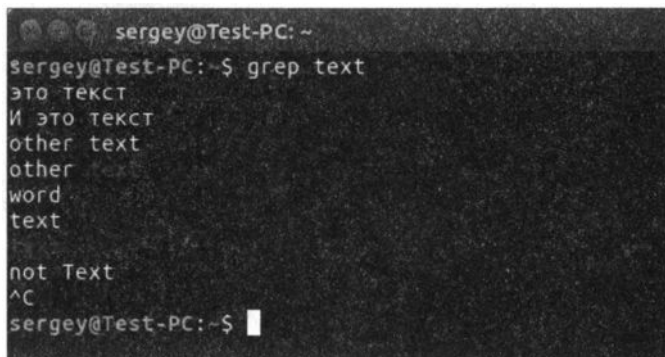
Таблица 13.15. Служебные символы шаблона поиска

Символ	Описание
.	Любой произвольный символ
*	Множитель. Предыдущий символ может повторяться любое число раз
.*	Любая последовательность символов
^ <i>шаблон</i>	Привязка шаблона к началу строки
<i>шаблон</i> \$	Привязка шаблона к концу строки
[<i>СИМВОЛЫ</i>]	Указывает на любой из символов, которые перечислены в квадратных скобках

Как вы заметили, функционал у программы `grep` просто огромный, но он того стоит. Возможно, вам и не придется в повседневной практике использовать все возможности этой программы, но знать их не помешает. Итак, давайте немного попробуем поработать с командой `grep`.

В Терминале выполним следующую команду:

```
grep text
```



```
sergey@Test-PC: ~  
sergey@Test-PC: ~$ grep text  
это текст  
И это текст  
other text  
other text  
word  
text  
  
not Text  
^C  
sergey@Test-PC: ~$
```

Рис. 13.9. Результат работы команды `grep` без параметров

Теперь попробуйте вводить различный текст, а также слово *text* и наблюдайте за результатом (рис. 13.9).

Разберемся с тем, что происходит при выполнении команды `grep` без передачи ей каких-либо параметров.

Так как команде `grep` мы не передали ни одного параметра, кроме произвольного текста, то она переключается в режим обработки стандартного ввода. Таким образом, при выполнении команды Терминал ожидает от нас ввода каких-либо данных и будет находиться в этом режиме до тех пор, пока не получит команду завершения ввода, т. е. не будет нажато сочетание клавиш `<Ctrl>+<C>`. О том, что была послана команда завершения ввода, свидетельствует последняя строка `"^C"`.

В первой, второй и третьей строках мы вводим произвольный текст, и после нажатия клавиши `<Enter>` ничего не происходит. Однако когда мы ввели слово *text*, то Терминал выделил его красным светом. Это произошло потому, что команда `grep` обнаружила переданный ей образец текста.

Теперь давайте произведем простой поиск шаблонной фразы в файле `myfile.txt`, который находится в домашнем каталоге:

```
grep Третья myfile.txt
```

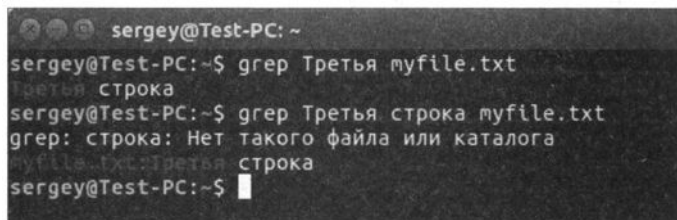
В этом примере, мы сообщаем команде `grep`, что нам требуется найти строку "Третья" в файле `myfile.txt`. В ответ на это команда `grep` отобразит искомую строку и выделит ее красным цветом.

Посмотрите внимательно на команду и попытайтесь увидеть в ней ошибки. Если вы ничего не заметили, то ничего страшного, потому что в данном варианте написания команда будет продолжать работать. Но что если мы попробуем найти две фразы: "Третья строка"?

```
grep Третья строка myfile.txt
```

И вот что получится (рис. 13.10).

При выполнении команды мы можем наблюдать ошибку поиска. Как вы думаете, почему она произошла? Дело в том, что при поиске шаблона, состоящего из двух или более слов, следует обрамлять этот искомый шаблон двойными либо одинарными кавычками.



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ grep Третья myfile.txt  
Третья строка  
sergey@Test-PC:~$ grep Третья строка myfile.txt  
grep: строка: Нет такого файла или каталога  
myfile.txt:Третья строка  
sergey@Test-PC:~$
```

Рис. 13.10. Поиск шаблонной фразы, состоящий из двух слов

ными кавычками. Если этого не сделать, то при выполнении такой ошибочно составленной команды программа `grep` будет считать, что слово "строка" — это файл, в котором нужно произвести поиск шаблонной фразы, переданной в первом аргументе, т. е. фразы "Третья". По этой причине при выполнении указанной выше команды мы наблюдаем сообщение о том, что каталог или файл с названием "строка" не найден в указанном домашнем каталоге. Поэтому при составлении шаблона из двух или более слов обрамляйте их кавычками:

```
grep "Третья строка" myfile.txt
```

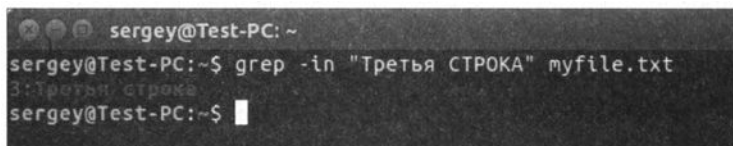
Теперь, когда искомый шаблон обрамлен кавычками, мы не увидим никаких ошибок, и заданная фраза будет найдена при условии, что она присутствует в самом файле.

Давайте еще раз посмотрим на составленный нами запрос и попробуем его модернизировать таким образом, чтобы получить больше информации об искомой фразе. Ничего не приходит на ум? Тогда выполните следующую конструкцию команды `grep`:

```
grep -in "Третья СТРОКА" myfile.txt
```

В этом примере мы добавили ключи `-i` и `-n`, которые позволяют не учитывать регистр при поиске (ключ `-i`) и выводить номер строки (ключ `-n`). Саму искомую фразу мы напечатали в разном регистре, т. е. первое слово с заглавной буквы, а второе в верхнем регистре.

По выполнении данной команды искомая фраза будет найдена и будет указан номер строки, на которой находится искомая фраза (рис. 13.11).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ grep -in "Третья СТРОКА" myfile.txt  
3:Третья строка  
sergey@Test-PC:~$
```

Рис. 13.11. Команда `grep` выполнила поиск без учета регистра и вывела номер строки

Ну что ж, результат нас устраивает. А как быть, если мы точно не помним, как правильно пишется искомая строка, знаем только, что она начинается с буквы *т*, а заканчивается слогом *ка*? Сейчас попробуем найти искомый шаблон. Давайте выполним следующую команду:

```
grep -in "^т.*ка" myfile.txt
```

Наверняка вам не понятен смысл этого выражения. Будем разбираться вместе. Итак, у нас есть шаблон `^т.*ка`, обранный кавычками. Как мы помним, кавычки нужны для того, чтобы у нас была возможность искать не только одно слово, а несколько. К тому же, использование кавычек делает шаблон несколько нагляднее. Однако сейчас нас интересует, что находится между кавычками.

Первый символ `^` (каре́т) указывает на то, что поиск должен производиться с начала строки, а не со второго или последующего символа строки. Затем идет буква `т`, т. е. начало искомого шаблона. Поскольку мы точно не знаем, какие символы идут после буквы `т`, то указываем символ точки, а так как еще и неизвестно количество символов, то мы должны указать множитель, т. е. сообщить команде `grep`, что данный символ может повторяться любое количество раз. Таким образом, получается конструкция `.*`, которая означает любую последовательность символов. Ну и завершается шаблон слогом `ка`, на который должна оканчиваться искомая строка. Остальное нам уже знакомо.

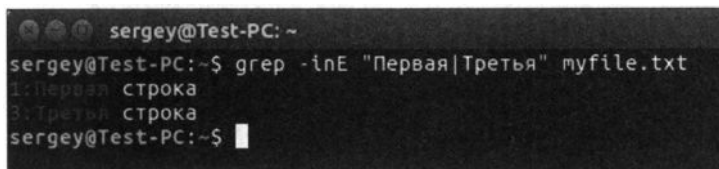
Такая с виду сложная конструкция шаблона поиска позволяет найти строку, состоящую из двух слов.

Рассмотрим ситуацию, когда требуется найти два слова, но при этом они могут быть расположены на разных строках:

```
grep -inE "Первая|Третья" myfile.txt
```

Обратите внимание на то, что мы используем ключ `-E`, который предписывает команде `grep` рассматривать наш шаблон как расширенное регулярное выражение. В самом шаблоне мы указываем два искомых слова и разделяем их символом вертикальной черты (`|`). Вертикальная черта указывается для того, чтобы перечислить искомые слова.

Результатом выполнения данной конструкции будет список, содержащий строки из двух слов, которые находятся на разных строках (рис. 13.12).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ grep -inE "Первая|Третья" myfile.txt  
1:Первая строка  
3:Третья строка  
sergey@Test-PC:~$
```

Рис. 13.12. Поиск двух слов на разных строках

Чтобы просмотреть количество вхождений искомого шаблона в файле `myfile.txt`, выполните следующую команду:

```
grep -ic "строка" myfile.txt
```

Данная команда выведет количество вхождений искомой строки, т. е. количество повторений искомого слова. Обратите внимание, что для этого мы используем ключ `-c` и при этом убрали ключ `-n`, который выводит номер строки, которая содержит искомый шаблон.

До сих пор мы работали с содержимым файлов и могли заметить, что команда `grep` просматривает только содержимое файлов, но не их имена. Однако это только на первый взгляд. Существует программный канал `pipe`, т. е. взаимодействие между процессами для обмена каких-либо данных. Это означает, что результат работы одной программы мы можем направить другой. Будет произведен обмен данными между двумя программами. Каким образом это происходит, и чем это может быть нам полезным? Давайте рассмотрим следующую ситуацию.

Итак, например, нам нужно получить список всех файлов в домашнем каталоге, имена которых заканчиваются на `".txt"`. При этом мы хотим, чтобы этот список обработала программа `grep`. Для этого нам придется выполнить следующую команду:

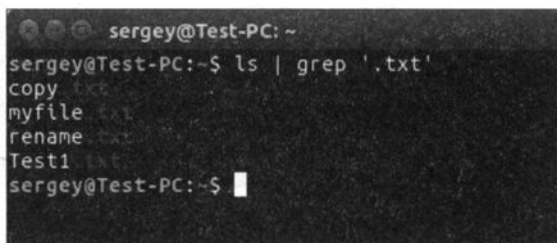
```
ls | grep '.txt'
```

Теперь поясним, что же делает эта команда.

Команда `ls` без передачи ей каких-либо параметров выводит имена объектов из текущего каталога. В нашем случае это домашний каталог пользователя. После вертикальной линии (`|`) идет команда `grep`, которой передан простой шаблон имени файла. Несколько запутано?

На самом деле все довольно просто. Сначала выполняется первая часть команды до вертикальной черты (`|`), а затем уже результат выполнения передается команде, которая находится справа от вертикальной черты (`|`). То есть получается, что результат работы команды слева был передан команде справа. Таким образом, вертикальная черта указывает командному интерпретатору `bash`, что результат работы левой конструкции команды должен быть передан команде, которая находится справа от вертикальной черты.

По выполнении этой команды мы увидим примерно следующий результат (рис. 13.13).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ ls | grep '.txt'  
copy.txt  
myfile.txt  
rename.txt  
Test1.txt  
sergey@Test-PC:~$
```

Рис. 13.13. Создание программного канала `pipe`

Выполненная нами команда возвратила список файлов, имена которых заканчиваются на `".txt"`.

Сейчас мы увидели лишь небольшой минимум из того, что умеет выполнять команда `grep`. На полное перечисление всех возможностей программы `grep` ушло бы несколько десятков страниц.

Управление остановкой системы

Мы уже столько знаем о работе в Терминале, что для нас не удивительно, что Терминал также позволяет управлять работой операционной системы с помощью определенных команд.

Для управления остановкой работы операционной системы предназначена команда `shutdown` (от англ. *shutdown* — выключить).

Перед тем как переходить к описанию команды `shutdown`, давайте внесем некоторую ясность, чтобы в дальнейшем не возникало путаницы и лишних вопросов.

По мере развития и совершенствования операционной системы Linux и самих компьютеров для управления питанием компьютера использовались разные команды. Так, первой командой завершения работы операционной системы стала команда `halt`. Она завершала работу системы, но не выключала ее питание. Эта устаревшая команда была предназначена для старых компьютеров, которые не поддерживали расширенное управление питанием. После выполнения этой команды появлялось сообщение о том, что теперь уже можно отключить питание.

Второй устаревшей командой после `halt` стала команда `poweroff`. Она позволяла не только завершать работу операционной системы, но и отключать ее питание. Одна эта команда не смогла выдержать "конкуренцию временем". Например, команда `poweroff` "не умела" выключать компьютер в заданное время. На первый взгляд может показаться странным, зачем кому-то может понадобиться выключать компьютер в строго определенное время, но такая ситуация встречается довольно часто, когда требуется, например, уйти с работы пораньше и выключить компьютер, который выполняет роль файлообменника. Разумеется, если выключить такой компьютер, то возникнет очень неприятная ситуация, при которой остановится работа всех тех, кто в данный момент пользуется ресурсами этого файлообменника. Так вот, чтобы не произошла такая ситуация, можно запланировать выключение компьютера в определенное время, но команда `poweroff` не располагает таким функционалом, и ей на смену пришла команда `shutdown`, о которой сейчас и пойдет речь.

Давайте для начала посмотрим на синтаксис команды `shutdown`:

```
shutdown [опции...] [время] [сообщение...]
```

Параметр `[опции...]` принимает значения, перечисленные в табл. 13.16.

Таблица 13.16. Параметры команды `shutdown`

Опции	Описание
-h --halt	Завершение работы компьютера
-p --poweroff	Завершение работы операционной системы
-r --reboot	Перезагрузка компьютера

Таблица 13.16 (окончание)

Опции	Описание
-h	Параметр аналогичен параметру <code>--poweroff</code>
-k	Параметр позволяет отправить предупреждение о том, что компьютер будет выключен или перезагружен
--no-wall	Параметр позволяет не отправлять сообщения о перезагрузке или выключении компьютера
-c	Параметр позволяет отменить запланированное отключение или перезагрузку компьютера
--help	Вывод справочной информации

Давайте на примере убедимся в возможностях команды `shutdown`. Для начала запланируем выключение компьютера в 18:05. Для этого выполним следующую команду:

```
shutdown -P 18:05
```

В ответ на это мы получим сообщение со статусом запланированной операции (рис. 13.14).

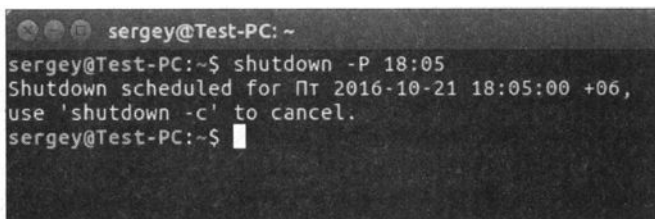


Рис. 13.14. Запланировано выключение компьютера

Для того чтобы отменить запланированное отключение или перезагрузку компьютера, выполните следующую команду:

```
shutdown -c
```

Вы также можете добавить произвольное сообщение к запланированной операции. Давайте для примера добавим сообщение к запланированной операции перезагрузки компьютера:

```
shutdown -r 18:05 [Перезагружаемся в 18:05]
```

Таким образом, мы оповестим пользователей о том, что работа компьютера будет прервана по той или иной причине.

Чтобы осуществить перезагрузку компьютера прямо сейчас, выполните следующую команду:

```
shutdown -r now
```

Она позволит выполнить перезагрузку. Аналогичным образом производится и выключение системы:

```
shutdown -P now
```

Получается, что команда `shutdown` позволяет не только перезагрузить компьютер, но также выключить питание.

ПРИМЕЧАНИЕ

Для перезагрузки существует еще и команда `reboot`, которая на первый взгляд аналогичная команде `shutdown -r now`. Однако это только на первый взгляд. На деле же команда `reboot` отличается от более функциональной команды `shutdown` тем, что последняя позволяет более мягко завершать работу операционной системы, вызывая для этого системные и локальные `rc.d`-скрипты с параметром `stop`. При выполнении команды `reboot` данные `rc.d`-скрипты не выполняются, и ресурсоемкие процессы просто не успевают сохранить свои данные.

Выводы

Мы изучили довольно большой объем полезных команд, с которыми нам придется сталкиваться при работе в ОС Ubuntu Linux.

Мы научились создавать файлы и каталоги, а также переименовывать и перемещать их.

Мы теперь знаем, как просматривать содержимое файла в Терминале, а также производить поиск по шаблону с помощью команды `grep`. И в качестве закрепления материала по работе в Терминале мы научились управлять перезагрузкой и выключением компьютера.

ГЛАВА 14



Работа с привилегиями root

О привилегиях root

В любой современной операционной системе присутствует учетная запись, которая имеет самые наибольшие права. Пользователю этой учетной записи доступны практически все полномочия в системе, разумеется, в разумных пределах, которые определены самой операционной системой.

Так, в ОС Linux, в частности Ubuntu Linux, присутствует учетная запись root (читается как "рут").

Пользователь root (от англ. *root* — корень) — это встроенная учетная запись в UNIX-подобных системах, которая имеет идентификатор (UID, User Identifier — идентификатор пользователя) 0. Эта учетная запись обладает административными полномочиями в системе, и ее (учетной записи) владелец может выполнять практически все операции в системе. Подобных административных прав лишены остальные учетные записи, и при выполнении тех операций, которым необходимы права пользователя root, будет выводиться соответствующее предупреждение о необходимости выполнения операции от имени пользователя root.

Ограничить права пользователя root практически невозможно.

В ранних дистрибутивах ОС Ubuntu Linux присутствовала возможность авторизоваться под именем учетной записи root и тем самым можно было получить неограниченные привилегии над ОС. Однако в связи со злоупотреблением работой под учетной записью root разработчики дистрибутивов решили отключить данную учетную запись.

Сделано это было по причине того, что пользователи, однажды авторизовавшись под учетной записью с полными административными правами, начинали злоупотреблять данной привилегией и тем самым постоянно работали под учетной записью root. Эта привилегия только на первый взгляд приносила удобства в работе с ОС. На деле же пользователи получили большое количество проблем, связанных с безопасностью их систем.

Представьте ситуацию, когда пользователь работает под учетной записью root и ему случайно попадает вредоносная программа, которую он успешно запускает.

Так как запущен сеанс пользователя root, то программа получит те же самые права, а это значит, что она сможет выполнять все, что предусмотрел разработчик этой вредоносной программы. Это одна из самых простых ситуаций, при которой постоянная работа от имени root может нанести урон операционной системе.

Таким образом, разработчики дистрибутивов приходят к единому мнению, что нужно отключить учетную запись root. Разумеется, учетная запись не была удалена, просто зайти под именем этой учетной записи теперь нельзя.

Однако отключение учетной записи root вовсе не означает, что вы не сможете выполнять никаких операций, требующих административных привилегий.

Для выполнения административных задач предназначена команда `sudo`, которая временно предоставляет те самые административные права, которые необходимы для выполнения операции, требующей прав root.

Команда `sudo` вместо прав root

Для выполнения операций, требующих административных прав root, разработчиками операционной системы была придумана утилита `sudo`, которая предоставляет нужные права для выполнения той или иной команды.

`sudo` (от англ. *substitute user and do* — подменить пользователя и выполнить) — это утилита, предоставляющая административные привилегии (привилегии root), когда это необходимо, т. е. по первому запросу пользователя.

Получается, что при выполнении команды `sudo` пользователю, выполнившему эту команду, доступны те же полномочия, что и пользователю root.

Выполнять команду `sudo` могут не все пользователи, а только те, которые входят в группу **admin**. По умолчанию первая учетная запись, которую вы создавали на этапе установки ОС Ubuntu Linux, получает право на использование команды `sudo`. То есть фактически получается, что первая учетная запись сразу получает право на использование этой команды. А как быть с другими пользователями, неужели они тоже могут получить право на использование команды `sudo`? В ОС Ubuntu Linux присутствует файл `/etc/sudoers`, где прописаны правила, на основании которых система уже определяет, позволено ли тому или иному пользователю выполнять команду `sudo`.

Что использовать: `sudo` или `su`?

Очень часто в технической литературе или иных справочных источниках можно встретить упоминания команды `su` вместо `sudo`. Отсюда возникает огромное количество вопросов на тему, какую же команду использовать? Давайте разберемся в этом спорном вопросе и расставим все точки над *i*.

Итак, до появления команды `sudo` ее роль играла команда `su`. Запущенная команда `su` запрашивала пароль пользователя root и подменяла текущее имя пользователя на root. Таким образом, пользователю, выполнившему команду `su`, помимо своего

пароля, приходилось также помнить пароль пользователя `root`. Пользователь, который мог выполнять соответствующую команду, должен был входить в группу **wheel**, которой разрешалось выполнение команды `su`. Разумеется, это было серьезной проблемой в безопасности системы.

По этой причине разработчиками было решено ввести новую команду `sudo`, которая заменит собой устаревшую команду `su`.

С возникновением команды `sudo` у администратора системы появилась возможность указать список разрешенных команд индивидуально для каждого пользователя, в то время как команда `su` этого не предусматривала.

Отличием новой команды стало и то, что команда `sudo` запрашивает у пользователя его собственный пароль, а не пароль пользователя `root`. А также команда `sudo` записывает всю пользовательскую активность в `syslog`-канал `authpriv`, т. е. в файл `/var/log/auth.log`.

Важным стало и то, что при лишении пользователя права выполнять команду `sudo` пользователя достаточно просто вывести из группы **admin** или исключить из файла `/etc/sudoers` (если он подвергся изменению), в то время как пользователь, выполнявший команду `su`, должен был забыть пароль пользователя `root`.

Конечно же никто не забывал пароль пользователя `root`, а администратор системы, как правило, его не менял даже после удаления пользователя из группы **wheel**. Таким образом, система могла подвергнуться взлому по той причине, что исключенный пользователь команды `su` уже знал пароль суперпользователя `root`.

Это были наиболее веские причины отказа от устаревшей команды `su` в пользу усовершенствованной команды `sudo`.

Команда `sudo`

Теперь, когда мы знаем о причинах появления команды `sudo` и ее назначении, стоит поговорить о ней более подробно.

Для начала познакомимся с синтаксисом команды `sudo`. В общем виде команда `sudo` выглядит следующим образом:

```
sudo <команда> [параметры]
```

В первом аргументе `<команда>` указывается имя программы, которую необходимо выполнить с правами `root`. Во втором аргументе `[параметры]` задаются необязательные параметры, которые могут быть следующими (табл. 14.1).

Таблица 14.1. Параметры команды `sudo`

Параметр	Описание
<code>-A</code> <code>--askpass</code>	Позволяет использовать вспомогательную программу для ввода пароля
<code>-b</code> <code>--background</code>	Позволяет выполнить указанную команду в фоновом режиме

Таблица 14.1 (продолжение)

Параметр	Описание
-C --close-from= <i>num</i>	Позволяет закрыть все дескрипторы файлов, которые больше или равны значению, переданному в параметре <i>num</i>
-E --preserve-env	Позволяет сохранить пользовательское окружение при выполнении команды
-e --edit	Позволяет редактировать файлы вместо выполнения команды
-g --group= <i>group</i>	Позволяет выполнить команду от имени или ID указанной в параметре <i>group</i> группы
-H --set-home	Позволяет установить для переменной HOME домашний каталог
-h --host= <i>host</i>	Позволяет выполнить команду на узле, при условии если таковая поддерживается модулем ядра
-i --login	Позволяет запустить оболочку входа в систему от имени указанного пользователя, а также задать команду, которая будет выполнена при входе в систему
-K --remove-timestamp	Позволяет полностью удалить файл <i>timestamp</i>
-k --reset-timestamp	Позволяет объявить недействительным файл <i>timestamp</i>
-l --list	Позволяет показать список прав пользователя или проверить заданную команду
-n --non-interactive	Позволяет использовать автономный режим без вывода запросов пользователю
-P --preserve-groups	Позволяет сохранить вектор группы вместо установки целевой группы
-p --prompt= <i>prompt</i>	Позволяет использовать указанный запрос пароля
-r --role= <i>role</i>	Позволяет создать контекст безопасности SELinux с указанной ролью, переданной в параметре <i>role</i>
-S --stdin	Позволяет читать пароль из стандартного ввода
-s --shell	Позволяет запустить оболочку от имени указанного пользователя, а также задать команду, которая будет выполнена при запуске оболочки
-t --type= <i>type</i>	Позволяет создать контекст безопасности SELinux указанного типа
-U --other-user= <i>user</i>	Позволяет в режиме списка показывать права пользователя
-u --user= <i>user</i>	Позволяет выполнить команду (или редактировать файл) от имени или ID указанного пользователя

Таблица 14.1 (окончание)

Параметр	Описание
<code>-v</code> <code>--validate</code>	Позволяет обновить временную метку пользователя без выполнения команды
<code>--</code>	Позволяет прекратить обработку аргументов командной строки

Как вы видите, параметров у команды `sudo` достаточно.

Давайте в качестве примера использования команды `sudo` выполним следующую команду:

```
sudo -ll
```

После того как вы для выполнения команды нажмете клавишу <Enter>, Терминал запросит ваш пароль. Стоит учитывать, что при вводе пароля Терминал не будет отображать какие-либо символы. Это свойство Консоли и Терминала специально было введено для усиленной безопасности, чтобы никто посторонний не мог не только подсмотреть ваш пароль, но даже не смог увидеть длину пароля.

Просто введите свой пароль и нажмете клавишу <Enter>. Данная команда позволит вывести текущую конфигурацию программы `sudo` (рис. 14.1).

```

sergey@Test-PC: ~
sergey@Test-PC:~$ sudo -ll
[sudo] пароль для sergey:
Matching Defaults entries for sergey on Test-PC:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/
bin\:/sbin\:/bin\:/snap/bin

User sergey may run the following commands on Test-PC:

Sudoers entry:
    RunAsUsers: ALL
    RunAsGroups: ALL
    Команды:
        ALL
sergey@Test-PC:~$

```

Рис. 14.1. Вывод текущей конфигурации программы `sudo`

В данном выводе отображена текущая конфигурация прав пользователя, от имени которого выполнялась команда `sudo` с параметрами `-ll`. Эта конфигурация находится в файле `/etc/sudoers`, который мы обсудим несколько позже.

Теперь рассмотрим пример того, как можно узнать привилегии другого пользователя. Для этого нужно выполнить следующую команду:

```
sudo -lU sergey
```

В этом примере первый параметр `-l` дает возможность вывести список прав пользователя. Вторым параметром `-U` позволяет задать имя конкретного пользователя,

которое мы передаем последним параметром. В нашем случае мы узнаем права пользователя `sergey`. Этим же методом мы можем узнать право пользователя `root`, передав команде `sudo` в качестве последнего параметра имя пользователя `root`.

"Все это прекрасно", — скажете вы. Но как быть, если необходимо открыть сессию в Терминале под именем пользователя `root`, а не под нашей учетной записью?

Для этих целей разработчики предусмотрели ключ `-i`, который переводит текущего пользователя в сессию от имени пользователя `root`:

```
sudo -i
```

Терминал запросит ваш пароль и перейдет в окружение пользователя `root` (рис. 14.2).

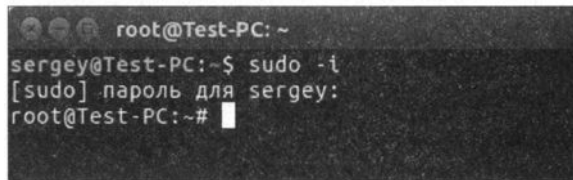


Рис. 14.2. Терминал открыл сессию от имени пользователя `root`

Не замечаете ничего необычного? Взгляните тогда на третью строку. Как видите, приглашение Терминала изменилось и выглядит теперь так:

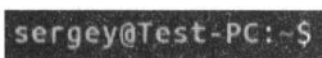
```
root@Test-PC:~#
```

Итак, давайте рассмотрим изменившееся приглашение Терминала и разберем его на составляющие (табл. 14.2).

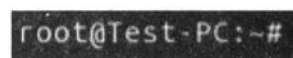
Таблица 14.2. Описание составляющих текста приветствия

Параметр	Описание
root	Имя учетной записи пользователя
@	Символ разделителя
Test-PC	Имя компьютера
:	Символ разделителя
~	Каталог выполнения команды. В нашем случае команды выполняются в домашнем каталоге пользователя <code>root</code>
#	Приглашение к выполнению команды с правами пользователя <code>root</code>

Для наглядности сравним приглашение от имени пользовательской учетной записи и приглашение от имени пользователя `root` (рис. 14.3).



а



б

Рис. 14.3. Сравнение приглашения обычного пользователя (а) и пользователя `root` (б)

Справа у нас приглашение от имени `root`, а слева от стандартной учетной записи. Как вы уже заметили, поменялась первая и последняя части приглашения ввода. Так, вместо имени пользователя `sergey` у нас стоит имя `root`, а вместо символа `$` стоит `#`.

Таким образом, символ `#` в строке приглашения указывает на то, что команды будут выполнены от имени пользователя `root`, а не от стандартной учетной записи.

Обратите внимание, что ключ `-i` работает в окружении пользователя `root`. Это значит, что вместо вашего домашнего каталога `/home/sergey/` вы будете работать в каталоге `/root/`, т. е. в домашнем каталоге пользователя `root`.

ПРИМЕЧАНИЕ

Как вы помните, домашнее окружение пользователя `root` было выведено в корень файловой системы для того, чтобы его было удобней монтировать на другой раздел жесткого диска.

Не всегда бывает удобным работать в пользовательском окружении `root`, да и не всегда это правильно. По этой причине лучше всего работать в своем пользовательском окружении. Неужели придется каждый раз при использовании команды `sudo` с ключом `-i` менять текущий каталог? Нет. Есть гораздо более удобный вариант:

```
sudo -s
```

Данная команда позволяет работать от имени пользователя `root`, но уже в своем пользовательском окружении. Этот вариант предпочтительнее, нежели приведенный выше. Поэтому при выборе ключа `-i` или `-s` вы должны исходить из принципа целесообразности: действительно ли нужно вам работать в окружении пользователя `root` или можно выполнить те же самые операции, но уже в своем окружении?

В большинстве случаев применение ключа `-s` позволяет решить все требуемые задачи.

Запуск графических программ от имени администратора

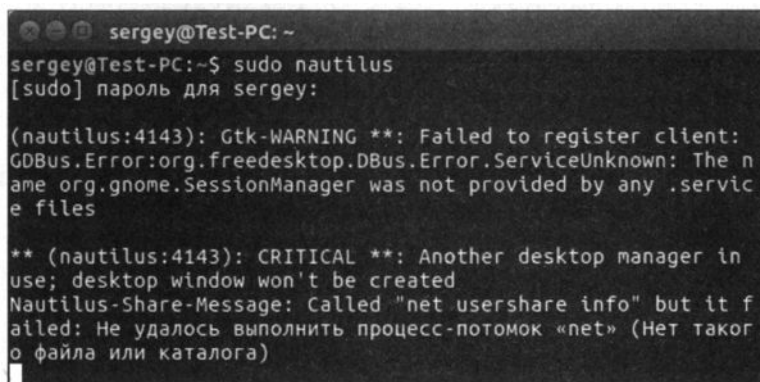
Теперь, когда мы умеем запускать программы с административными привилегиями, стоит сказать несколько слов о запуске графических программ в административном режиме.

К примеру, если вы запустите файловый менеджер `Nautilus` следующей командой:

```
sudo nautilus
```

в Терминале появятся вот такие ошибки (рис. 14.4).

Эти ошибки не критичны, и вы можете спокойно работать с файловым менеджером `Nautilus`. Однако его работа будет ограничена временем жизни терминальной сессии. Это означает, что если закрыть окно Терминала, то одновременно с Терминалом будет закрыт файловый менеджер `Nautilus`.



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ sudo nautilus  
[sudo] пароль для sergey:  
  
(nautilus:4143): Gtk-WARNING **: Failed to register client:  
GDBus.Error:org.freedesktop.DBus.Error.ServiceUnknown: The n  
ame org.gnome.SessionManager was not provided by any .servic  
e files  
  
** (nautilus:4143): CRITICAL **: Another desktop manager in  
use; desktop window won't be created  
Nautilus-Shares-Message: Called "net usershare info" but it f  
ailed: Не удалось выполнить процесс-потомок «net» (Нет таког  
о файла или каталога)
```

Рис. 14.4. Запуск Nautilus через Терминал

"Почему это происходит?" — спросите вы. Дело в том, что изначально все команды, запускаемые в ОС Ubuntu Linux, предназначались для работы в консольном режиме, т. е. без графической оболочки. Именно поэтому Терминал несколько иначе "реагирует" на запуск графических программ.

Возникает резонный вопрос: а как же в таком случае "правильно" запускать графические программы от имени администратора?

Для этих целей в ОС Ubuntu предусмотрен "графический аналог" команды `sudo` — `gksu`, перед использованием его следует установить:

```
sudo apt install gksu
```

После установки `gksu` выполнение этой команды позволит выполнить запуск любого графического программного обеспечения от имени пользователя `root`. При закрытии окна Терминала запущенное графическое приложение продолжит работать.

Теперь, чтобы запустить любое графическое приложение с правами администратора, например Nautilus, нам потребуется всего лишь выполнить следующую команду:

```
gksu nautilus
```

В ответ на это `gksu` запросит ваш пароль уже не в Терминале, а в графическом окне (рис. 14.5).

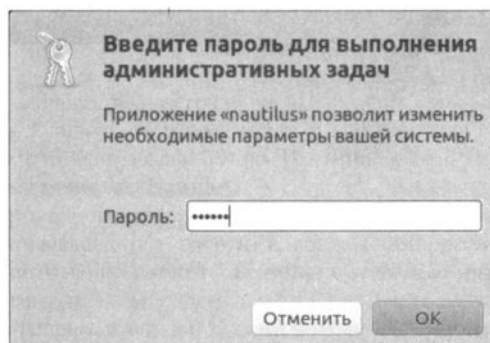


Рис. 14.5. Запрос пароля пользователя

При первом запуске файлового менеджера Nautilus будет выведено предупреждение о том, что для данной программы не существует каталога с конфигурационными файлами (рис. 14.6).

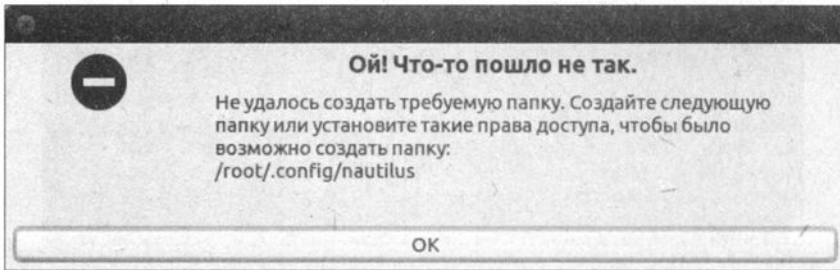


Рис. 14.6. Предупреждение о том, что в домашнем каталоге пользователя не существует конфигурационного файла

Не стоит пугаться этого предупреждения, потому что по нажатию кнопки **ОК** требуемый каталог будет создан и последующий запуск программы уже будет без соответствующего предупреждения.

Таким способом производится запуск графических программ с правами root.

Настройка *sudo* и прав доступа на выполнение команд

Ни для кого не секрет, что выполнение команды `sudo` регламентируется файлом `/etc/sudoers`. Он описывает права доступа конкретных пользователей системы на выполнение тех или иных команд, требующих административных привилегий.

Ранее мы уже затрагивали тему отображения текущей конфигурации `sudo` следующей командой:

```
sudo -ll
```

При выводе конфигурации `sudo` информация считывается из файла `/etc/sudoers`. Так как этот файл является текстовым и его можно с легкостью прочитать, то у многих сразу же возникает желание "напрямую" отредактировать этот файл в соответствии со своими личными предпочтениями. Однако этого делать не стоит, т. к. в случае прямого редактирования данного файла существует большая вероятность допустить синтаксическую ошибку, и это повлечет за собой невозможность дальнейшего запуска команды `sudo`. Во избежание возможных ошибок при редактировании файла разработчики операционной системы придумали специальный инструмент модификации `/etc/sudoers`, который называется `visudo`.

Visudo — это утилита в составе ОС Ubuntu Linux, специально предназначенная для редактирования файла `/etc/sudoers`. Данная программа позволяет наиболее безопасно производить редактирование файла `/etc/sudoers`. *Visudo*, при ее вызове, временно блокирует файл `/etc/sudoers` и создает временный файл, в который записываются

все изменения. Перед сохранением `visudo` проверяет файл на наличие синтаксических ошибок, а затем сохраняет все изменения в оригинальный файл `/etc/sudoers`.

Настройка `sudo` осуществляется посредством добавления записей в файл `/etc/sudoers`. В качестве примера разрешим определенному пользователю выполнять команду `sudo`. Для начала нам нужно запустить редактор `visudo`:

```
sudo visudo
```

Как видите, для редактирования файла `/etc/sudoers` средствами `visudo` совсем необязательно указывать путь к самому файлу. При выполнении данной команды запустится редактор `visudo` (рис. 14.7).

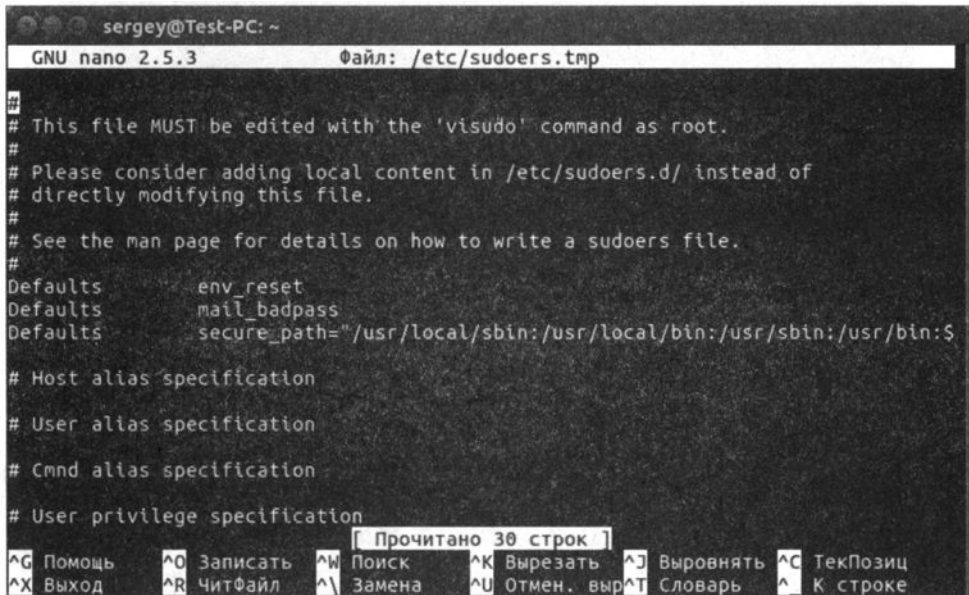


Рис. 14.7. Запущен редактор `visudo`

Перемещение курсора в редакторе `visudo` осуществляется с помощью клавиш-стрелок, а не с помощью мыши. Однако прокручивать файл можно колесиком мыши. Установите курсор после строки

```
# Members of the admin group may gain root privileges
```

и нажмите клавишу `<Enter>` для того, чтобы с новой строки написать следующее:

```
test-user ALL=(ALL) ALL
```

Данная строка позволит пользователю с именем `test-user` иметь возможность запуска команды `sudo`. Для сохранения проделанных изменений нажмите сочетание клавиш `<Ctrl>+<O>`. В ответ на это вам будет предложено сохранить изменения во временный файл `/etc/sudoers.tmp` (рис. 14.8).

Вы можете сохранить этот файл с предложенным именем, однако его конфигурация не будет учитываться. Для внесения изменений в оригинальный файл следует

выбрать имя `/etc/sudoers` и нажать клавишу `<Enter>`. В ответ будет выведено предупреждение о том, что файл `/etc/sudoers` существует, и появится вопрос, следует ли его перезаписать (рис. 14.9).

```

sergey@Test-PC: ~
GNU nano 2.5.3      Файл: /etc/sudoers.tmp      Изменён

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
test-user ALL=(ALL) ALL
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
Имя файла для записи: /etc/sudoers.tmp
^G Помощь      M-D Формат ДОМ-A Доп. в наM-B Резерв. копи
^C Отмена      M-M Формат МаM-P Доп. в коAT К файлам

```

Рис. 14.8. Сохранение изменений в файле `/etc/sudoers`

```

sergey@Test-PC: ~
GNU nano 2.5.3      Файл: /etc/sudoers.tmp      Изменён

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
test-user ALL=(ALL) ALL
Файл существует, ПЕРЕЗАПИСАТЬ ?
Y Да
N Нет      ^C Отмена

```

Рис. 14.9. Внесение изменений в оригинальный файл

Нажмите клавишу `<Y>` для подтверждения изменений оригинального файла. Для выхода из режима редактирования нажмите сочетание клавиш `<Ctrl>+<X>`.

Обратите внимание, что добавление тому или иному пользователю прав на выполнение команды `sudo` вовсе не означает, что ему больше не нужно вводить пароль учетной записи. Это всего лишь наделит пользователя правом на выполнение команды `sudo`.

Для отключения запроса на ввод пароля при использовании определенных команд нужно после строки `"# Cmnd alias specification"` в файле `/etc/sudoers` добавить строку с перечислением тех команд, на выполнение которых не нужен запрос пароля. Данная строка будет выглядеть следующим образом:

```
Cmd_Alias MY_CMD = /usr/sbin/visudo, /usr/bin/apt
```

Теперь поясним, что означает эта запись.

Итак, первое слово `Cmd Alias` означает, что следующее за ним слово будет являться псевдонимом для набора команд. В нашем случае второе слово `MY_CMD` является псевдонимом и может быть произвольным.

После псевдонима `MY_CMD` идет знак равенства, который означает, что команды, перечисленные после `=`, должны быть ассоциированы с этим псевдонимом.

Теперь нужно настроить конфигурационный файл таким образом, чтобы соотнести конкретную учетную запись со списком команд, для которых мы хотим отключить запрос ввода пароля. Для этого нам нужно в конец файла `/etc/sudoers` добавить следующую строку:

```
test_user ALL=(ALL) NOPASSWD: MY_CMD
```

Она задает отключение пароля для учетной записи `test_user`. Обратите внимание, что последним словом указан псевдоним для списка команд, для которых мы отключили запрос на ввод пароля.

ПРИМЕЧАНИЕ

Не забывайте, что отключение запроса ввода пароля ставит под сомнение безопасность вашей системы. Данное действие разумно лишь для домашнего компьютера, но не рекомендуется для работы на компьютерах с важными данными.

Существует также еще и способ полного отключения запроса на ввод пароля, к которому стоит прибегать лишь в крайних случаях и только на домашних компьютерах, но никак не на рабочих серверах.

Итак, если вы твердо решили отменить запрос на ввод пароля при выполнении команд, требующих административных привилегий, то добавьте строку

```
Defaults:sergey !authenticate
```

в файл `/etc/sudoers` после секции `"# See the man page for details on how to write a sudoers file."`.

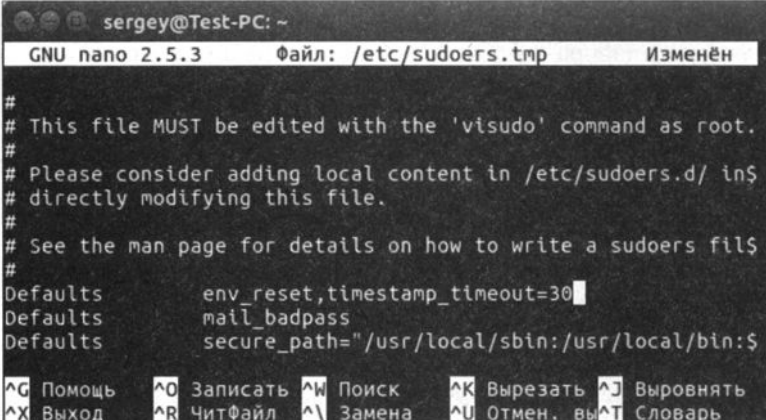
После сохранения изменений в оригинальном файле от вас больше не потребуется ввод пароля. Помните, что использовать данную возможность нужно лишь в том случае, если вы точно уверены, что такая отмена запроса пароля вам действительно необходима.

Время действия введенного пароля

Замечали ли вы, что введенный пароль в Терминале действует в течение какого-то времени, а затем его приходится вводить вновь? По умолчанию это время действия *sudo-сессии* равняется 15 минутам. Таким образом, по истечении этого времени вам придется заново вводить пароль при использовании команд, требующих повышенных привилегий. Изменить это значение довольно легко. Откройте для редактирования уже знакомый нам файл `/etc/sudoers` и найдите в нем строку `"Defaults env_reset"`. Поставьте после нее запятую и добавьте следующий параметр:

```
timestamp_timeout=30
```

У вас должно получиться нечто подобное (рис. 14.10).



```
sergey@Test-PC: ~  
GNU nano 2.5.3      Файл: /etc/sudoers.tmp      Изменён  
#  
# This file MUST be edited with the 'visudo' command as root.  
#  
# Please consider adding local content in /etc/sudoers.d/ in$  
# directly modifying this file.  
#  
# See the man page for details on how to write a sudoers fil$  
#  
Defaults      env_reset,timestamp_timeout=30  
Defaults      mail_badpass  
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:$  
^G  Помощь      ^O  Записать     ^W  Поиск        ^K  Вырезать     ^J  Вывернуть  
^X  Выход        ^R  ЧитФайл     ^\  Замена       ^U  Отмен. вы    ^T  Словарь
```

Рис. 14.10. Задание времени действия введенного пароля

В нашем случае число 30 означает время действия sudo-сессии, которое равно 30 минутам. Если вы хотите задать другой промежуток времени действия sudo-сессии, то введите свое время в минутах.

Если установить параметр равным нулю, то вам придется каждый раз вводить пароль, когда выполняются команды с `sudo`. Если же вы установите значение -1, то пароль будет запрошен только один раз, пока вы не закроете окно Терминала.

Выводы

Мы рассмотрели тему работы с привилегиями root в ОС Ubuntu. Мы узнали, зачем вместо учетной записи root была введена команда `sudo` и почему пользоваться ею нужно с осторожностью.

Мы также научились запускать графические приложения с административными привилегиями и редактировать конфигурацию прав на использование команды `sudo`.

ГЛАВА 15



Установка драйверов в ОС Ubuntu

Драйверы в ОС Ubuntu

Знакомясь с ОС Ubuntu Linux, вы могли заметить, что эта операционная система по умолчанию определяет практически все оборудование и не требует дополнительной его установки, например видео- или звуковой карты и т. п. Однако, справедливости ради, стоит отметить, что в силу некоторых особенностей ОС Ubuntu не может содержать весь набор драйверов для всех видов оборудования. Это связано в первую очередь с тем, что объем дистрибутива будет в несколько раз превышать размер самой операционной системы.

Для современной операционной системы важным свойством будет наличие доступа в Интернет сразу же после ее установки. Согласитесь, очень сложной окажется ситуация, когда на компьютере с новой ОС потребуется установить какую-либо программу или драйвер, а доступ в Интернет будет отсутствовать в связи с тем, что эта операционная система не смогла распознать сетевую карту. Конечно, можно воспользоваться другим компьютером с доступом в Сеть, но это крайне неудобно. Ведь не каждый будет рад "бегать" от одного компьютера к другому, пытаясь установить нужное ему оборудование.

Все то, что устанавливается вместе с ОС Ubuntu, является *свободным программным обеспечением*. Это касается и драйверов. Сторонние драйверы, которые еще называются *проприетарными*, требуют дополнительной установки: они не устанавливаются автоматически из-за их требований лицензионного соглашения.

Чтобы просмотреть список доступных проприетарных драйверов в вашей операционной системе, откройте **Параметры системы** и в разделе **Система** выберите **Программы и обновления**, а затем в открывшемся одноименном окне перейдите на вкладку **Дополнительные драйверы** (рис. 15.1).

ПРИМЕЧАНИЕ

У вас должно быть рабочее подключение к Интернету, т. к. поиск доступных драйверов производится с помощью Интернета.

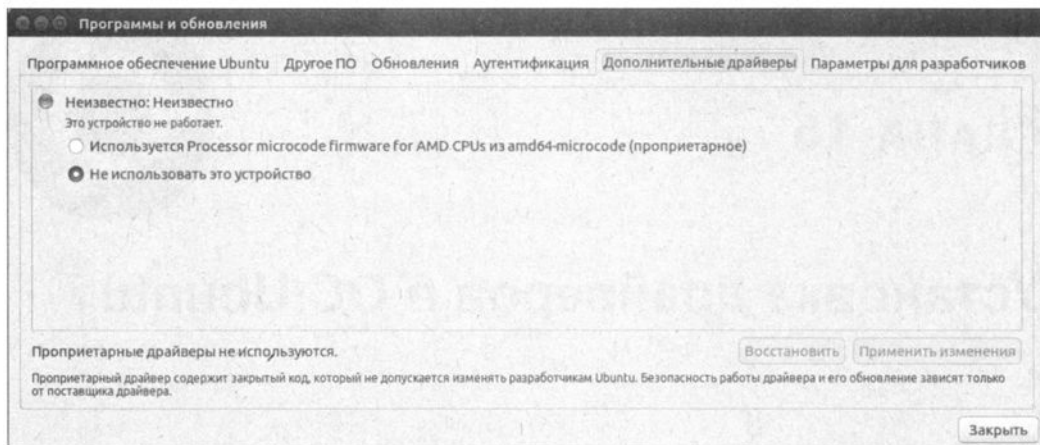


Рис. 15.1. Доступные проприетарные драйверы

Если у вас пустой список дополнительных драйверов, это значит, что ваша операционная система полностью поддерживает установленное оборудование в вашем компьютере, и вам не требуется дополнительной установки драйверов.

Если вам все же необходимо установить проприетарный драйвер, то следует установить переключатель в первое положение **Используется...** и нажать кнопку **Применить изменения**. Однако за стабильность работы драйвера не ручаются даже сами разработчики, поэтому вероятны ситуации, когда подобная установка может привести к возникновению проблем в работе операционной системы. По этой причине следует всегда быть готовым к возникновению таких ошибок и иметь запасной вариант для устранения неполадок в работе с ОС.

В любом случае отчаиваться не стоит, т. к. в большинстве случаев все работает по умолчанию, а если ваше оборудование не определилось — всегда есть возможность поискать решение в Интернете. Разумеется, если ваша сетевая карта была успешно определена в ОС Ubuntu.

Установка сетевой карты Realtek

Логичнее всего начинать установку драйверов с сетевой карты, если сетевая карта не определилась операционной системой и у вас нет доступа к Интернету. В этом случае вам ничего не остается, кроме как прибегнуть к рабочему компьютеру, с помощью которого вы сможете скачать необходимый драйвер.

Для начала нам нужно понять, нашла ли операционная система сетевую карту. Для этого в Терминале выполняем следующую команду:

```
ifconfig -a
```

Если повезло, то вывод команды будет примерно таким (рис. 15.2).

В первой строке напротив названия интерфейса `enp1s0` указаны его сетевые параметры, второй строкой `lo` обозначен интерфейс локальной петли, предназначенной для проверки передающего устройства.

```
sergey@Test-PC: ~  
sergey@Test-PC:~$ ifconfig -a  
enp1s0    Link encap:Ethernet  HWaddr d0:50:99:95:d9:65  
          inet addr:192.168.1.130  Bcast:192.168.1.255  Mask:255.255.255.0  
          inet6 addr: fe80::750e:1bca:c54b:b52e/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:42591 errors:0 dropped:63 overruns:0 frame:0  
          TX packets:5901 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:7986945 (7.9 MB)  TX bytes:762749 (762.7 KB)  
  
lo        Link encap:Локальная петля (Loopback)  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING  MTU:65536  Metric:1  
          RX packets:1101 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:1101 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1  
          RX bytes:168796 (168.7 KB)  TX bytes:168796 (168.7 KB)  
  
sergey@Test-PC:~$
```

Рис. 15.2. Отображение доступных сетевых интерфейсов

ПРИМЕЧАНИЕ

Если ОС Ubuntu не смогла определить вашу сетевую карту, то в списке доступных сетевых интерфейсов будет отображаться только *локальная петля*, т. е. виртуальный интерфейс `lo`.

В нашем случае сетевая карта определилась и не требует дополнительной установки драйверов. Но как быть, если сетевая карта отсутствует в выводе доступных сетевых интерфейсов, а вы не помните точное название и модель вашей сетевой карты? Для таких случаев предназначена следующая команда:

```
lspci -knn | grep "Eth"
```

Вывод команды будет примерно следующим (рис. 15.3).

```
sergey@Test-PC: ~  
sergey@Test-PC:~$ lspci -knn | grep "Eth"  
01:00.0 Ethernet controller [0200]: Realtek Semiconductor Co., Ltd. RTL8111/  
8168/8411 PCI Express Gigabit Ethernet Controller [10ec:8168] (rev 11)  
sergey@Test-PC:~$
```

Рис. 15.3. Отображение информации о сетевом интерфейсе

Из вывода приведенной выше команды мы можем узнать, что у нас в компьютере используется сетевая карта Realtek, совместимая с моделями RTL8111/8168/8411. Теперь нам нужно зайти на официальный сайт производителя сетевой карты Realtek, адрес которого <http://www.realtek.com>, и в разделе **Downloads** произвести поиск по нашей модели сетевой карты (рис. 15.4).

Как видим, вторым результатом поиска представлены драйверы именно для нашей модели сетевой карты. Теперь, перейдя по найденной ссылке, выберите версию драйвера для ОС Linux и сохраните файл в домашнем каталоге.

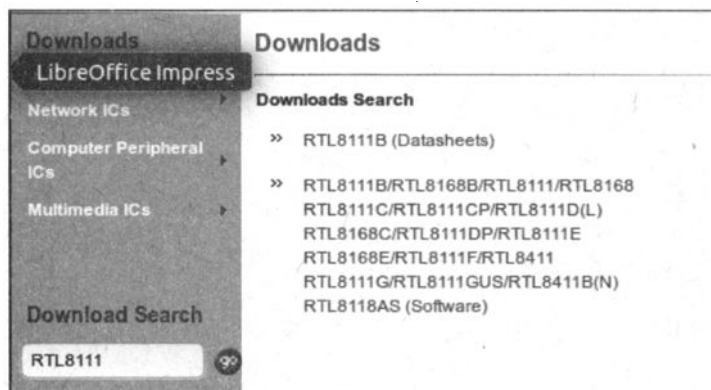


Рис. 15.4. Поиск драйвера для сетевой карты

Откройте Терминал и разархивируйте скачанный файл драйвера:

```
tar vjxf 0007-r8168-8.043.02.tar.bz2
```

ПРИМЕЧАНИЕ

В нашем случае файл архива с драйвером носит название 0007-r8168-8.043.02.tar.bz2. У вас имя архива может отличаться. Учтите этот факт при выполнении команд.

В результате выполнения команды в домашнем каталоге у вас появится каталог r8168-8.043.02.

Нам нужно убедиться в том, что у нас не установлен встроенный модуль сетевой карты. Для этого выполните следующую команду:

```
lsmod | grep r8168
```

ПРИМЕЧАНИЕ

Обратите внимание, что после слова `grep` идет часть названия нашего архива. Именно эта часть и является названием модуля драйвера сетевой карты.

Если при выполнении данной команды модуль был найден, то его следует удалить. Удаление модуля производится следующей командой:

```
sudo rmmod r8168
```

После этого перейдите в каталог с распакованным исходным кодом драйвера сетевой карты:

```
cd r8168-8.043.02
```

Нам нужно запустить скрипт установки драйвера:

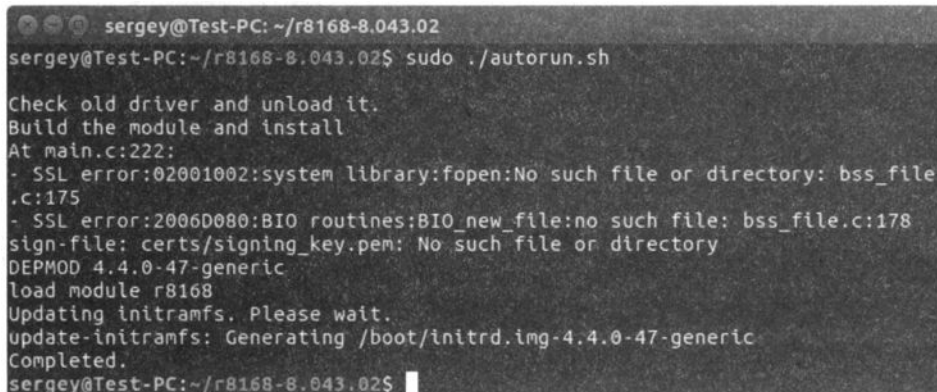
```
sudo ./autorun.sh
```

Если все прошло успешно, то через пару десятков секунд драйвер сетевой карты будет установлен (рис. 15.5).

Нам только осталось проверить, загружен ли установленный драйвер и появилась ли в списке интерфейсов наша сетевая карта:

```
lsmod | grep r8168
ifconfig -a
```

В первой команде мы проверяем, присутствует ли в списке модулей ядра модуль сетевой карты, а вторая команда отображает список доступных сетевых интерфейсов.



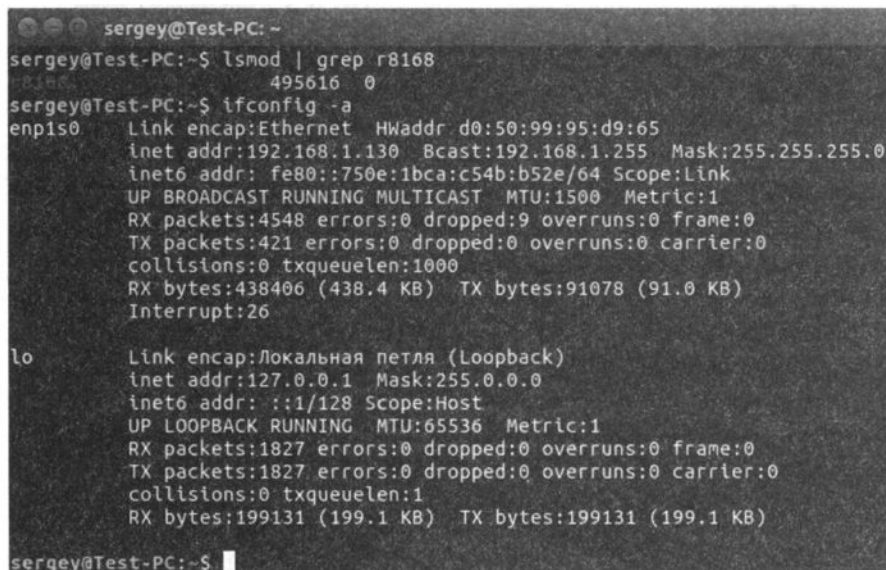
```
sergey@Test-PC: ~/r8168-8.043.02
sergey@Test-PC:~/r8168-8.043.02$ sudo ./autorun.sh

Check old driver and unload it.
Build the module and install
At main.c:222:
- SSL error:02001002:system library:fopen:No such file or directory: bss_file.c:175
- SSL error:2006D080:BI0 routines:BI0_new_file:no such file: bss_file.c:178
sign-file: certs/signing_key.pem: No such file or directory
DEPMOD 4.4.0-47-generic
load module r8168
Updating initramfs. Please wait.
update-initramfs: Generating /boot/initrd.img-4.4.0-47-generic
Completed.
sergey@Test-PC:~/r8168-8.043.02$
```

Рис. 15.5. Драйвер сетевой карты успешно установлен

Если все прошло удачно, у вас должна быть примерно следующая картина (рис. 15.6).

На этом заканчивается установка драйвера сетевой карты Realtek. Вам останется только настроить параметры подключения в соответствии с теми данными, которые



```
sergey@Test-PC: ~
sergey@Test-PC:~$ lsmod | grep r8168
r8168                495616  0
sergey@Test-PC:~$ ifconfig -a
enp1s0: Link encap:Ethernet  HWaddr d0:50:99:95:d9:65
        inet addr:192.168.1.130  Bcast:192.168.1.255  Mask:255.255.255.0
        inet6 addr: fe80::750e:1bca:c54b:b52e/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:4548 errors:0 dropped:9 overruns:0 frame:0
        TX packets:421 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:438406 (438.4 KB)  TX bytes:91078 (91.0 KB)
        Interrupt:26

lo:      Link encap:Локальная петля (Loopback)
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:1827 errors:0 dropped:0 overruns:0 frame:0
        TX packets:1827 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1
        RX bytes:199131 (199.1 KB)  TX bytes:199131 (199.1 KB)

sergey@Test-PC:~$
```

Рис. 15.6. Сетевая карта успешно установлена и определена в ОС

были выданы вам вашим провайдером. О том, как настраивать сетевое подключение, мы говорили в начале книги.

Установка драйвера беспроводной сети Broadcom

ОС Ubuntu не всегда удается корректно определить устройство для беспроводной сети, например, такая проблема часто возникает в ноутбуках, где беспроводная сеть отказывается работать без установки соответствующего драйвера.

В качестве примера мы рассмотрим установку драйвера для беспроводной сети Wi-Fi для сетевой карты Broadcom BCM4313.

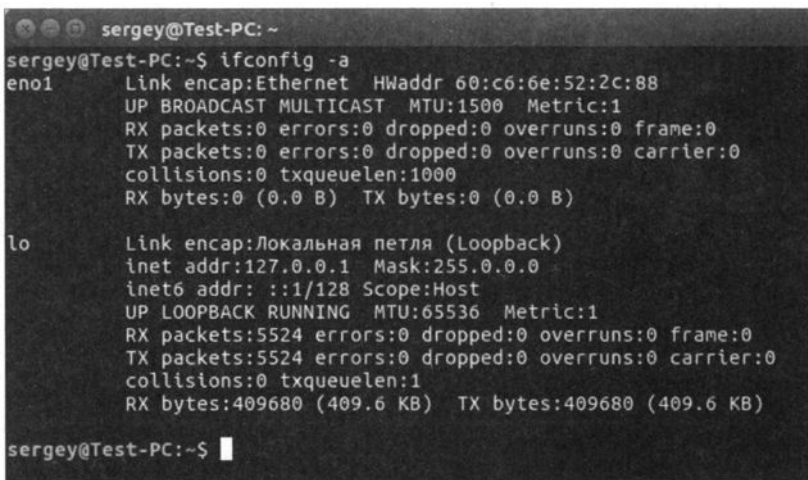
ПРИМЕЧАНИЕ

Сразу стоит сделать оговорку, что драйвер беспроводной сети BCM4313 успешно устанавливается системой по умолчанию, но в качестве примера мы его удалили.

Для начала нам нужно проверить, присутствует ли в списке интерфейсов сетевая карта для беспроводной сети:

```
ifconfig -a
```

Как мы можем убедиться, эта сетевая карта не отображается в списке интерфейсов (рис. 15.7).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ ifconfig -a  
eno1      Link encap:Ethernet  HWaddr 60:c6:6e:52:2c:88  
          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)  
  
lo        Link encap:Локальная петля (Loopback)  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING  MTU:65536  Metric:1  
          RX packets:5524 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:5524 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1  
          RX bytes:409680 (409.6 KB)  TX bytes:409680 (409.6 KB)  
  
sergey@Test-PC:~$
```

Рис. 15.7. Беспроводная сетевая карта отсутствует в списке интерфейсов

Отсутствие беспроводной сетевой карты в списке интерфейсов вовсе не означает, что ее нет физически. Поэтому теперь наша задача — выяснить название и ее модель:

```
lspci -knn | grep "Network"
```

А вот и показалась сама беспроводная карточка Wi-Fi (рис. 15.8).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ lspci -knn | grep "Network"  
07:00.0 Network controller [0280]: Broadcom Corporation BCM4313  
802.11bgn Wireless Network Adapter [14e4:4727] (rev 01)  
Subsystem: Hewlett-Packard Company BCM4313 802.11bgn Wir  
eless Network Adapter [103c:1795]  
sergey@Test-PC:~$
```

Рис. 15.8. Вывод названия и модели беспроводной сетевой карты

Теперь нам известны название и модель сетевой карты Wi-Fi — Broadcom BCM4313. Остается только отправиться на официальный сайт производителя <https://www.broadcom.com> и скачать сам драйвер устройства. Для этого лучше всего воспользоваться внутренним поисковым механизмом сайта, запустив поиск по модели устройства: BCM4313. После этого в результатах выдачи перейдите по найденной ссылке, а затем в соответствующий раздел загрузки драйверов — **Downloads**. На момент написания этих строк ссылка выглядела следующим образом:

<https://www.broadcom.com/products/wireless/wireless-lan-bluetooth/bcm4313>

Скачайте драйвер для ОС Linux, учитывая ее разрядность. Так как у нас 64-разрядная ОС Ubuntu, то соответственно мы скачиваем 64-битный драйвер, предварительно приняв лицензионное соглашение. Сохраните файл в домашнем каталоге. Наш загруженный файл называется `hybrid-v35_64-nodebug-pcoem-6_30_223_271.tar.gz`.

Теперь "подготовим" систему для последующей установки драйвера.

Первым делом нам необходимо удалить все лишнее, т. е. все то, что могло быть установлено вами либо операционной системой в попытках "заставить Wi-Fi работать". Проверим, загружены ли модули:

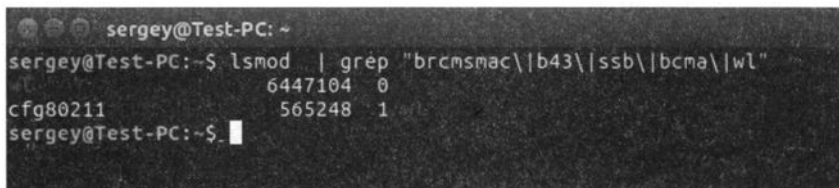
```
lsmod | grep "brcmsmac\|b43\|ssb\|bcma\|wl"
```

Этой командой мы проверяем, загружен ли в данный момент времени один из следующих модулей:

- ◆ `brcmsmac`;
- ◆ `b43`;
- ◆ `ssb`;
- ◆ `bcma`;
- ◆ `wl`.

Если по выполнению приведенной выше команды (рис. 15.9) вы обнаружили, что один или несколько модулей загружены и используются, то их нужно удалить соответствующей командой:

```
sudo rmmod brcmsmac  
sudo rmmod b43  
sudo rmmod ssb  
sudo rmmod bcma  
sudo rmmod wl
```



```

sergey@Test-PC: ~
sergey@Test-PC:~$ lsmod | grep "brcmsmac\|b43\|ssb\|bcma\|wl"
        6447104  0
cfg80211      565248  1
sergey@Test-PC:~$

```

Рис. 15.9. Обнаружен модуль wl

Вы можете выполнить все команды сразу, но целесообразней будет выполнить именно ту, которая удалит модуль, который у вас загружен и используется.

ПРИМЕЧАНИЕ

В нашем случае обнаружен используемый модуль wl, поэтому нам нужно удалить его командой `sudo rmmod wl`.

После удаления ненужных модулей нам следует добавить их в черный список для того, чтобы операционная система не пыталась загружать их:

```

echo "blacklist brcmsmac" >> sudo /etc/modprobe.d/blacklist.conf
echo "blacklist b43" >> sudo /etc/modprobe.d/blacklist.conf
echo "blacklist ssb" >> sudo /etc/modprobe.d/blacklist.conf
echo "blacklist bcma" >> sudo /etc/modprobe.d/blacklist.conf

```

Обратите внимание, что мы не добавляем в черный список модуль wl по той причине, что именно он нам потребуется для нормальной работы беспроводной сети Wi-Fi.

Подготовительные действия закончены, и теперь нужно распаковать загруженный драйвер. Давайте для удобства в домашнем каталоге пользователя создадим каталог wifi:

```
mkdir wifi
```

Распакуем содержимое архива `hybrid-v35_64-nodebug-pcoem-6_30_223_271.tar.gz` в только что созданный каталог wifi:

```
tar -xvf hybrid-v35_64-nodebug-pcoem-6_30_223_271.tar.gz -C /home/sergey/wifi/
```

Сделаем каталог wifi текущим:

```
cd wifi
```

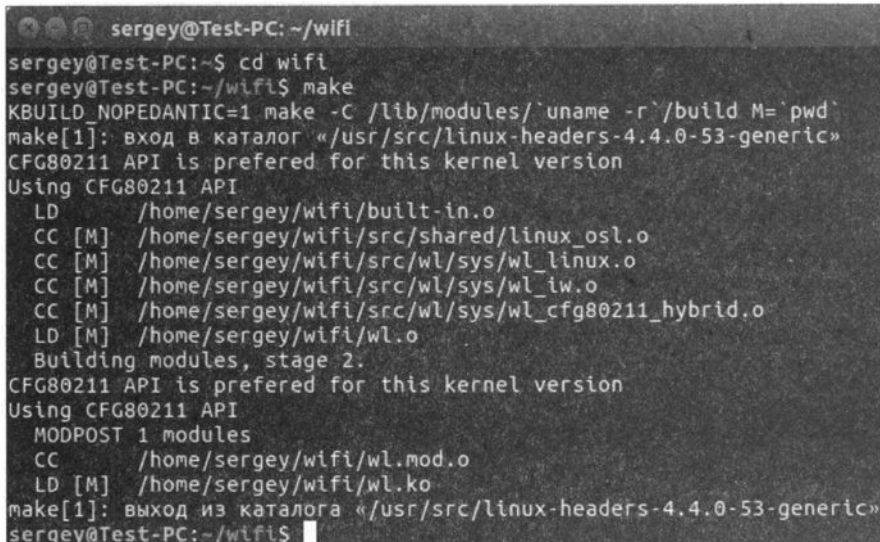
Теперь осталось запустить компиляцию драйвера:

```
make
```

Процесс компиляции драйвера пройдет за считанные секунды (рис. 15.10).

Выполнив команду `make`, мы только скомпилировали файлы драйвера, но сам драйвер еще не установлен. В результате компиляции в каталоге `/home/sergey/wifi/` будет создан файл `wl.ko`, который необходимо поместить в каталог модулей ядра. Для начала нам необходимо узнать версию используемого ядра Linux:

```
uname -r
```



```
sergey@Test-PC: ~/wifi
sergey@Test-PC:~$ cd wifi
sergey@Test-PC:~/wifi$ make
KBUILD_NOPEADANTIC=1 make -C /lib/modules/`uname -r`/build M='pwd'
make[1]: вход в каталог «/usr/src/linux-headers-4.4.0-53-generic»
CFG80211 API is preferred for this kernel version
Using CFG80211 API
LD      /home/sergey/wifi/built-in.o
CC [M]  /home/sergey/wifi/src/shared/linux_osl.o
CC [M]  /home/sergey/wifi/src/wl/sys/wl_linux.o
CC [M]  /home/sergey/wifi/src/wl/sys/wl_iw.o
CC [M]  /home/sergey/wifi/src/wl/sys/wl_cfg80211_hybrid.o
LD [M]  /home/sergey/wifi/wl.o
Building modules, stage 2.
CFG80211 API is preferred for this kernel version
Using CFG80211 API
MODPOST 1 modules
CC      /home/sergey/wifi/wl.mod.o
LD [M]  /home/sergey/wifi/wl.ko
make[1]: выход из каталога «/usr/src/linux-headers-4.4.0-53-generic»
sergey@Test-PC:~/wifi$
```

Рис. 15.10. Компиляция драйвера успешно завершена

Информация о версии ядра нам понадобится при выполнении следующей команды:

```
sudo cp /home/sergey/wifi/wl.ko /lib/modules/4.4.0-53-
generic/kernel/net/wireless/
```

Этой командой мы копируем скомпилированный файл `wl.ko` в каталог `/lib/modules/4.4.0-53-generic/kernel/net/wireless/`. Обратите внимание, что в каталоге `/lib/modules/` находится три каталога с разными версиями ядра. Поэтому перед выполнением команды копирования файла мы выяснили используемую версию ядра. В нашем случае — `4.4.0-53-generic`.

Создадим список зависимостей модулей:

```
sudo depmod
```

После этого добавим в ядро наш скопированный модуль `wl.ko`:

```
sudo modprobe wl
```

Осталось только перезагрузить операционную систему и проверить работоспособность беспроводного сетевого соединения Wi-Fi.

ПРИМЕЧАНИЕ

Перезагрузка ОС потребуется лишь в том случае, когда после добавления в ядро модуля беспроводное соединение не было установлено. Как правило, перезагрузка не требуется.

На этом этапе установки драйверов для беспроводной сети Wi-Fi Broadcom BCM4313 заканчивается.

Установка драйвера видеокарты из репозитория

Если ваша ОС Ubuntu загрузилась в графическом режиме, то это означает, что операционная система корректно распознала вашу видеокарту, и от вас не требуются дополнительные манипуляции для ее последующей настройки. Однако бывают ситуации, когда текущая версия драйвера работает крайне нестабильно. В этом случае вам ничего не остается, кроме как обновить драйвер.

При установке драйвера для видеокарты первым делом стоит обращать внимание на свободные драйверы, т. к. они разрабатываются с учетом специфики конкретной операционной системы, и все найденные ошибки оперативно исправляются сообществом. Несвободные (проприетарные) иногда могут работать нестабильно, т. к. исходный код этих драйверов закрыт от сообщества разработчиков, о чем и предупреждают разработчики ОС Ubuntu.

В этом разделе мы рассмотрим общий вариант установки драйвера видеокарты из PPA-репозитория.

ПРИМЕЧАНИЕ

Не лишним будет сказать, что рассматриваемый ниже репозиторий содержит разрабатываемые версии драйверов, которые тоже не всегда являются стабильными.

Итак, вне зависимости от наименования производителя видеокарты существует репозиторий с различными версиями драйверов для видеоустройств. Однако некоторые версии находятся в стадии разработки и тестирования и иногда могут работать нестабильно, но даже они могут быть полезными в случае невозможности работы с базовым видеодрайвером. Хватит вступительных слов, перейдем к делу.

Для обновления до последней версии видеодрайвера нам необходимо добавить новый источник, откуда будет загружена свежая версия видеодрайвера:

```
sudo add-apt-repository ppa:oibaf/graphics-drivers
```

На запрос ввода пароля введите свой пароль и для подтверждения изменения списка источников нажмите клавишу <Enter>. Убедиться в том, что мы добавили новый источник, можно на вкладке **Другое ПО** раздела **Программы и обновления** в окне **Параметры системы** (рис. 15.11).

Теперь нам нужно сообщить программе установки, что обновился список пакетов:

```
sudo apt-get update
```

Осталось обновить систему:

```
sudo apt-get dist-upgrade
```

После выполнения команды перезагрузите систему, и все обновления вступят в силу.

ПРИМЕЧАНИЕ

Если возникла ошибка при обновлении, то рекомендуется выполнить команду `sudo apt-get install -f` и повторить обновление списка пакетов и обновление системы.

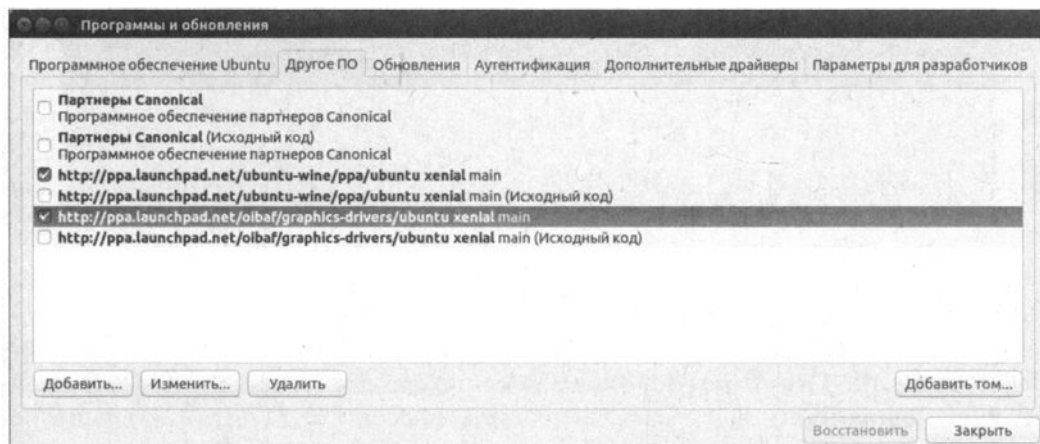


Рис. 15.11. Добавлен новый источник репозитория PPA

Это все, что касается установки более свежих драйверов, которые ставятся автоматически при обновлении системы. Помимо этого способа, существуют еще и другие варианты установки драйверов, о которых пойдет речь чуть ниже.

Общий метод установки видеодрайвера AMD/ATI Radeon

В ранних версиях ОС Ubuntu, до версии 16.04, существовала возможность установить проприетарный драйвер AMD Catalyst, который в мире Linux известен как `fglrx`. Однако в связи с тем что производитель драйвера AMD решил в корне поменять концепцию для своего драйвера, компания Canonical решила запретить драйвер `fglrx` в версии ОС Ubuntu 16.04. Вместо этого теперь предлагается установить драйвер `radeon` или `amdgpu`. Разумеется, никто не запрещает вам самостоятельно установить драйвер `fglrx`, предварительно проверив перед этим совместимость, но компания Canonical просто перестала его поддерживать и хранить в своих репозиториях.

ПРИМЕЧАНИЕ

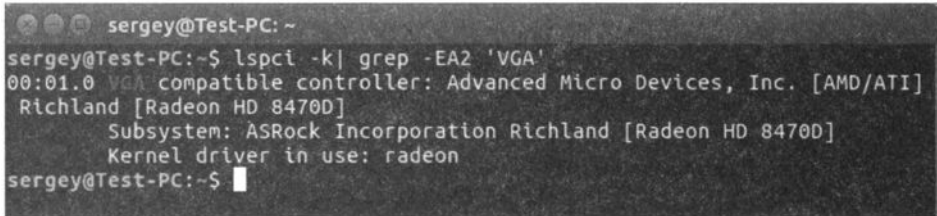
В связи с появлением открытого драйвера `amdgpu` и нестабильной работой `fglrx` в ОС Linux драйвер `fglrx` стали массово исключать из Linux-дистрибутивов, поэтому его установку мы рассматривать не будем.

Итак, для начала нам нужно узнать модель видеокарты и выяснить используемый видеодрайвер:

```
lspci -k | grep -EA2 'VGA'
```

Выполнение данной команды позволит нам получить информацию о видеокарте и об используемом видеодрайвере (рис. 15.12).

- У нас имеется видеокарта Radeon HD 8470D и используется драйвер `radeon`. Данный драйвер, точнее его модуль, используется ядром операционной системы и не является проприетарным драйвером.



```
sergey@Test-PC: ~
sergey@Test-PC:~$ lspci -k | grep -EA2 'VGA'
00:01.0 VGA compatible controller: Advanced Micro Devices, Inc. [AMD/ATI]
Richland [Radeon HD 8470D]
Subsystem: ASRock Incorporation Richland [Radeon HD 8470D]
Kernel driver in use: radeon
sergey@Test-PC:~$
```

Рис. 15.12. Получение информации о видеокарте и используемом видеодрайвере

Первым делом мы должны проверить, поддерживается ли наша видеокарта драйвером `amdgpu`. Для этого на странице <http://support.amd.com/en-us/kb-articles/Pages/AMDGPU-PRO-Driver-for-Linux-Release-Notes.aspx> в разделе **AMD Radeon Product Compatibility** ищем нашу видеокарту Radeon HD 8470D. Как видим, на данный момент драйвером `amdgpu` поддерживаются только новейшие видеокарты серии Radeon RX, Radeon R9, Radeon R7, Radeon Pro WX-series, AMD FirePro W-Series и AMD FirePro S-Series. Нашей видеокарты в этом списке нет, значит, немного "не повезло", и нам не удастся воспользоваться новой версией `amdgpu`. По этой причине для видеокарты Radeon HD 8470D лучшим решением будет оставить стандартный драйвер `radeon`.

Но как же быть, если у вас установлена видеокарта, которая поддерживается драйвером `amdgpu`? Давайте рассмотрим вариант установки `amdgpu` на поддерживаемую видеокарту.

После того как мы узнали модель видеокарты, нам нужно отправиться на официальный сайт производителя <http://www.amd.com> и в строке поиска выполнить запрос по слову *amdgpu*.

ПРИМЕЧАНИЕ

На момент написания этих строк самой последней версией драйвера `amdgpu` была версия 16.15, которая расположена по ссылке <http://support.amd.com/en-us/kb-articles/Pages/AMDGPU-PRO-Driver-for-Linux-Release-Notes.aspx>.

В разделе загрузки выберите ссылку **AMDGPU-Pro Driver Version 16.50 for Ubuntu 16.04**. Сохраните файл в домашнем каталоге. Теперь наш загруженный файл `amdgpu-pro-16.50-362463.tar.xz` нужно разархивировать:

```
tar -Jxvf amdgpu-pro-16.50-362463.tar.xz
```

Далее необходимо перейти в каталог с распакованными файлами:

```
cd amdgpu-pro-16.50-362463
```

и выполнить установку видеодрайвера:

```
./amdgpu-pro-install
```

Терминал запросит у вас пароль. Для подтверждения установки введите `y` и нажмите клавишу `<Enter>` (рис. 15.13).

После того как драйвер установится, выполните следующую команду:

```
sudo usermod -a -G video $LOGNAME
```

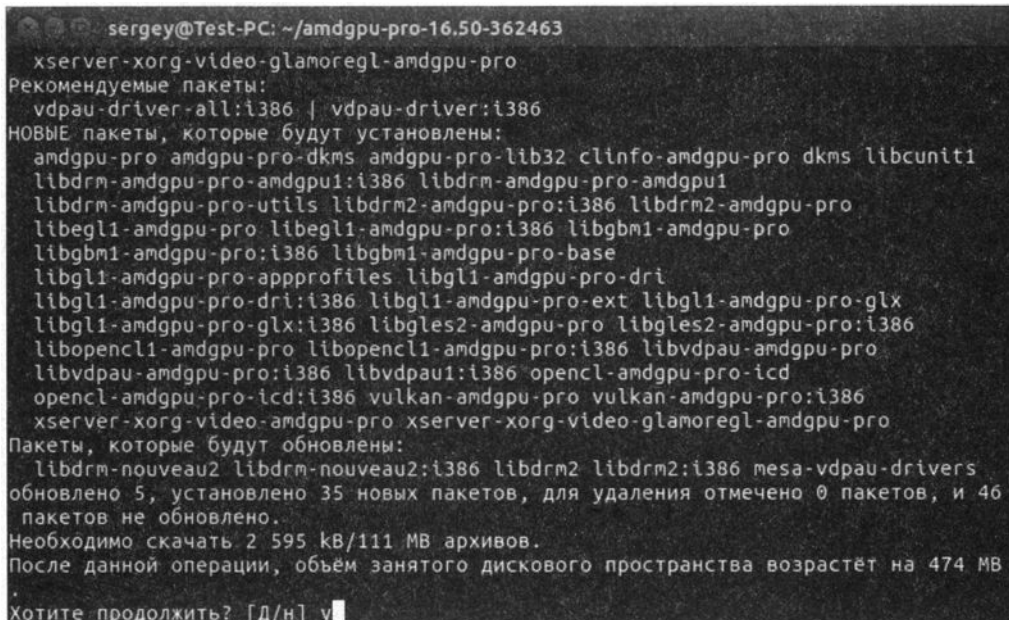


Рис. 15.13. Терминал запрашивает подтверждение установки

Эта команда позволит добавить текущего пользователя в группу **video**. Осталось только перезагрузить систему:

```
shutdown -r now
```

Таким способом выполняется установка открытого драйвера amdgru.

В случае возникновения проблем в работе драйвера, например когда невозможно выполнить вход в графическую оболочку, следует удалить установленный видеодрайвер amdgru. Делается это следующим образом: переключитесь в консольный режим (сочетание клавиш <Ctrl>+<Alt>+<F1>), затем после ввода имени и пароля выполните следующую команду:

```
sudo amdgpu-pro-uninstall
```

Подтвердите удаление видеодрайвера amdgru и после удаления перезагрузите операционную систему.

Установка аудиодрайвера

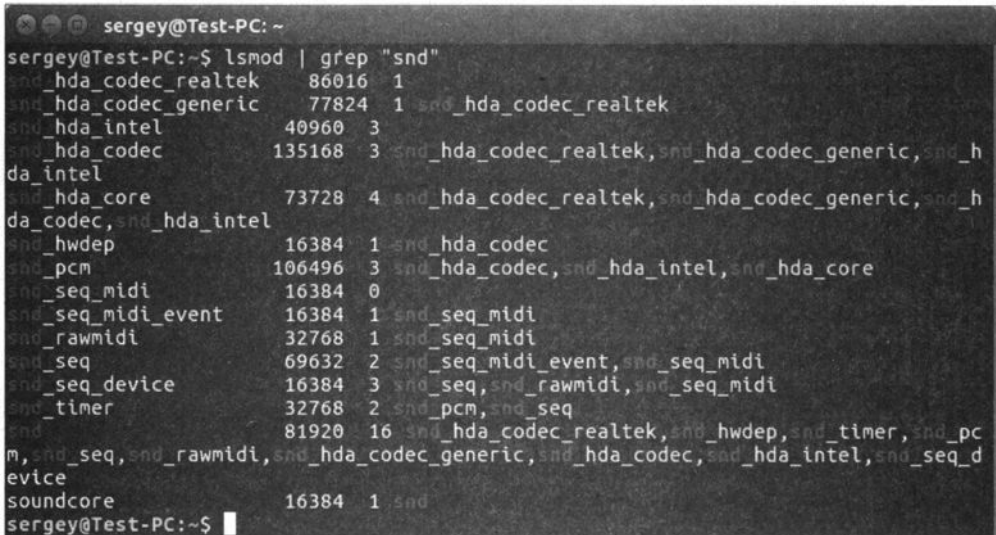
В Linux-подобных операционных системах за звук отвечает один-единственный аудиодрайвер, который носит название ALSA.

ALSA (Advanced Linux Sound Architecture — продвинутая звуковая архитектура Linux) — это целая архитектура звуковых драйверов, обеспечивающая поддержку огромного количества звуковых карт, начиная от старых моделей и заканчивая продвинутыми современными моделями.

Да, вы не ослышались, ALSA способна работать практически со всеми аудиокартами независимо от того или иного производителя. Это избавляет вас от поиска нужного драйвера для вашей аудиокарты. Вам нужно всего лишь установить ALSA, и аудиокарта заработает так, будто вы установили ее "родной драйвер".

Набор звуковых драйверов ALSA пришел на смену Open Sound System (OSS), который не смог конкурировать со свободным набором звуковых драйверов ALSA.

В большинстве случаев дополнительно настраивать или устанавливать ALSA не требуется, т. к. все работает по умолчанию (рис. 15.14).



```

sergey@Test-PC: ~
sergey@Test-PC:~$ lsmod | grep "snd"
snd_hda_codec_realtek      86016  1
snd_hda_codec_generic     77824  1 snd_hda_codec_realtek
snd_hda_intel              40960  3
snd_hda_codec             135168  3 snd_hda_codec_realtek,snd_hda_codec_generic,snd_h
da_intel
snd_hda_core               73728  4 snd_hda_codec_realtek,snd_hda_codec_generic,snd_h
da_codec,snd_hda_intel
snd_hwdep                  16384  1 snd_hda_codec
snd_pcm                    106496  3 snd_hda_codec,snd_hda_intel,snd_hda_core
snd_seq_midi               16384  0
snd_seq_midi_event         16384  1 snd_seq_midi
snd_rawmidi                32768  1 snd_seq_midi
snd_seq                    69632  2 snd_seq_midi_event,snd_seq_midi
snd_seq_device             16384  3 snd_seq,snd_rawmidi,snd_seq_midi
snd_timer                  32768  2 snd_pcm,snd_seq
snd                         81920  16 snd_hda_codec_realtek,snd_hwdep,snd_timer,snd_pc
m,snd_seq,snd_rawmidi,snd_hda_codec_generic,snd_hda_codec,snd_hda_intel,snd_seq_d
evice
soundcore                  16384  1 snd
sergey@Test-PC:~$

```

Рис. 15.14. Звуковая карта была успешно определена при установке ОС

При выводе команды `lsmod | grep "snd"` все аудиодрайверы представлены в виде модулей ядра, названия которых начинаются с префикса `snd_`.

В случае если у вас возникают проблемы в работе со звуком, то для начала рекомендуется обновить ALSA из репозитория, который предоставляет Ubuntu Audio Development Team:

```
sudo apt-add-repository ppa:ubuntu-audio-dev/ppa
```

Эта команда добавляет в стандартный репозиторий список источников на более свежую версию ALSA.

После изменения списка репозитория нужно обновить этот список:

```
sudo apt-get update
```

В ответ на эту команду будет загружен новый список аудиодрайверов из только что добавленного репозитория. Осталось только обновить систему:

```
sudo apt-get upgrade
```

Далее перезагружаем операционную систему и проверяем работоспособность звукового устройства.

Установка ALSA из исходных текстов

Сейчас мы рассмотрели вариант установки ALSA из репозитория, однако существует возможность установки звуковой архитектуры ALSA из исходных текстов, которые нужно сначала скачать, а затем скомпилировать и уже только потом настроить работу звукового устройства.

ПРИМЕЧАНИЕ

К данному способу установки звуковой архитектуры ALSA следует прибегать лишь в крайних случаях, т. к. этот способ является не лучшим решением проблемы и зачастую не избавляет от всех проблем, связанных с воспроизведением звука.

Для начала зайдём на страницу проекта ALSA по адресу <http://alsa-project.org> и в левом меню выберем ссылку **Download**. На этой странице нам будут представлены следующие пакеты:

- ◆ **Drivers** — пакет содержит компоненты модуля ядра. На данный момент загружать данный пакет не требуется, т. к. разработчики ядра тесно сотрудничают с разработчиками ALSA, и все компоненты ALSA сразу же включаются в ядро ОС Linux;
- ◆ **Firmware (alsa-firmware)** — данный пакет содержит бинарные драйверы для различных производителей аудиоустройств;
- ◆ **Library (alsa-lib)** — пакет содержит пользовательские библиотеки, которые требуются для запуска компонентов звуковой архитектуры ALSA;
- ◆ **Plugins (alsa-plugins)** — пакет содержит плагины для поддержки дополнительных возможностей звуковой архитектуры;
- ◆ **Utilities (alsa-utils)** — пакет содержит набор утилит, которые позволяют управлять настройкой звуковой архитектуры ALSA;
- ◆ **Tools (alsa-tools)** — пакет содержит инструменты, предназначенные для более тонкой настройки звуковой архитектуры ALSA;
- ◆ **PuALSA (pyalsa)** — пакет содержит необходимые зависимости для работы некоторых библиотек звуковой архитектуры ALSA;
- ◆ **OSS compat lib (alsa-oss)** — пакет содержит в себе набор библиотек, которые необходимы для обеспечения совместимости с устаревшей Open Sound System (OSS).

Нам необходимы только 4 пакета: **Firmware**, **Library**, **Plugins** и **Utilities**.

ПРИМЕЧАНИЕ

При желании, если вы являетесь истинным ценителем звука и любите более детально настраивать его качество, то вам понадобится пакет **Tools**. Мы же его рассматривать не будем, т. к. его установка ничем не отличается от установки других пакетов звуковой архитектуры ALSA.

Скачайте необходимые пакеты и сохраните их в домашнем каталоге для удобства их дальнейшей установки. В итоге мы получим 4 файла:

- ◆ `alsa-firmware-1.0.29.tar.bz2`;
- ◆ `alsa-lib-1.1.2.tar.bz2`;
- ◆ `alsa-plugins-1.1.1.tar.bz2`;
- ◆ `alsa-utils-1.1.2.tar.bz2`.

Распакуем эти архивы:

```
tar jxvf alsa-firmware-1.0.29.tar.bz2
tar jxvf alsa-lib-1.1.2.tar.bz2
tar jxvf alsa-plugins-1.1.1.tar.bz2
tar jxvf alsa-utils-1.1.2.tar.bz2
```

После выполнения этих команд в домашнем каталоге будет создано 4 одноименных каталога. Как вы уже догадались, сейчас мы будем устанавливать каждый пакет по отдельности.

Сначала установим пакет Firmware (`alsa-firmware`):

```
cd alsa-firmware-1.0.29
./configure
make
sudo make install
```

Первая команда делает текущим каталог `alsa-firmware-1.0.29`, вторая команда выполняет конфигурацию для подготовки к компиляции. Третья команда подготавливает исходные файлы к установке и передает управление последней команде, которая уже производит установку скомпилированных файлов.

То же самое проделываем и с пакетом Library (`alsa-lib`):

```
cd alsa-lib-1.1.2
./configure
make
sudo make install
```

Теперь нужно установить плагины, т. е. пакет Plugins (`alsa-plugins`):

```
cd alsa-plugins-1.1.1
./configure
make
sudo make install
```

И последним шагом будет установка утилит Utilities (`alsa-utils`). Для начала нужно **выгрузить службу `alsa`**:

```
sudo /sbin/alsa unload
```

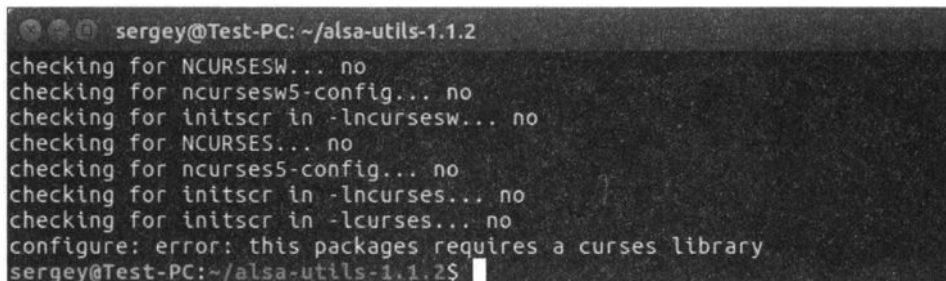
Перейдем в каталог с исходными текстами утилиты:

```
cd alsa-utils-1.1.2
```

А теперь начинается самое интересное — попытаемся скомпилировать наш набор утилит:

```
./configure
```

Не так тут все просто. Ошибочка вышла (рис. 15.15).



```
sergey@Test-PC: ~/alsa-utils-1.1.2
checking for NCURSES... no
checking for ncursesw5-config... no
checking for initscr in -lncursesw... no
checking for NCURSES... no
checking for ncurses5-config... no
checking for initscr in -lncurses... no
checking for initscr in -lcurses... no
configure: error: this packages requires a curses library
sergey@Test-PC: ~/alsa-utils-1.1.2$
```

Рис. 15.15. Требуется установка библиотеки curses

Ошибка "this packages requires a curses library" означает, что для компиляции данного пакета требуется установленная в системе библиотека curses. Сейчас мы ее установим:

```
sudo apt-get install libncurses5-dev
```

Если сейчас попытаетесь еще раз выполнить команду `./configure`, то все пройдет без ошибок, но до выполнения следующей команды установки. Поэтому давайте установим недостающие пакеты:

```
sudo apt install gawk
sudo apt install xmlto
```

Первая команда установит реализацию языка программирования AWK, а вторая — внешний интерфейс для инструментального набора XSL toolchain.

ПРИМЕЧАНИЕ

Пакет `xmlto` имеет большой размер — примерно 400 Мбайт, и его установка может занять некоторое время.

Вот теперь настала очередь завершить процесс установки пакета Utilities (`alsa-utils`). Выполните команду конфигурирования:

```
./configure
```

Как видите, теперь команда выполнилась без ошибок, и можно перейти к выполнению второй команды:

```
make
```

Все файлы готовы к установке:

```
sudo make install
```

Если сообщения об ошибках не появились, то это значит, что установка ALSA прошла успешно.

Теперь можно проверить работоспособность звука или настроить звучание в утилите `AlsaMixer`, которая запускается по одноименной команде (рис. 15.16).

По умолчанию `AlsaMixer` отображает уровень громкости звуковой архитектуры `PulseAudio`, и если нажать клавишу `<F6>`, то можно выбрать звуковую карту и произвести более детальную настройку звука (рис. 15.17).

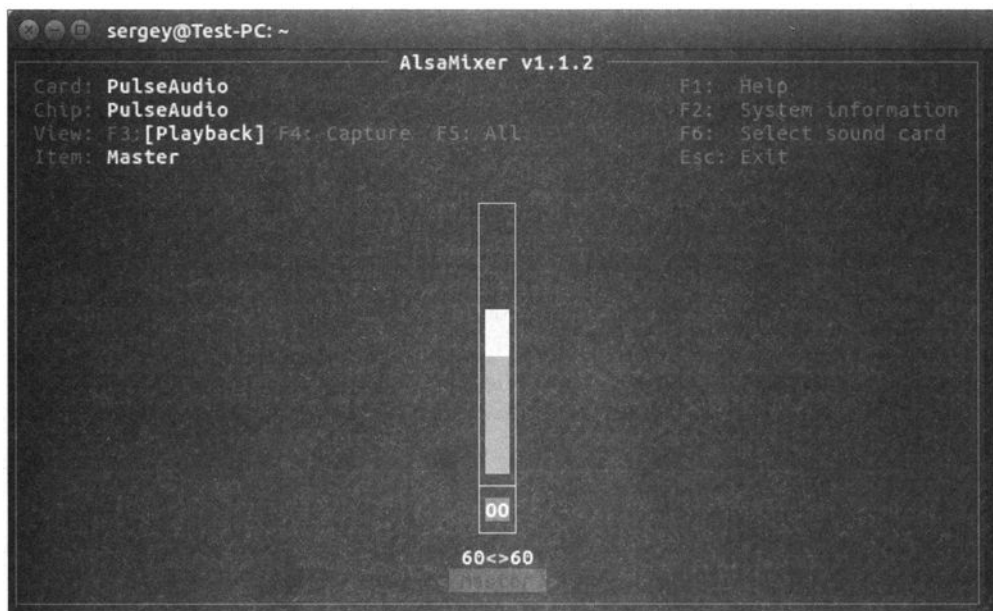


Рис. 15.16. Окно утилиты AlsaMixer

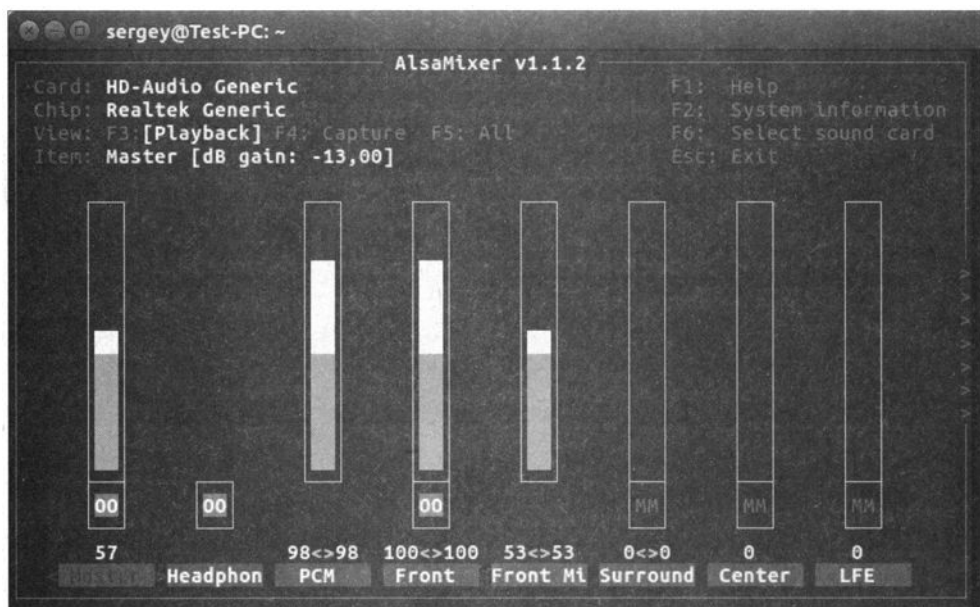


Рис. 15.17. Настройка звуковой карты в AlsaMixer

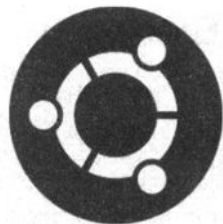
Подобным образом устанавливается и настраивается звуковая архитектура ALSA. При этом стоит учитывать, что сам процесс установки и настройки ALSA не всегда может быть завершен успешно в силу специфики той или иной модели звуковой карты вашего компьютера.

Выводы

Мы рассмотрели довольно сложную тему установки драйверов для компьютера в ОС Ubuntu Linux.

Мы узнали, что для ОС Ubuntu Linux существуют как свободные, так и проприетарные драйверы, а также познакомились со способами их установки. Например, мы научились устанавливать драйвер видеокарты из репозитория, а также драйверы из исходных текстов.

ГЛАВА 16



Установка принтера в ОС Ubuntu

Об установке принтеров, сканеров и МФУ

Установка и настройка принтеров в ОС Ubuntu Linux значительно отличается от установки в ОС Windows. Основное различие состоит не только в самом процессе установки, а в том, что не всегда удастся найти подходящий драйвер принтера. В связи с этим часто приходится искать некое промежуточное решение. Так, вместо какой-то конкретной модели принтера приходится устанавливать драйвер от другой модели, которая схожа по функционалу за исключением некоторых функций. По этой причине часто приходится довольствоваться лишь базовыми возможностями, предоставляемыми драйвером принтера для той или иной модели.

Однако стоит отметить, что производители современных моделей принтеров часто стараются удовлетворить потребности пользователей различных операционных систем. Так, именитые производители принтеров выпускают драйверы для своих устройств не только для ОС Windows, но и для ОС Linux, в частности для ОС Ubuntu Linux.

Практически любой принтер можно установить и настроить для работы в ОС Ubuntu Linux, даже при отсутствии нужных драйверов для вашей модели принтера. По умолчанию ОС Ubuntu без труда сможет распознать наиболее распространенные модели принтеров и обеспечить их корректную работу даже без установки драйверов от соответствующей модели устройства.

В данной главе мы рассмотрим как установку принтера с помощью драйверов от производителя, так и установку принтера в систему без наличия соответствующих драйверов. Материал будет относиться не только к установке принтера и сканера, но и таких устройств, как МФУ (многофункциональные устройства).

Где взять необходимые драйверы?

При поиске драйверов любого устройства первым делом нужно обращаться на сайт производителя вашего устройства. Если там нет нужных драйверов, тогда следует изучить материалы Интернета. Возможно, решение вашей проблемы уже имеется,

и вам нужно только следовать инструкциям, которые предлагают более опытные пользователи.

ПРИМЕЧАНИЕ

Перед покупкой принтера той или иной модели желательно почитать отзывы в Интернете касаясь этой модели и убедиться в том, что она будет работать с вашей операционной системой.

Итак, для начала приведем общие инструкции по наиболее популярным производителям устройств, а затем на конкретных примерах рассмотрим установку соответствующего оборудования. Изучать установки устройств каждого производителя принтеров мы не будем, т. к. эти процессы имеют схожие черты, а приведем лишь краткие инструкции по основным производителям, а затем рассмотрим установку принтера Epson L800 и МФУ Epson Stylus CX5900.

Устройства Brother

Перейдите на сайт производителя устройства: <http://www.brother.com>. Затем выберите раздел **Product Support**, а в нем страну, на языке которой вам будет удобнее работать с сайтом. Выполните поиск по модели устройства или вручную введите модель вашего принтера и следуйте инструкциям, которые приводит производитель вашего устройства.

Устройства Canon

Перейдите на сайт производителя устройства: <http://www.canon.ru>. Затем перейдите в раздел **Поддержка**. Выполните поиск по модели устройства или вручную введите модель вашего принтера и следуйте инструкциям, которые приводит производитель вашего устройства.

Устройства Epson

Перейдите на сайт производителя устройства: <http://global.epson.com>. Выберите раздел **Products & Drivers**, а затем раздел **Printing Solutions**. Щелкните по ссылке **Driver downloads**, вручную введите модель вашего принтера и следуйте инструкциям, которые приводит производитель вашего устройства.

Устройства HP

Перейдите на сайт с драйверами для устройств HP: <http://hplipopensource.com>. Затем перейдите в раздел **Download** и скачайте программу HP Linux Imaging and Printing. Данная программа позволит установить и настроить работу устройств HP.

Устройства Samsung

Перейдите на сайт производителя устройства: <http://www.samsung.com>. Затем перейдите на главную страницу поддержки, щелкнув по ссылке **Поддержка**, а затем **Главная**. Выполните поиск по модели устройства или вручную введите модель

вашего принтера и следуйте инструкциям, которые приводит производитель вашего устройства.

Устройства Xerox

Перейдите на сайт производителя устройства: <https://www.xerox.com>. Затем перейдите в раздел **Support and Drivers**, щелкнув по ссылке **Support** и в выпадающем меню выбрав **Support and Drivers**. Выполните поиск по модели устройства или вручную введите модель вашего принтера и следуйте инструкциям, которые приводит производитель вашего устройства.

Установка службы CUPS

В ОС Ubuntu Linux за работу с принтерами отвечает служба CUPS, без которой вы просто не сможете выполнить печать нужного вам документа.

Служба CUPS (Common UNIX Printing System, общая UNIX-система печати) — это сервер печати для UNIX-подобных операционных систем. Служба CUPS позволяет управлять заданиями печати, обеспечивая сетевую печать с помощью протокола IPP (Internet Printing Protocol, протокол межсетевой печати).

Таким образом, без установленной службы CUPS вы просто не сможете добавить принтер в систему (рис. 16.1).

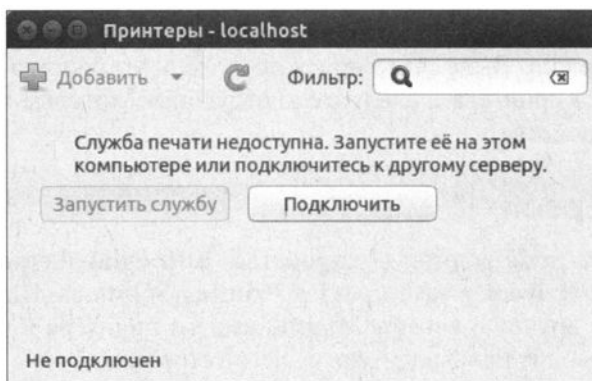


Рис. 16.1. Служба CUPS еще не установлена

Для начала система предлагает подключить службу печати, т. е. службу CUPS. Однако та еще не установлена, и даже если нажать кнопку **Подключить**, то ничего хорошего из этого не выйдет. Система просто не найдет эту службу и выдаст ошибку подключения. Поэтому сначала ее установим. Выполните в Терминале следующую команду:

```
sudo apt install cups
```

В результате будет установлена служба CUPS. Теперь можно переходить к процессу установки принтера.

ПРИМЕЧАНИЕ

В случае возникновения проблем с установкой службы CUPS или при ее дальнейшей работе выполните полное удаление этой службы командой `sudo apt purge cups`. После этого заново установите эту службу.

Установка принтера Epson L800 средствами ОС

Сейчас мы рассмотрим вариант установки принтера Epson L800 средствами ОС Ubuntu Linux, т. е. без соответствующих драйверов. У вас уже должна быть установлена служба CUPS.

Для начала включите ваш принтер и подключите принтер с помощью USB-кабеля. После чего откройте **Параметры системы** и в разделе **Оборудование** нажмите на значок **Принтеры**. Откроется окно установленных в систему принтеров (рис. 16.2).

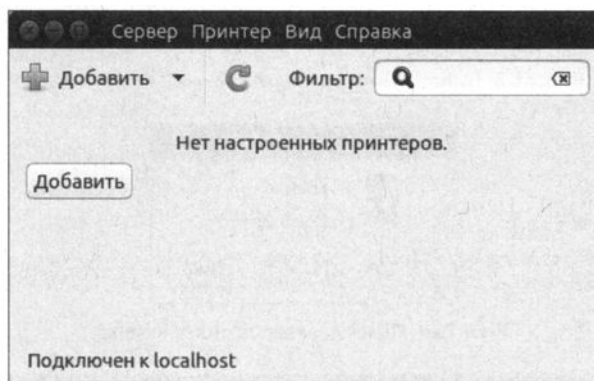


Рис. 16.2. Принтеры, установленные в ОС

В нашем случае установленные принтеры отсутствуют. Для того чтобы установить принтер, нажмите кнопку **Добавить**. Откроется окно выбора принтера (рис. 16.3).

В левой части окна выберите принтер **Epson L800** и нажмите кнопку **Вперед**. Запустится поиск необходимых драйверов для устройства (рис. 16.4).

По окончании поиска появится окно выбора драйвера для принтера (рис. 16.5).

В предоставленном списке выберите наименование производителя устройства, в нашем случае это **Epson**, и нажмите кнопку **Вперед**.

Откроется окно выбора модели принтера (рис. 16.6).

Здесь начинается самое интересное. В предоставленном списке моделей принтеров не значится нашей модели, однако даже из этой ситуации есть выход.

Наверняка каждый из вас знает, что многие устройства имеют несколько схожих моделей, которые отличаются друг от друга некоторыми функциями, например, отсутствием дисплея, более высокой скоростью печати и т. п. Все это нам дает определенные преимущества: мы можем установить драйвер от похожей модели

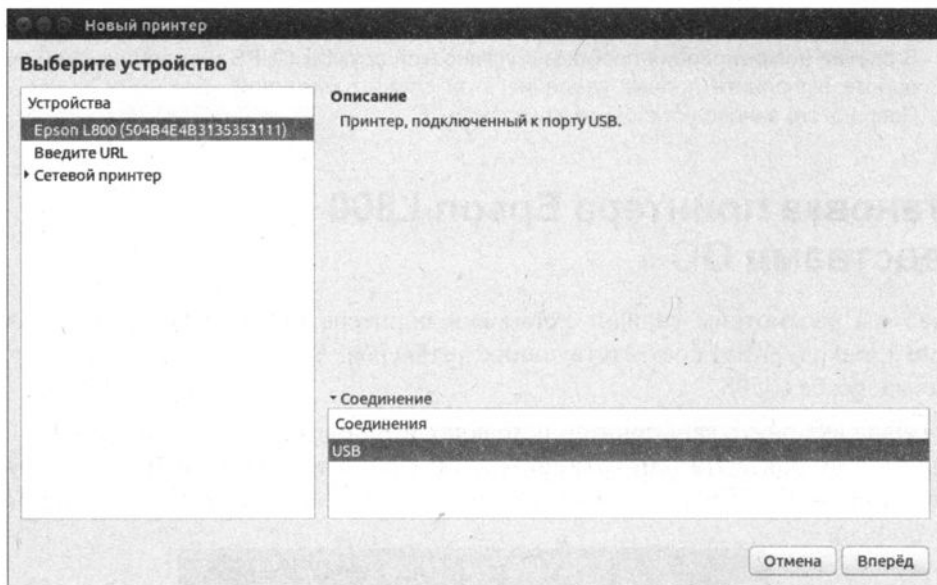


Рис. 16.3. Выбор принтера, подключенного к компьютеру

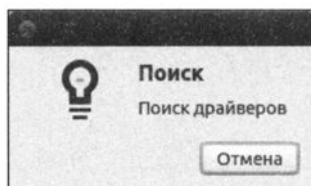


Рис. 16.4. Поиск драйверов для принтера

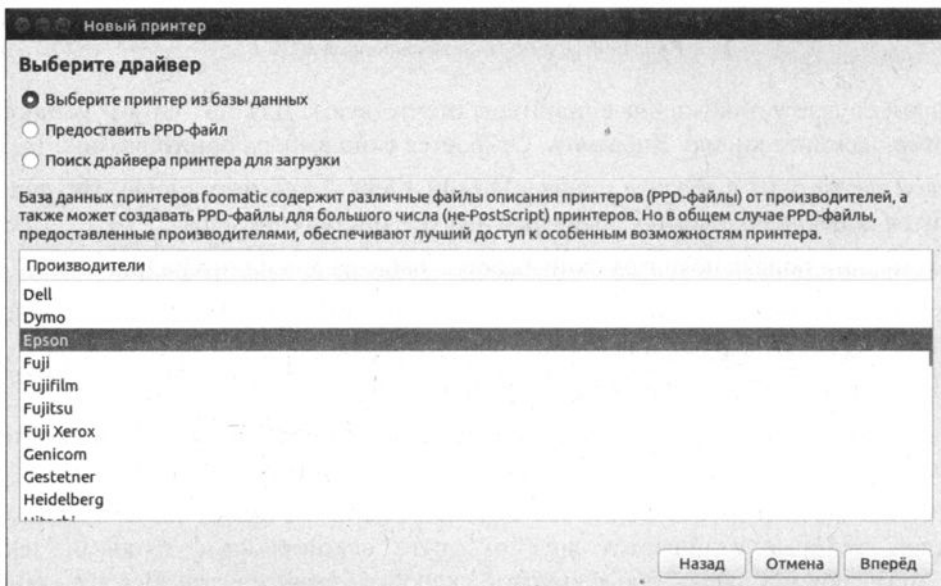


Рис. 16.5. Окно выбора драйвера для принтера

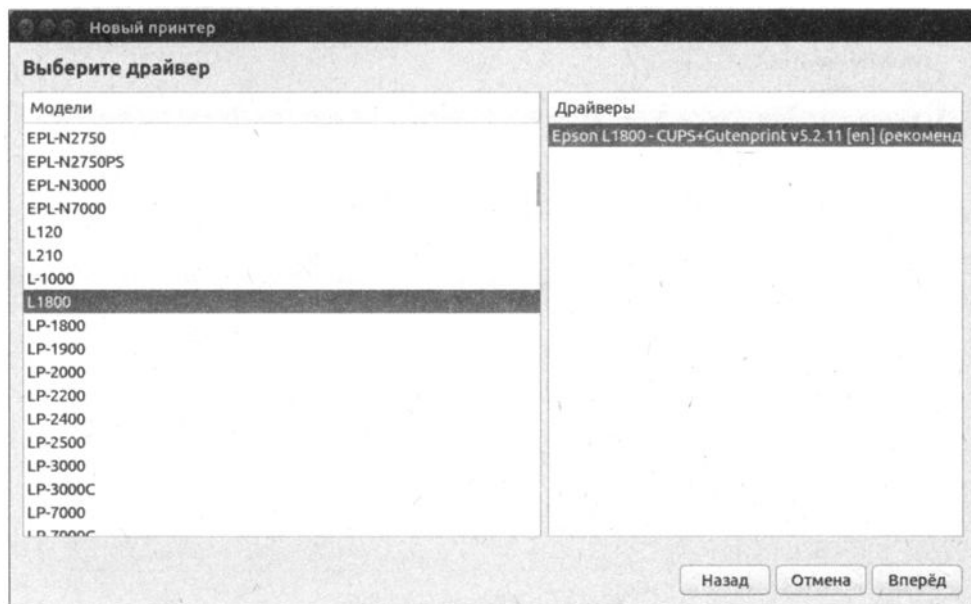


Рис. 16.6. Выбор модели принтера

принтера и работать с устройством без каких-либо затруднений. Конечно, некоторые функции принтера могут отсутствовать в этом случае, но, тем не менее, принтер будет работоспособен, и им можно пользоваться практически так же, как и с "родными" драйверами.

Так как наша модель принтера L800 отсутствует в списке, можно выбрать модель **L1800** и нажать кнопку **Вперед**.

Следующим этапом система предложит задать имя принтера, его описание и расположение (рис. 16.7).

Вы вправе ввести свои данные, но можно оставить все как есть и нажать кнопку **Применить**.

По окончании настройки принтера вам будет предложено напечатать пробную страницу, которая позволит оценить правильность установки устройства в операционной системе (рис. 16.8).

После этого, если вы выбрали печать пробной страницы, появится окно, сообщающее о том, что тестовая страница отправлена в очередь печати (рис. 16.9).

Нажмите кнопку **ОК**. Завершающим этапом установки принтера Epson L800 будет появившееся окно свойств локального принтера (рис. 16.10).

Нажмите кнопку **ОК**.

Данное окно можно вызвать и в последующей работе с принтером, открыв **Параметры системы**, а затем меню **Принтеры**. После этого в списке установленных принтеров выделить нужный принтер и в контекстном меню, вызываемом правой кнопкой мыши, выбрать пункт меню **Свойства** (рис. 16.11).

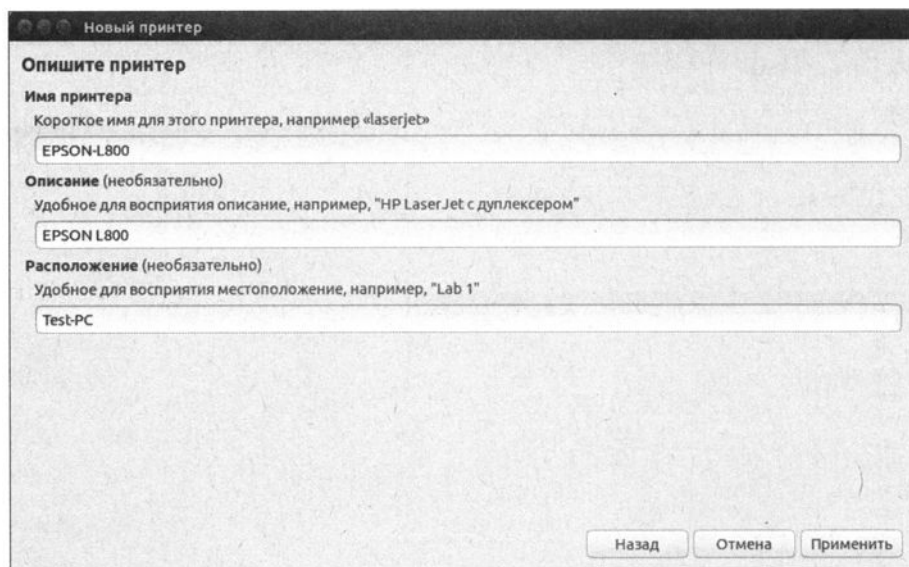


Рис. 16.7. Описание принтера

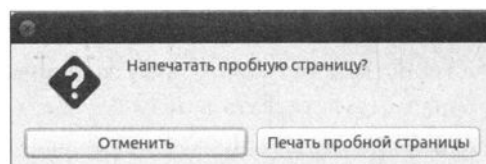


Рис. 16.8. Печать пробной страницы

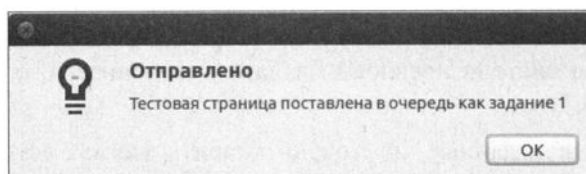


Рис. 16.9. Тестовая страница отправлена в очередь печати

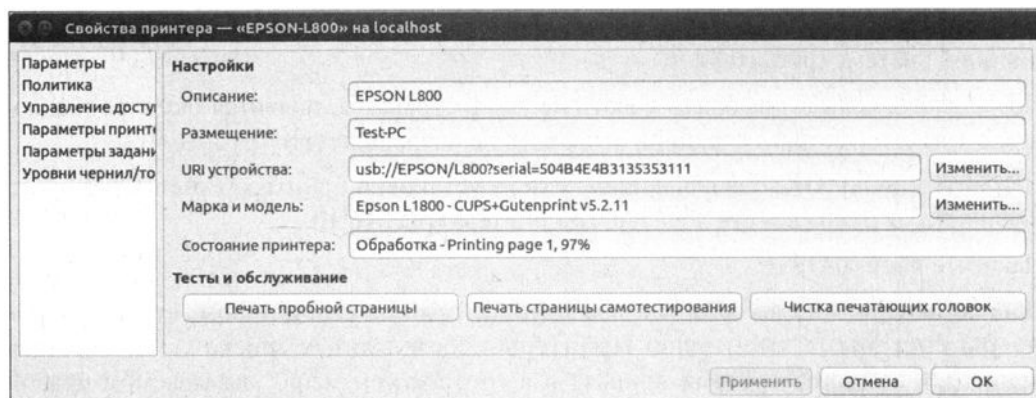


Рис. 16.10. Окно свойств локального принтера

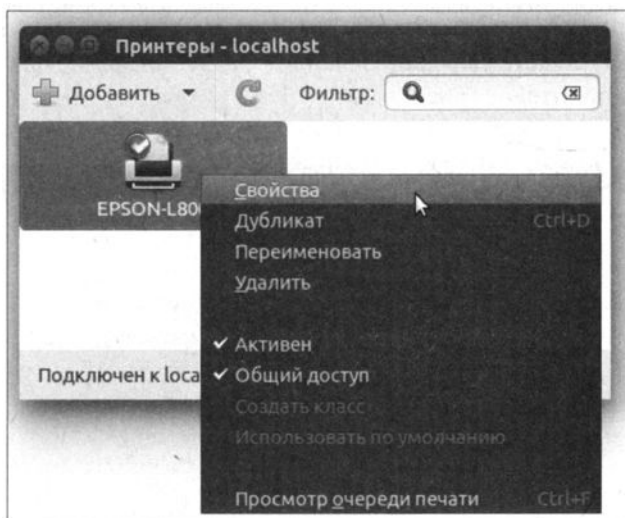


Рис. 16.11. Меню свойств установленного принтера

На этом процесс установки принтера Epson L800 закончен. Теперь он полностью работоспособен.

Настройка общего доступа

Часто случается так, что с одним принтером работает несколько пользователей с разных компьютеров. Данные компьютеры соединены между собой в сеть и для совместного доступа им просто необходимо иметь равный доступ к принтеру, который физически может быть подключен только к одному компьютеру. Переставлять USB-кабель в нужный компьютер каждый раз, когда это необходимо — занятие весьма бесполезное. Гораздо удобнее настроить совместный, т. е. общий доступ к этому принтеру, и просто отправлять документы на печать со своего компьютера на удаленный принтер.

ПРИМЕЧАНИЕ

Стоит отметить, что при совместном доступе к принтеру компьютер, к которому физически подключен принтер, должен быть включен. В противном случае печать будет отложена и возобновлена после включения данного компьютера.

В ОС Ubuntu Linux совместный доступ настраивается через знакомую нам службу CUPS.

Сама служба CUPS работает на 631 порту TCP-протокола и имеет Web-интерфейс для удобства администрирования. Для этого используется любой браузер, будь то Firefox либо Google Chrome. Запустите удобный для вас браузер и в адресной строке введите следующий адрес:

<http://localhost:631/>

Откроется Web-интерфейс службы CUPS. В открывшемся окне выберите раздел **Administration**. Вы будете перенаправлены в раздел администрирования установленных в системе принтеров (рис. 16.12).

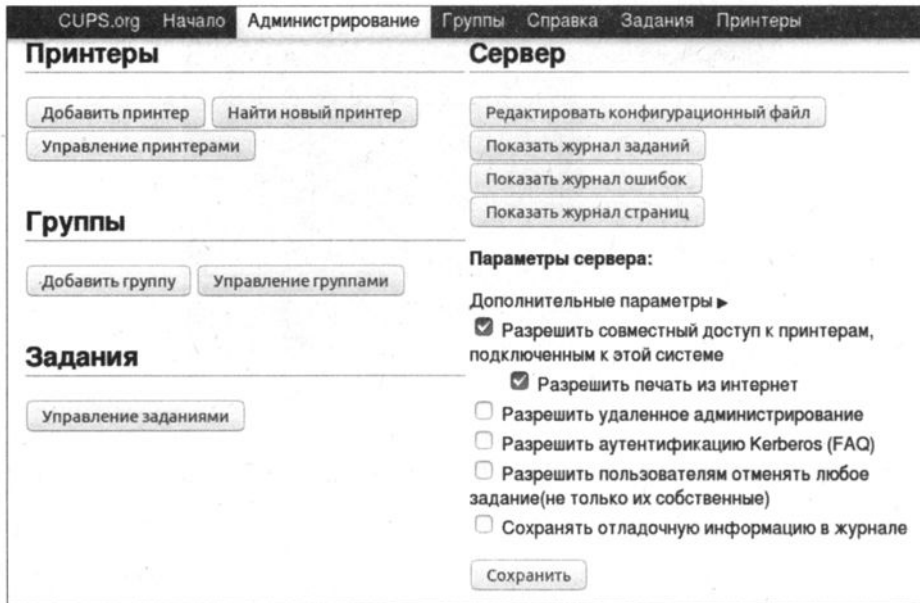


Рис. 16.12. Раздел администрирования службы GUPS

Для разрешения совместного доступа к установленному в системе принтеру нужно установить флажки **Разрешить совместный доступ к принтерам, подключенным к этой системе** и **Разрешить печать из интернет**. Для сохранения проделанных изменений нажмите кнопку **Сохранить**. Для подтверждения внесенных изменений служба GUPS потребует ввода вашего имени и пароля (рис. 16.13).

Введите имя пользователя и пароль и нажмите кнопку **ОК**. После этого служба GUPS будет перезапущена, и установленный принтер будет доступен по сети.

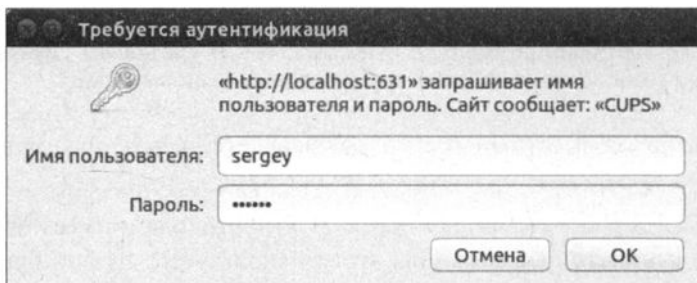


Рис. 16.13. Запрос аутентификации службы GUPS

Установка МФУ Epson Stylus CX5900 с помощью драйверов

Для установки МФУ Epson Stylus CX5900 в ОС Ubuntu можно воспользоваться драйверами от производителя. Для этого перейдите на сайт <http://global.epson.com> и в открывшейся странице выберите раздел **Products & Drivers**, а затем в раздел **Printing Solutions**. Щелкните по ссылке **Driver downloads** и в поле ввода введите модель устройства, в нашем случае достаточно ввести слово CX5900. Затем, для более точного поиска, выберите название ОС в раскрывающемся списке (рис. 16.14).

Найти поддержку для любых устройств

Введите название устройства

cx5900

е.г. XP-412

Операционная система

Linux

Рис. 16.14. Поиск драйверов для МФУ

Нажмите кнопку поиска, и вам будет предложено три позиции:

- ◆ **ESC/P-R Driver (generic driver)** — общий драйвер принтера;
- ◆ **Epson Printer Utility** — программное обеспечение для удобства работы с устройством;
- ◆ **All-in-one package** — драйвер сканера.

Для загрузки каждого модуля нажмите кнопку **Загрузить**. В ответ на это откроется страница загрузки нужного модуля. Нажмите кнопку **Принять**, расположенную внизу страницы, и чуть ниже самой кнопки раскроется список пакетов для загрузки (рис. 16.15).

Название файла	Размер файла	
epson-inkjet-printer-escpr-1.6.9-1lsb3.2.i486.rpm	3.03 MB	Загрузить
epson-inkjet-printer-escpr_1.6.9-1lsb3.2_i386.deb	3.49 MB	Загрузить
epson-inkjet-printer-escpr-1.6.9-1lsb3.2.x86_64.rpm	3.03 MB	Загрузить
epson-inkjet-printer-escpr_1.6.9-1lsb3.2_amd64.deb	3.48 MB	Загрузить
epson-inkjet-printer-escpr-1.6.9-1lsb3.2.src.rpm	3.23 MB	Загрузить
epson-inkjet-printer-escpr-1.6.9-1lsb3.2.tar.gz	3.46 MB	Загрузить

Рис. 16.15. Список файлов для загрузки

Выберите пакет, который называется `epson-inkjet-printer-escpr_1.6.9-1lsb3.2_amd64.deb`, и нажмите кнопку **Загрузить**.

ПРИМЕЧАНИЕ

Желательно выбирать именно тот файл, который предназначен для вашей операционной системы. Так, для ОС Ubuntu "родными" файлами установочных пакетов являются файлы, имеющие расширение `deb`. При этом желательно учитывать разрядность ОС. В нашем случае это 64-битная ОС Ubuntu.

Загрузите и сохраните файл на жесткий диск. То же самое проделайте и со вторым файлом Epson Printer Utility. Для удобства дальнейшего процесса установки сохраните данные файлы в домашнем каталоге вашей учетной записи.

При попытке загрузить драйвер сканера (**All-in-one package**) вместо списка пакетов будет предложено три варианта:

- ♦ **Package Download Page** — страница загрузки драйвера сканера;
- ♦ **Manual** — страница со справочной информацией;
- ♦ **Source File Download Page** — страница загрузки исходных файлов драйверов.

Выберите первый вариант **Package Download Page**, на открывшейся странице укажите нужную разрядность и скачайте драйвер, нажав кнопку **Download**. Будет загружен файл `iscan-bundle-1.0.3.x64.deb.tar.gz`.

Теперь перейдем к процессу установки загруженных драйверов.

Первым делом подключите USB-кабель от МФУ к компьютеру и включите устройство. Затем откройте Терминал и выполните следующую команду:

```
sudo dpkg -i /home/sergey/epson-inkjet-printer-escpr_1.6.9-1lsb3.2_amd64.deb
```

Данная команда позволит установить загруженный нами файл драйвера принтера. Теперь следует установить следующий пакет:

```
sudo dpkg -i /home/sergey/epson-printer-utility_1.0.0-1lsb3.2_amd64.deb
```

Выполненная команда позволит установить дополнительное программное обеспечение для удобства работы с принтером.

Теперь нам остается установить третий пакет — драйвер сканера, который у нас называется `iscan-bundle-1.0.3.x64.deb.tar.gz`. Обратите внимание, что сам драйвер сканера у нас заархивирован, и перед установкой его нужно распаковать:

```
tar xaf /home/sergey/iscan-bundle-1.0.3.x64.deb.tar.gz
```

Так как мы разархивировали установочный пакет, нам нужно перейти в каталог, который появился после распаковки архива:

```
cd iscan-bundle-1.0.3.x64.deb
```

Теперь нам остается только запустить процесс установки сканера:

```
./install.sh
```

Примерно через минуту процесс установки сканера будет завершен.

Все это замечательно, но если вы сейчас попытаетесь воспользоваться принтером, то заметите, что в разделе **Принтеры**, в **Параметрах системы**, наш только что установленный принтер не значится. Неужели мы что-то сделали неправильно? Дело в том, что мы просто установили драйверы для работы принтера, а нужно еще и вручную добавить сам принтер, т. е. МФУ. Сделать это можно следующим образом. Откройте **Параметры системы**, затем выберите пункт **Принтеры**, нажмите кнопку **Добавить**. Откроется уже знакомое нам окно добавления принтера (рис. 16.16).

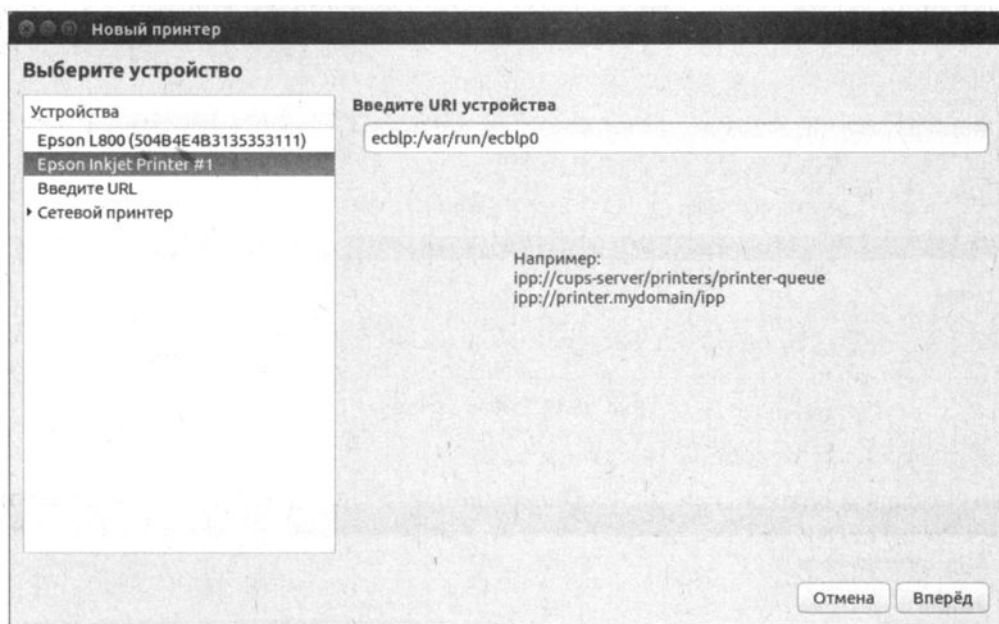


Рис. 16.16. Окно добавления принтера

В левой части окна добавления принтера выберите пункт **Epson Inkjet Printer #1** и нажмите кнопку **Вперед**.

На несколько секунд отобразится окно поиска драйверов, а затем откроется окно выбора драйвера (рис. 16.17).

Выберите **Epson** и нажмите кнопку **Вперед**. Теперь вам предстоит указать модель вашего устройства (рис. 16.18).

В левой части окна выберите пункт **Stylus CX5900**. Обратите внимание, что в правой части окна, в разделе **Драйверы**, отображается установленная нами версия драйвера: `epson-inkjet-printer-escpr 1.6.9-1lsb3.2`. Нажмите кнопку **Вперед**. Теперь нам нужно задать описание и имя принтера (рис. 16.19).

Вы вправе оставить все как есть и нажать кнопку **Применить**. После этого система предложит вам напечатать пробную страницу и отобразит свойства установленного принтера (рис. 16.20).

Нажмите кнопку **ОК**, и МФУ Epson Stylus CX5900 будет добавлено в систему. Теперь можете воспользоваться вашим принтером.

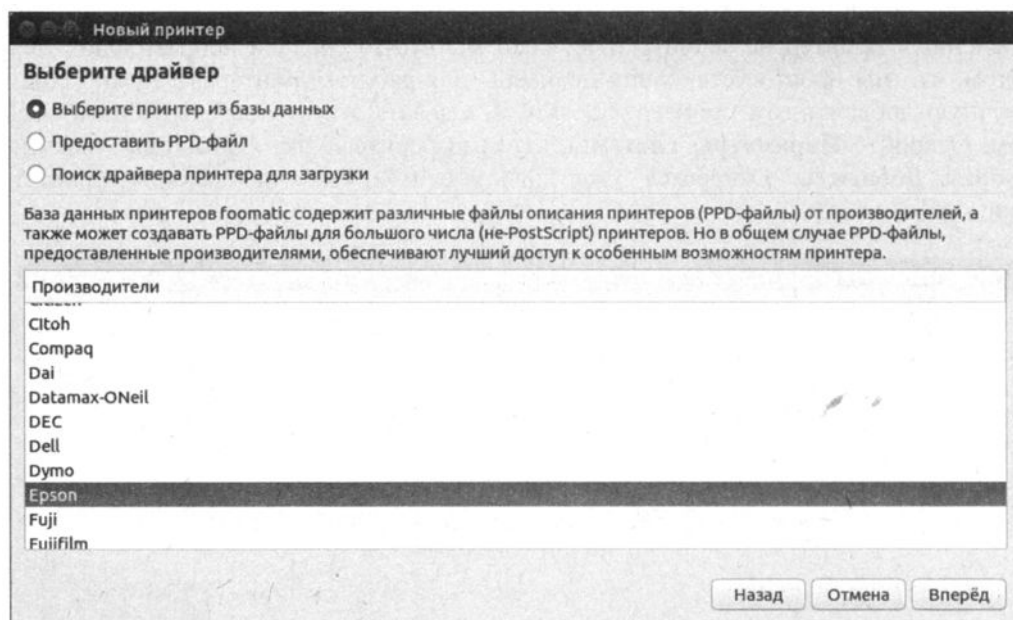


Рис. 16.17. Выбор драйвера

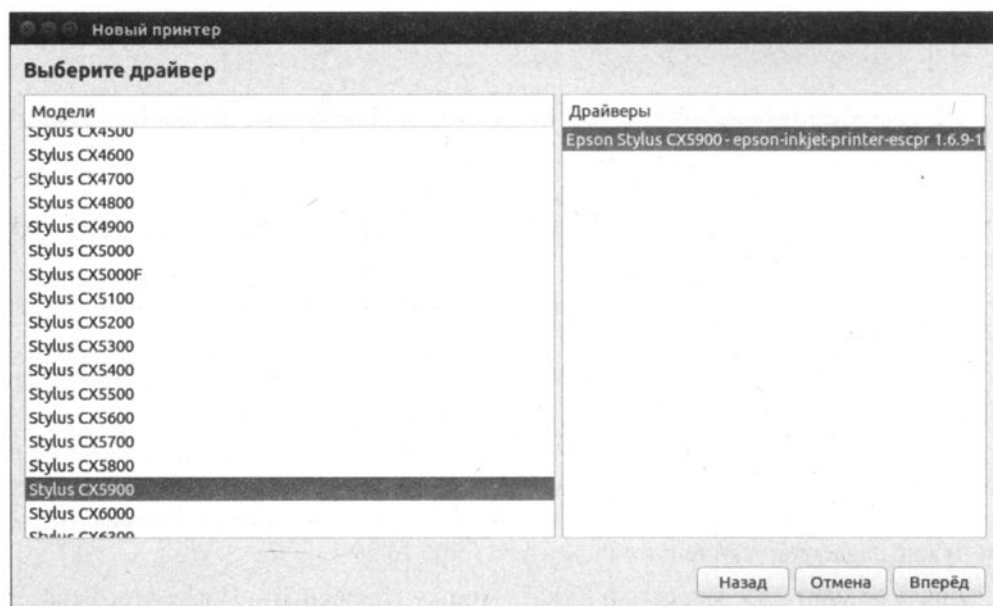


Рис. 16.18. Выбор модели принтера

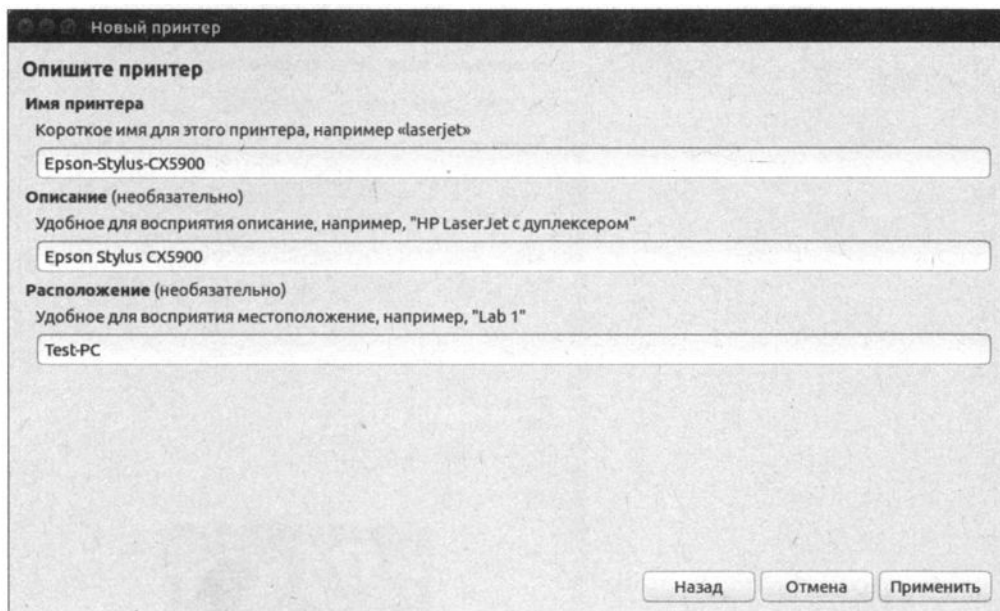


Рис. 16.19. Задание имени, расположения и описание принтера

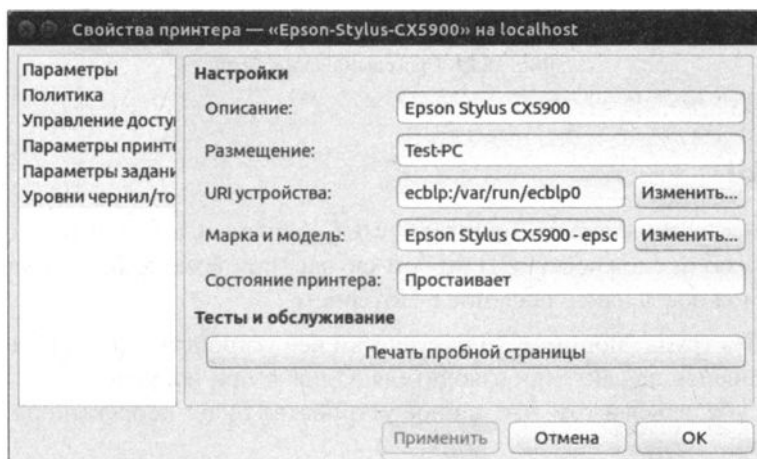


Рис. 16.20. Свойства установленного принтера

Для того чтобы воспользоваться сканером, выполните в Терминале следующую команду:

```
iscan
```

Запустится установленная нами программа сканирования Image Scan (рис. 16.21).

Запустить данную программу можно также из главного меню, выполнив поиск по слову *scan* или *epson*.

На этом процесс установки МФУ Epson Stylus CX5900 завершен.

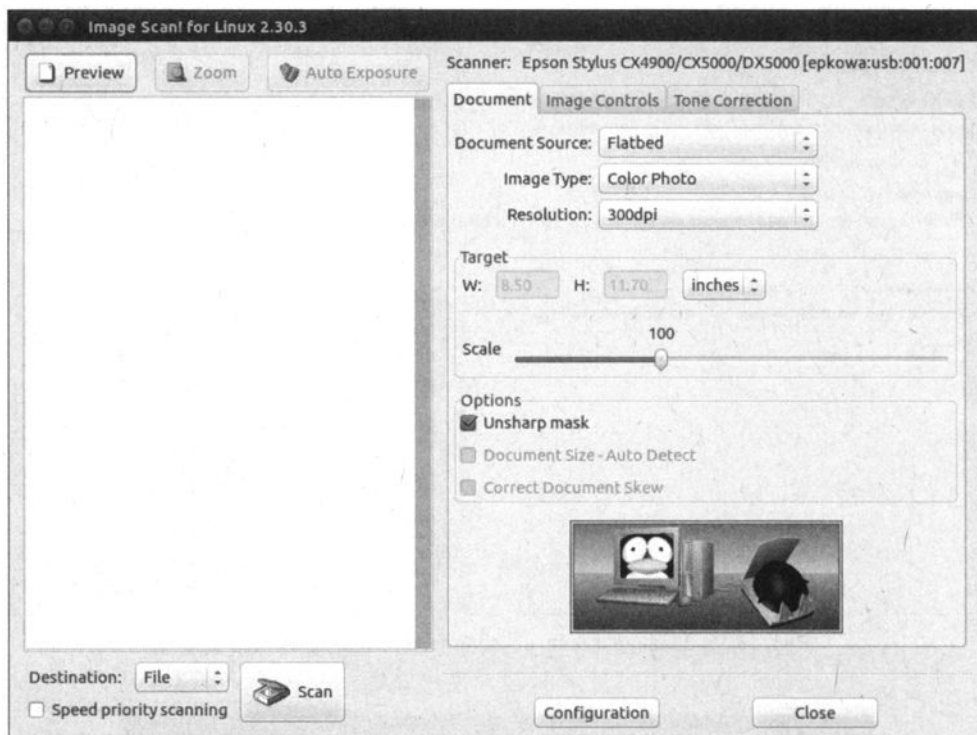


Рис. 16.21. Программа Image Scan

Выводы

Мы научились устанавливать принтеры с ОС Ubuntu. Сам процесс установки не отличается особой сложностью. В любом случае, при возникновении проблем всегда есть возможность найти решение в Интернете.

ОС Ubuntu без труда "понимает" практически все устройства, даже при отсутствии нужных драйверов на сайте производителя. Однако при покупке нового устройства необходимо убедиться в том, что данное устройство будет корректно работать в ОС Ubuntu Linux.

В ходе полученных в этой главе знаний и умений вы без труда сможете установить и настроить любой принтер в этой операционной системе.

ГЛАВА 17



Установка и удаление программ в ОС Ubuntu

Особенности установки и удаления программ

Теперь, когда мы уже столько знаем об ОС Ubuntu, пришло время не только поговорить об установке программ, но и самостоятельно приступить к выполнению этого процесса. Однако для начала поговорим о некоторых особенностях процесса установки программного обеспечения, которые присущи ОС Ubuntu.

В ОС Ubuntu Linux, в отличие от ОС Windows, процесс установки программного обеспечения сводится к автоматической установке по первому запросу пользователя. Для этого достаточно выполнить определенную команду в Терминале, и нужная программа будет установлена. В ОС Windows все происходит иначе: необходимо сначала скачать программу, а затем уже произвести ее установку. Помимо этого в ОС на базе Linux принято использование отдельных конфигурационных файлов для той или иной программы, которые хранятся в строго определенном каталоге. В ОС Windows подобная конфигурация хранится в реестре.

ПРИМЕЧАНИЕ

В последнее время в ОС Windows также наметилась тенденция использования отдельных конфигурационных файлов для хранения настроек программ (как было раньше в Windows 3.x), однако не все разработчики ПО используют эту возможность и "по привычке" хранят настройки программ в реестре.

Немаловажным отличием особенностей программного обеспечения ОС Ubuntu Linux от ОС Windows является то, что сами установочные файлы, предназначенные для ОС Ubuntu Linux, имеют сравнительно небольшой размер, в отличие от программ, предназначенных для ОС Windows. Связано это с тем, что установочные файлы ПО для ОС Windows содержат все необходимые наборы системных библиотек. В программном обеспечении для ОС Ubuntu Linux содержатся лишь файлы самого ПО, а системные библиотеки изначально уже хранятся в ОС Ubuntu Linux.

Таким образом получается, что каждая установленная программа в ОС Windows несет вместе с собой файлы системных библиотек, которые уже в большинстве случаев присутствуют в ОС. При удалении той или иной программы зачастую уда-

ляется лишь часть программы, а другая часть, например системные библиотеки, остается.

Не будем далее вдаваться в подробности, а рассмотрим варианты установки программного обеспечения в ОС Ubuntu Linux:

- ◆ установка приложений с помощью Менеджера приложений Ubuntu;
- ◆ установка приложений из deb-пакетов;
- ◆ установка приложений из Терминала;
- ◆ установка приложений из исходных текстов.

Чуть ниже мы более подробно рассмотрим каждый из вариантов установки программного обеспечения в ОС Ubuntu Linux, а пока поговорим про зависимости пакетов.

Зависимости пакетов в ОС Ubuntu Linux

В технической документации часто встречается упоминание о зависимостях пакетов, однако о том, что такое *зависимость пакетов*, упомянуть забывают. Давайте попробуем разобраться в этом.

Особенность ОС Ubuntu Linux в том, что она является модульной системой, т. е. операционная система в целом состоит из множества других программ, которые взаимодействуют между собой. Одна и та же системная библиотека в ОС Ubuntu Linux может использоваться несколькими программами одновременно. Таким образом, при использовании одной и той же системной библиотеки разработчику приложения не нужно включать в пакет эту же копию библиотеки. Все, что нужно, уже будет присутствовать в операционной системе. Соответственно, за вычетом этой системной библиотеки уменьшается размер самого устанавливаемого пакета.

Что случится, если необходимая системная библиотека не будет присутствовать в ОС? При установке такого пакета средствами менеджера пакетов, который умеет проверять зависимости, не случится ничего плохого. Просто при установке приложения будет выведена соответствующая ошибка. Однако если устанавливать приложение пакетным менеджером, который не проверяет зависимости, то приложение установится, но его запуск станет невозможным. По этой причине разработчик программы должен самостоятельно позаботиться о зависимостях, которые могут возникнуть при установке его приложения. Каким образом он может это реализовать? Дело в том, что файл программного пакета в большинстве случаев состоит из двух частей: *архива с бинарными файлами программ*, которые будут установлены, и *спецификации самого программного пакета*, т. е. файла, в котором перечислены зависимости и предоставляемые этим программным пакетом системные библиотеки.

Таким образом, при установке подобного пакета пакетный менеджер в первую очередь проверяет по своей базе установленных пакетов и их зависимостей, не нарушатся ли существующие зависимости при установке того или иного программного пакета. Если при установке, удалении или обновлении программного пакета нару-

шаются существующие зависимости, например, вместе с удаляемым приложением будет удалена определенная системная библиотека, которая также необходима для работы других приложений, то менеджер пакетов выдаст сообщение об ошибке и остановит свою работу.

Такая система зависимостей оказалась довольно сложной и не всегда приводила к нужному результату. Для решения этой проблемы была введена концепция *репозитория* — своеобразной базы данных, которая содержит список программных пакетов для конкретного дистрибутива. Более подробно о репозитории мы поговорим чуть позже, а теперь перейдем к установке приложений.

Установка приложений с помощью Менеджера приложений Ubuntu

На современных мобильных устройствах вам наверняка не раз приходилось пользоваться такими приложениями, как Google Play Market или Apple Store. В ОС Ubuntu есть подобная реализация, которая называется *Менеджером приложений Ubuntu*.

Запустить Менеджер приложений Ubuntu можно из панели запуска, щелкнув на соответствующем значке (рис. 17.1).



Рис. 17.1. Менеджер приложений Ubuntu на панели запуска

Откроется окно Менеджера приложений Ubuntu (рис. 17.2).

Менеджер приложений Ubuntu чем-то схож по своему функционалу с Google Play Market и Apple Store и для своей работы требует подключения к Интернету, т. к. программы, загружаемые в Менеджере приложений Ubuntu, загружаются именно из сети.

В верхней части окна имеется три вкладки:

- ♦ **Все** — вкладка открывается по умолчанию и содержит приложения, разделенные по категориям;
- ♦ **Установлено** — список установленного программного обеспечения. На этой же вкладке имеется возможность удалить ненужное приложение;
- ♦ **Обновления** — на вкладке отображаются доступные обновления. Обратите внимание на цифру 1 рядом с названием вкладки — это уведомление о том, что для вашей системы доступно обновление, которое необходимо установить, нажав соответствующую кнопку на вкладке.

Менеджер приложений Ubuntu имеет довольно простой и понятный интерфейс, и вам не составит труда найти нужное приложение.

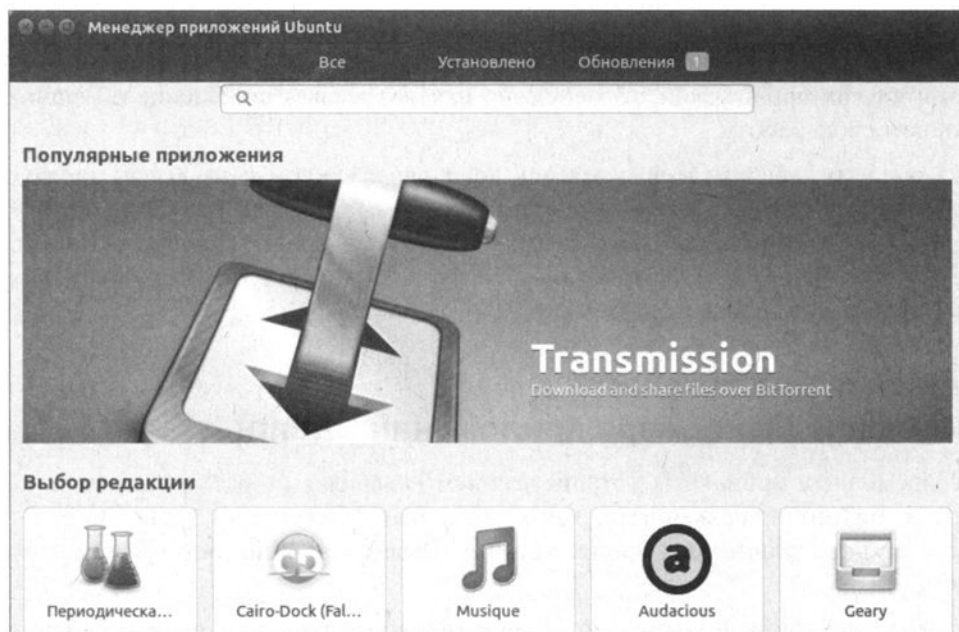


Рис. 17.2. Окно Менеджера приложений Ubuntu

ПРИМЕЧАНИЕ

Стоит помнить, что не все приложения имеются в Менеджере приложений Ubuntu, например, браузер Google Chrome доступен для загрузки на соответствующем сайте.

Давайте в качестве примера установим torrent-клиент qBittorrent. Для этого в строке поиска Менеджера приложений Ubuntu введите слово qBittorrent. Результат поиска отобразится в этом же окне (рис. 17.3).



Рис. 17.3. Результат поиска программы qBittorrent

Теперь, если щелкнуть на результате поиска, например на кратком описании программы, откроется страница с подробными сведениями о найденной программе (рис. 17.4).

Для установки программы qBittorrent нажмите кнопку **Установить**. Для подтверждения установки появится окно аутентификации пользователя (рис. 17.5).

Введите свой пароль и нажмите кнопку **Аутентифицировать**. Начнется установка программы qBittorrent, о ходе которой в панели запуска будет свидетельствовать

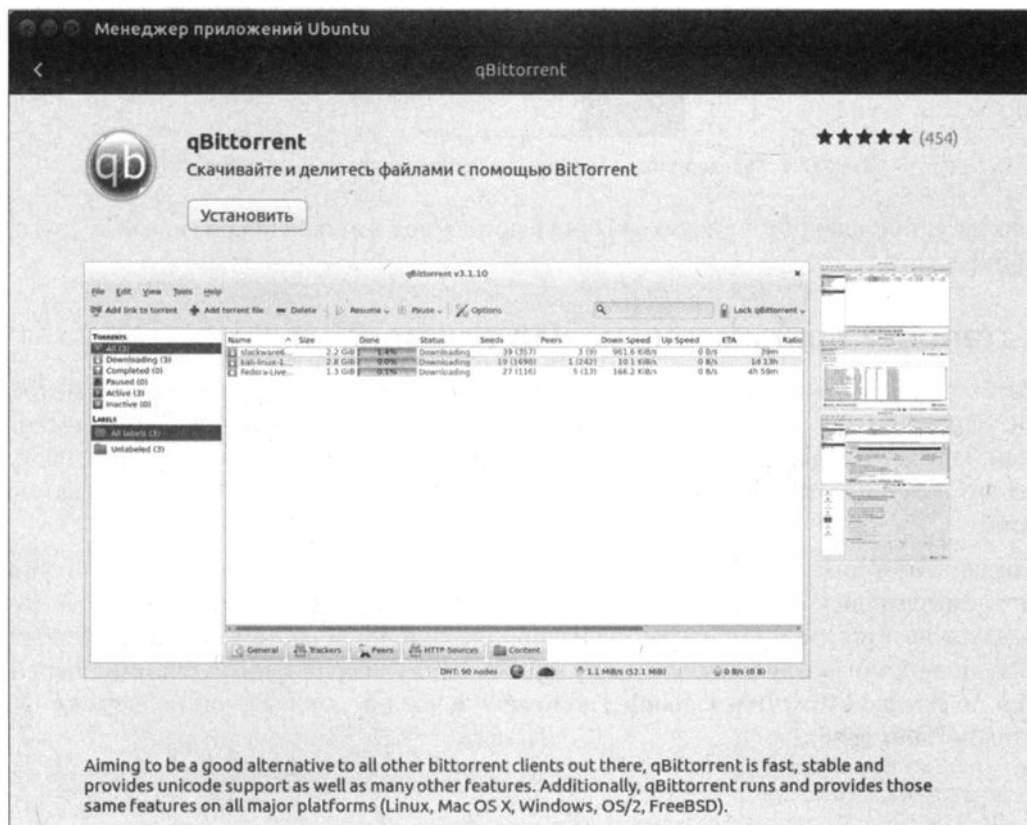


Рис. 17.4. Страница с подробным описанием программы qBittorrent

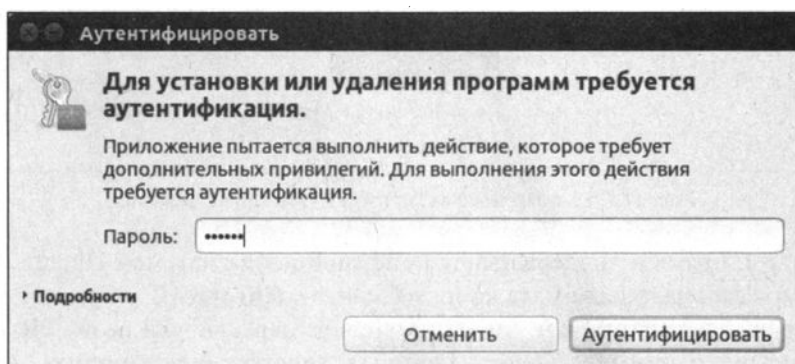


Рис. 17.5. Окно аутентификации пользователя

соответствующий информационный значок приложения с индикатором хода выполнения установки (рис. 17.6).

По окончании процесса установки программы qBittorrent индикатор процесса установки сменится привычным значком приложения qBittorrent.



Рис. 17.6. Индикатор хода выполнения установки программы qBittorrent

Запуск установленного приложения осуществляется тем же способом, что и другие приложения.

Установка приложений из deb-пакетов

Установку приложений с помощью Менеджера приложений Ubuntu мы рассмотрели, однако не все программы включены в этот список приложений. Так, популярный браузер Google Chrome отсутствует в списке Менеджера приложений Ubuntu, но это вовсе не означает, что теперь мы не сможем установить этот обозреватель сети.

Когда необходимое приложение отсутствует в Менеджере приложений Ubuntu, есть смысл поискать его на официальном сайте разработчика и убедиться в том, поддерживается ли это приложение нашей операционной системой. Давайте проверим, поддерживает ли браузер Google Chrome нашу операционную систему: перейдем по ссылке <https://www.google.ru/chrome/> и убедимся в наличии поддержки ОС Ubuntu Linux (рис. 17.7).

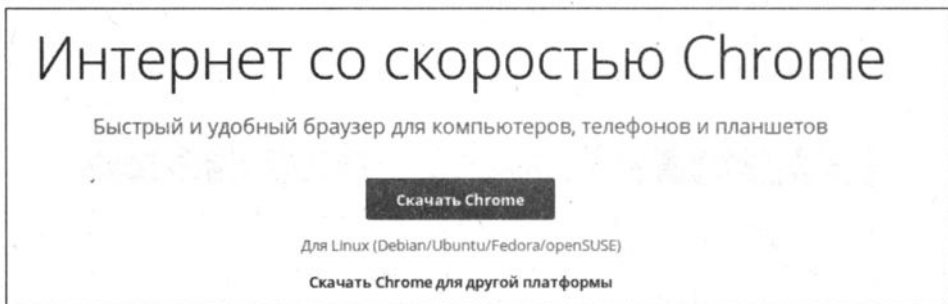


Рис. 17.7. Браузер Google Chrome доступен для ОС Ubuntu

Как мы видим, браузер поддерживается операционной системой Ubuntu Linux, поэтому мы можем смело нажимать кнопку **Скачать Chrome**. В открывшемся диалоговом окне выберите нужный пакет, установив переключатель **64 bit .deb (для Debian/Ubuntu)**, и нажмите кнопку **Принять условия и установить**. Сохраните загружаемый файл `google-chrome-stable_current_amd64.deb` в любом удобном вам каталоге.

Перейдем к процессу установки браузера Google Chrome. Установка программ из deb-пакетов может выполняться двумя различными способами:

- ◆ двойным щелчком мыши по загруженному deb-пакету;
- ◆ с помощью командной строки.

Первый метод установки отличается своей простотой и удобством и даже при отсутствии навыков вам не составит труда установить требуемый deb-пакет с помощью командной строки. Рассмотрим первый способ установки.

Откройте каталог, в который вы загрузили наш установочный пакет `google-chrome-stable_current_amd64.deb`, и дважды щелкните по нему (по пакету) левой кнопкой мыши. Откроется уже знакомое нам окно Менеджера приложений Ubuntu (рис. 17.8).

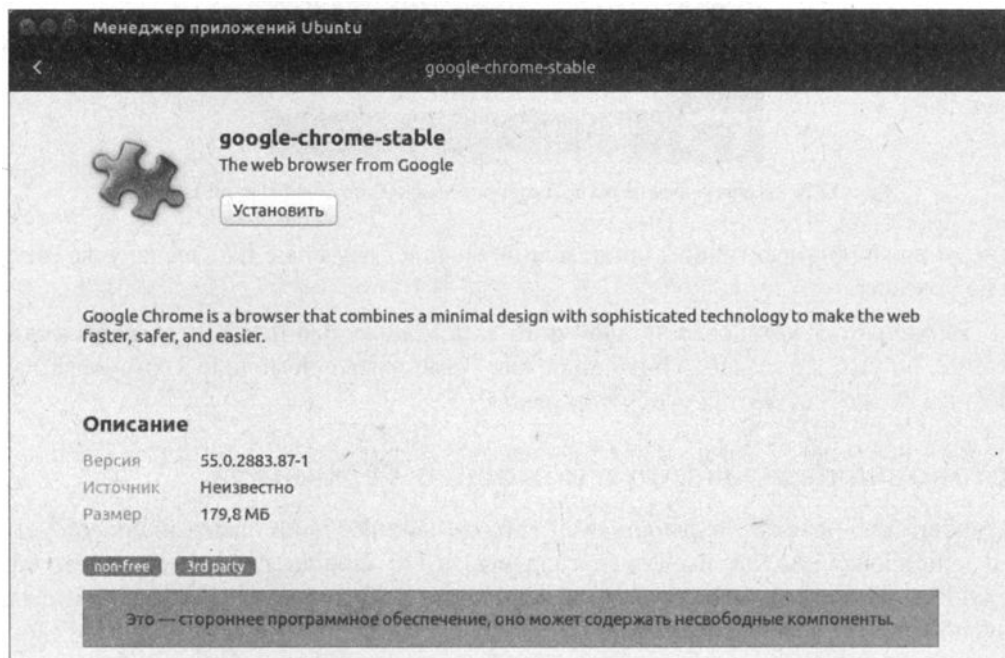


Рис. 17.8. Установка Google Chrome с помощью Менеджера приложений Ubuntu

ПРИМЕЧАНИЕ

В описании к программе появилось предупреждение о том, что она может содержать несвободные компоненты. Именно по этой причине браузер Google Chrome отсутствует в списке приложений Менеджера приложений Ubuntu.

Для установки Google Chrome нажмите кнопку **Установить**, а затем подтвердите установку, введя свой пароль и нажав кнопку **Аутентифицировать**.

Так как установка производилась с применением установочного deb-пакета, то в панели запуска не будет соответствующего значка приложения. Поэтому откройте главное меню, нажав соответствующую кнопку на панели запуска либо нажав кнопку <Super> (знакомая нам кнопка <Win>) и введя в строке поиска слово `chrome`. В результатах поиска отобразится установленное приложение Google Chrome. Запустите его.

Как только приложение запустится, на панели запуска появится его значок. Однако есть одна особенность: как только вы закроете браузер Google Chrome, значок при-

ложения исчезнет с панели запуска. Какими будут наши следующие действия? Правильно, нужно закрепить значок приложения на панели запуска. Для этого щелкните правой кнопкой мыши по значку работающего приложения и выберите пункт меню **Закрепить на панели** (рис. 17.9).

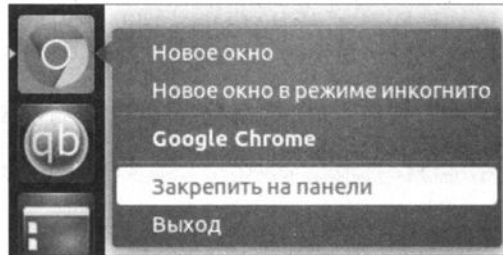


Рис. 17.9. Закрепление значка приложения Google Chrome на панели запуска

Теперь, даже при неактивном приложении, значок браузера с панели запуска никуда не исчезнет.

Так выполняется установка приложений с помощью deb-пакетов в графическом режиме, но этот же самый deb-пакет можно установить с помощью Терминала. Рассмотрим подробнее этот способ установки.

Установка deb-пакетов с помощью Терминала

Установка deb-пакетов "в два клика" — это самый простой и быстрый способ установки приложений. Как вы уже догадались, в ОС Ubuntu Linux существует возможность установки того же самого deb-пакета в Терминале. Делается это с помощью пакетного менеджера `dpkg`.

`Dpkg` (сокр. от *Debian package*) — пакетный менеджер для Debian-подобных операционных систем, который позволяет не только устанавливать и удалять пакеты, но и создавать их. Однако `dpkg` "не умеет" загружать deb-пакеты и устанавливать их. Процесс установки сводится к загрузке нужного пакета, а уже затем можно приступить к его установке.

Процесс установки сводится к следующему:

1. Из deb-пакета извлекаются файлы управления пакетом.
2. Производится проверка: имеется ли в системе более ранняя версия устанавливаемой программы, и если данное условие верное, то запускается сценарий `prerm`, который удаляет устаревшие компоненты (модули) устанавливаемой программы.
3. Если в устанавливаемом пакете присутствует сценарий `preinst`, то происходит его запуск.
4. Происходит распаковка файлов устанавливаемого пакета и резервное копирование старых компонентов программы, если была установлена предыдущая версия.

5. Если была обнаружена предыдущая версия устанавливаемого пакета, то запускается сценарий `postrm`, который обновляет конфигурационные настройки в соответствии с требованиями нового пакета.
6. Запускается сценарий `postinst`, который запускает настройку установленного пакета.

ПРИМЕЧАНИЕ

Пакетный менеджер `dpkg` "не умеет" устанавливать зависимости, поэтому прибегайте к его помощи лишь при установке тех программных пакетов, в которых вы уверены.

Синтаксис команды `dpkg` следующий:

`dpkg [ключ] <команда>`

Вкратце рассмотрим основные значения, которые принимает параметр `[ключ]` (табл. 17.1).

Таблица 17.1. Параметры команды `dpkg`

Сокращенный вариант	Полный вариант	Описание
<code>-i</code>	<code>--install</code>	Позволяет установить пакет, переданный в параметре <code><команда></code>
	<code>--instdir=<каталог></code>	Позволяет изменить корневой каталог для установки приложения, при этом не меняя административный каталог
	<code>--unpack</code>	Позволяет распаковать пакет без его установки и последующей настройки
	<code>--admindir=<каталог></code>	Позволяет использовать каталог, указанный в параметре <code><каталог></code> , вместо каталога <code>/var/lib/dpkg</code>
<code>-A</code>	<code>--record-avail</code>	Позволяет обновить запись о доступных пакетах информацией, которая указана в устанавливаемом пакете
	<code>--configure</code>	Позволяет переконфигурировать распакованный пакет
	<code>--log=<имя_файла></code>	Позволяет записать изменения состояний и действия в файл, переданный в параметре <code><имя_файла></code>
<code>-l</code>	<code>--list</code>	Позволяет вывести список установленных в системе пакетов с указанием их версии, архитектуры и краткого описания
<code>-L</code>	<code>--listfiles <имя_пакета></code>	Позволяет вывести список используемых пакетом файлов, имя которого передано в параметре <code><имя_пакета></code>
	<code>--triggers-only</code>	Позволяет запустить только послеустановочный триггер

Таблица 17.1 (окончание)

Сокращенный вариант	Полный вариант	Описание
-r	--remove	Позволяет удалить пакет, указанный в параметре <i><команда></i>
	--root= <i><каталог></i>	Позволяет произвести установку приложения в другой корневой каталог, указанный в параметре <i><каталог></i>
-p	--print-avail	Параметр позволяет получить информацию о доступной версии пакета, переданного в параметре <i><команда></i>
-P	--purge	Позволяет удалить пакет вместе с его конфигурационными файлами
-s	--status	Позволяет получить статус указанного пакета, переданного в параметре <i><команда></i>
-V	--verify	Позволяет проверить контрольную сумму установленных пакетов, а при указании конкретного пакета — его контрольную сумму

ПРИМЕЧАНИЕ

В табл. 17.1 приведены только наиболее используемые параметры команды `dpkg`. Полный список параметров можно узнать, выполнив команду `dpkg --help`.

Теперь, когда мы вкратце познакомились с основными параметрами команды `dpkg`, настало время произвести установку `deb`-пакета с помощью командной строки. Для установки загруженного нами пакета `google-chrome-stable_current_amd64.deb` выполните следующую команду:

```
sudo dpkg -i google-chrome-stable_current_amd64.deb
```

В большинстве случаев установка будет произведена без каких-либо дополнительных вопросов.

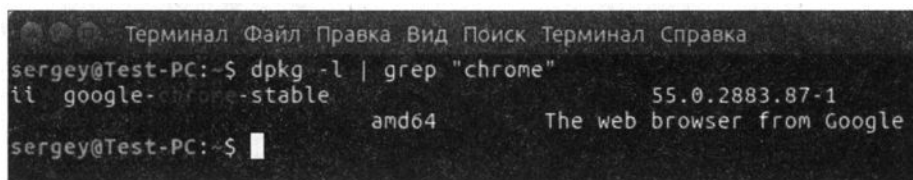
Для того чтобы проверить список установленных пакетов, выполните следующую команду:

```
dpkg -l
```

Однако вывод данной команды будет слишком длинный, и если нужно будет найти какой-то определенный пакет, то вам придется внимательно вчитываться в каждую строчку. Для того чтобы облегчить задачу поиска нужного пакета, следует задать условие поиска так:

```
dpkg -l | grep "chrome"
```

В результате будут отображены только те пакеты, в чьих именах есть слово `chrome` (рис. 17.10).



```
Терминал Файл Правка Вид Поиск Терминал Справка
sergey@Test-PC:~$ dpkg -l | grep "chrome"
ii  google-chrome-stable          55.0.2883.87-1
                                amd64            The web browser from Google
sergey@Test-PC:~$
```

Рис. 17.10. Найден установленный пакет Google Chrome

Обратите внимание, что пакет имеет название не просто `chrome`, а `google-chrome-stable`. Поэтому, если выполнить следующую команду:

```
dpkg -L chrome
```

то в ответ будет выведено сообщение о том, что данный пакет не установлен. Это происходит потому, что нужно вводить полное имя пакета. Исправим нашу команду:

```
dpkg -L google-chrome-stable
```

Теперь наша команда сработает так, как надо, и выведет список тех файлов, которые использует пакет `google-chrome-stable`.

Установка приложений из Терминала

Помимо рассмотренных выше способов установки приложений, существует еще способ установки приложений напрямую из Терминала. При этом от вас совершенно не требуется загрузка этих приложений на компьютер. Вам необходимо лишь только дать соответствующую команду Терминалу и нужная программа будет установлена.

В ОС Ubuntu для установки программ из Терминала давным-давно существует программа `apt`, которая является обычной надстройкой над уже знакомым нам пакетным менеджером `dpkg`.

`Apt` (Advanced Packaging Tool) — набор программного обеспечения, предназначенного для установки, удаления и обновления программных пакетов в операционных системах семейства Debian.

Программный пакет `apt` позволяет произвести установку приложений из репозитория либо выполнить установку с имеющегося носителя. `Apt` "умеет" решать зависимости пакетов, чего лишен пакетный менеджер `dpkg`.

Мы уже познакомились с командой `apt`, точнее, с командой `apt-get`, когда занимались установкой драйверов. Теперь познакомимся с этой командой подробнее.

У вас сразу возникнет вопрос: почему мы использовали команду `apt-get`, а не `apt`? И в чем их различие, если оно вообще имеется?

На заре становления ОС Ubuntu было принято использование оболочки `apt-get` для управления пакетами. Постепенно, когда ОС Ubuntu становилась все сложнее и функциональнее, оболочка `apt-get` перестала отвечать требованиям современной операционной системы, и был введен новый инструмент управления пакетами `apt`.

Таким образом, на смену длинной команде `apt-get` пришла более короткая команда `apt`, которая в то же время имеет более широкий функционал по сравнению с ее предшественником `apt-get`. Команда `apt` лишена всех тех недостатков, которые были присущи ее предшественнику `apt-get`, т. к. функционал новой команды разрабатывался уже с учетом потребностей современной операционной системы. При этом основной упор делался на простоту и удобство использования. Это далеко не весь список улучшений, которые были учтены при разработке `apt`.

Теперь все же ответим на вопрос, почему мы при установке драйверов использовали команду `apt-get`, а не `apt`? Дело в том, что команда `apt-get` еще некоторое время будет присутствовать в составе ОС Ubuntu, и пройдет какое-то время, прежде чем ее полностью исключат из состава операционной системы. Однако в технической литературе вы будете довольно часто встречать упоминание этой команды, и поэтому для общего представления нужно знать о существовании обеих команд. Но имейте в виду, что вместо устаревшей команды `apt-get` рекомендуется использовать современную `apt`.

ПРИМЕЧАНИЕ

В технической литературе довольно часто можно встретить упоминание команды `aptitude`, которая также является надстройкой над Advanced Packaging Tool, но предназначена именно для ОС Debian. В ОС Ubuntu эту команду можно использовать только после предварительной ее установки. Однако все же рекомендуется пользоваться именно командой `apt` вместо команд `apt-get` и `aptitude`.

В общем виде, синтаксис команды `apt` следующий:

`apt [ключ] <пакет>`

В качестве параметра `<пакет>` команде `apt` передается имя устанавливаемого пакета. Параметр `[ключ]` может принимать значения, перечисленные в табл. 17.2.

Таблица 17.2. Параметры команды `apt`

Ключ	Описание
<code>list</code>	Позволяет вывести список пакетов из указанных имен в параметре <code><пакет></code>
<code>search</code>	Позволяет найти информацию о пакете в описаниях пакетов
<code>show</code>	Позволяет отобразить дополнительные данные о пакете
<code>install</code>	Позволяет установить указанный пакет в параметре <code><пакет></code>
<code>remove</code>	Позволяет удалить пакет, переданный в параметре <code><пакет></code>
<code>purge</code>	Позволяет удалить не только указанный пакет, но и все его конфигурационные файлы
<code>autoremove</code>	Позволяет автоматически удалить все неиспользуемые пакеты
<code>update</code>	Позволяет обновить список доступных пакетов
<code>upgrade</code>	Позволяет обновить операционную систему, при этом обновляются и содержащиеся в системе пакеты, но в случае возникновения конфликтов из-за зависимостей пакетов такие пакеты будут пропущены

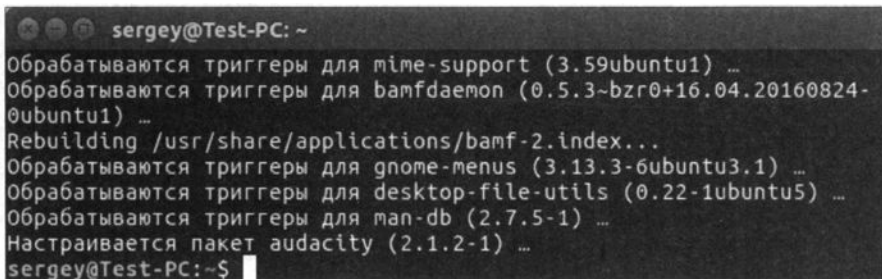
Таблица 17.2 (окончание)

Ключ	Описание
full-upgrade	Позволяет полностью обновить систему, при этом в случае возникновения конфликтов с зависимостью пакетов эти пакеты будут удалены и установлены новые версии пакетов
edit-sources	Позволяет редактировать файл с источниками пакетов

Итак, давайте на практике испробуем команду `apt` и установим аудиоредактор Audacity:

```
sudo apt install audacity
```

После ввода вашего пароля начнется установка приложения. О ее окончании свидетельствует приветственная строка Терминала, готовая к вводу новых команд (рис. 17.11).



```
sergey@Test-PC: ~
Обрабатываются триггеры для mime-support (3.59ubuntu1) ...
Обрабатываются триггеры для bamfdaemon (0.5.3-bzr0+16.04.20160824-
0ubuntu1) ...
Rebuilding /usr/share/applications/bamf-2.index...
Обрабатываются триггеры для gnome-menus (3.13.3-6ubuntu3.1) ...
Обрабатываются триггеры для desktop-file-utils (0.22-1ubuntu5) ...
Обрабатываются триггеры для man-db (2.7.5-1) ...
Настраивается пакет audacity (2.1.2-1) ...
sergey@Test-PC:~$
```

Рис. 17.11. Установка программы Audacity завершена

Обратите внимание, что Терминал не оповещает нас об окончании установки. Об успешности самой процедуры установки мы можем догадаться только по приветственной строке Терминала. При этом на панели запуска также отсутствует значок приложения, поэтому запуск приложения осуществляется либо по поиску названия программы в главном меню, либо из Терминала.

Удаление программ производится аналогичным образом:

```
sudo apt remove audacity
```

Помимо запроса пароля, от вас потребуется подтверждение удаления программы. Для подтверждения удаления следует нажать клавишу <у> либо <д> в Терминале.

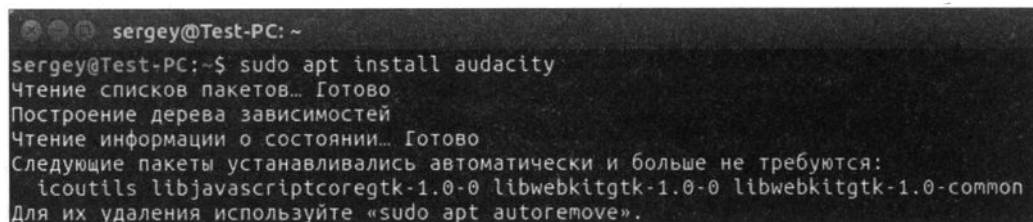
После удаления программы часто остаются неудаленными файлы настроек программ, правильнее будет выразиться — конфигурационные файлы. С конфигурационными файлами мы уже знакомы. Да, это те самые файлы, что обычно хранятся в каталоге `/etc/` и в домашнем каталоге пользователя. Вручную удалять эти файлы не стоит. Поэтому если вы хотите, чтобы вместе с программой были удалены и ее конфигурационные файлы, то воспользуйтесь следующей командой:

```
sudo apt purge audacity
```

Данный способ является наиболее правильным, т. к. программа `apt` сама удалит все, что ненужно, и при этом сделает это как можно корректно. Если удалить файлы вручную, то велика вероятность, что вы по ошибке удалите нужные системе файлы, и дальнейшая установка этого приложения будет заканчиваться ошибкой. Пользуйтесь теми средствами, что предлагают нам разработчики операционной системы.

Сейчас вам может показаться странным, откуда Терминал "берет" нужные программы для установки, ведь мы их не загружаем предварительно, а только указываем, какую программу установить. Все довольно просто: программа `apt` считывает файл `sources.list`, который находится в каталоге `/etc/apt/`, и на основании этого списка загружает последнюю версию требуемой программы. Удобно, не правда ли?

Когда устанавливается какая-либо программа, пакетный менеджер `apt` перед каждой установкой и удалением программ считывает информацию о зависимостях пакетов и в случае обнаружения конфликтов или неиспользуемых пакетов выдает соответствующее уведомление (рис. 17.12).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ sudo apt install audacity  
Чтение списков пакетов... Готово  
Построение дерева зависимостей  
Чтение информации о состоянии... Готово  
Следующие пакеты устанавливались автоматически и больше не требуются:  
  icoutils libjavascriptcoregtk-1.0-0 libwebkitgtk-1.0-0 libwebkitgtk-1.0-common  
Для их удаления используйте «sudo apt autoremove».
```

Рис. 17.12. Программа `apt` нашла неиспользуемые пакеты

Как видите, в нашей системе присутствует 4 неиспользуемых пакета. Нам они ни к чему, поэтому давайте их удалим:

```
sudo apt autoremove
```

После подтверждения удаления этих пакетов программа `apt` поочередно удалит каждый из этих четырех пакетов и избавит вас от их ручного удаления.

Установка приложений из исходных текстов

К установке программ из исходных текстов следует прибегать лишь только в крайних случаях, например, когда нужна определенная программа, но она распространяется только в исходных текстах и не поставляется в `deb`-пакетах. С установкой программ из исходных текстов мы уже сталкивались, когда устанавливали драйверы оборудования. Сейчас мы более подробно познакомимся с самим процессом установки программ из исходных текстов.

Давайте установим простейший видеоредактор Avidemux, который поставляется в исходных кодах. Для начала нам предстоит скачать архив с исходными текстами. Для этого перейдите на страницу проекта <http://fixounet.free.fr/avidemux/> и щелкните по ссылке **Downloads**. На открывшейся странице найдите раздел **Linux, source** и нажмите ссылку **FossHub** (рис. 17.13).

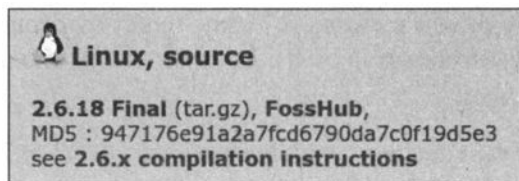


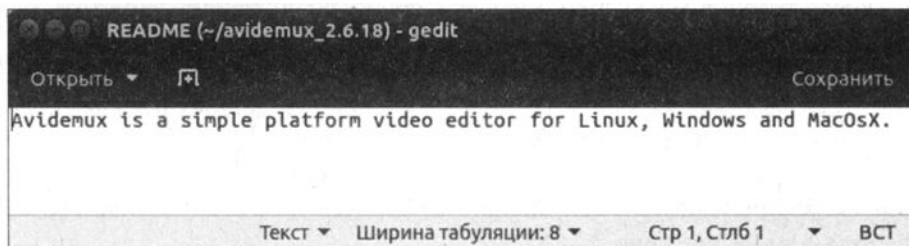
Рис. 17.13. Раздел с доступной версией программы Avidemux

Сохраните загружаемый файл `avidemux_2.6.18.tar.gz` в домашнем каталоге для дальнейшего удобства установки программы. Как мы видим, исходные тексты программы Avidemux упакованы в архив, который имеет расширение `tar.gz`, и первым делом нам нужно его распаковать:

```
tar -xvf avidemux_2.6.18.tar.gz
```

В результате выполнения команды в вашем домашнем каталоге будет создан каталог `~/avidemux_2.6.18/`, в котором появится все содержимое нашего архива.

Перед тем как устанавливать любое приложение из исходных текстов, рекомендуется заглядывать в файл `ReadMe`, который обычно находится в корне архива. Давайте и мы заглянем в этот файл, возможно, в нем есть инструкции по установке программы (рис. 17.14).

Рис. 17.14. Содержимое файла `ReadMe`

Что ж, как видим, в этом файле нет ничего полезного для нас. Значит, будем искать дальше. Попробуем поискать в корне архива файл `Install`, возможно, в нем есть какая-то информация по установке. Не нашли этот файл? Давайте вспомним, что мы упустили из виду, когда загружали архив с исходными кодами приложения Avidemux.

Как же мы могли забыть, ведь в разделе **Linux, source** под ссылкой **FossHub** была еще одна ссылка — **2.6.x compilation instructions** (см. рис. 17.13).

На открывшейся странице с инструкциями по установке приложения мы видим, что разработчики приложения предусмотрели только возможность создания готового пакета для нашей операционной системы. Следовательно, привычные нам команды `./configure`, `make` и `sudo make install` уже не сработают. Поэтому будем следовать инструкциям разработчиков.

Перед любыми манипуляциями с содержимым каталога нужно сделать его текущим:

```
cd avidemux_2.6.18
```

Теперь, когда мы находимся в каталоге `/home/sergey/avidemux_2.6.18/`, от нас потребуется запустить файл `bootStrap.bash` с дополнительными параметрами:

```
bash bootStrap.bash --deb
```

Однако при попытке выполнения данной команды ничего не произойдет, т. к. мы не учли требования, которых нужно придерживаться при установке приложения Avidemux. Для этого на странице с инструкциями находим раздел **Requirement list (debian/ubuntu)** и читаем требования. Там указано, что для сборки пакета программы Avidemux необходимо установить набор модулей, выполнив следующую команду:

```
sudo apt-get install gcc g++ make cmake pkg-config libpng12-dev fakeroot yasm  
libsqlite3-dev build-essential
```

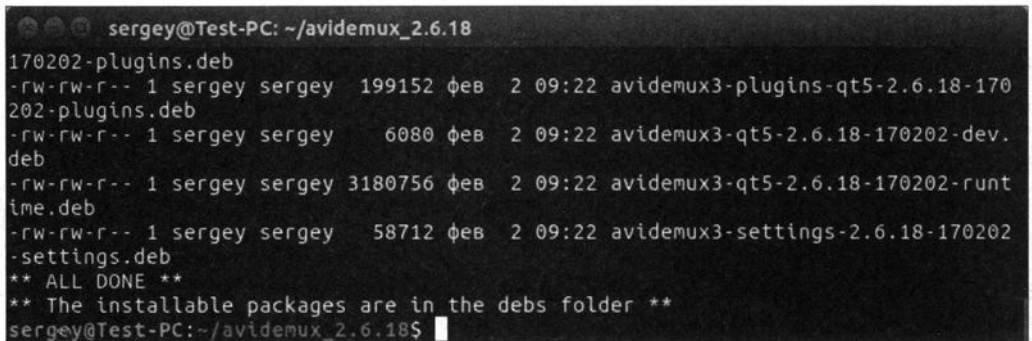
Установка данных модулей не займет много времени. От вас только потребуется подтвердить установку и дождаться ее окончания. Если сейчас снова попытаться запустить предыдущую команду, то у нас опять появится ошибка об отсутствии установленного модуля QT5, поэтому, как и рекомендуют разработчики, для дальнейшей установки необходим соответствующий модуль. Установим его:

```
sudo apt-get install qttools5-dev-tools qtbase5-dev
```

Теперь, когда установлен необходимый набор модулей, можно запустить процесс сборки, точнее, компиляции программы Avidemux:

```
bash bootStrap.bash --deb
```

Этой командой мы запустим процесс сборки. Сам процесс сборки занимает немного времени, но не так быстро, как установка необходимых компонентов для компиляции этого приложения. Об успешном процессе компиляции свидетельствует строка `*** ALL DONE ***` (рис. 17.15).



```
sergey@Test-PC: ~/avidemux_2.6.18
170202-plugins.deb
-rw-rw-r-- 1 sergey sergey 199152 фев 2 09:22 avidemux3-plugins-qt5-2.6.18-170
202-plugins.deb
-rw-rw-r-- 1 sergey sergey 6080 фев 2 09:22 avidemux3-qt5-2.6.18-170202-dev.
deb
-rw-rw-r-- 1 sergey sergey 3180756 фев 2 09:22 avidemux3-qt5-2.6.18-170202-runt
ime.deb
-rw-rw-r-- 1 sergey sergey 58712 фев 2 09:22 avidemux3-settings-2.6.18-170202
-settings.deb
** ALL DONE **
** The installable packages are in the debs folder **
sergey@Test-PC:~/avidemux_2.6.18$
```

Рис. 17.15. Процесс компиляции был успешно завершен

В процессе компиляции выводилось слишком много информации, и просмотреть это все немного затруднительно, но иногда полезно. Так, в нашем случае, исходя из полученной информации в процессе компиляции, можно узнать, какие каталоги были созданы и не было ли ошибок в процессе сборки приложения.

Сейчас нас больше интересует то, куда были помещены бинарные файлы самой программы Avidemux. Если бы мы почитали выводимую информацию в процессе компиляции, то нам бы стало многое понятно. А не разумнее ли обратиться к инструкции по установке, которую предоставили нам разработчики? Да, точно, в инструкции от разработчиков есть строка "the result will be in the *install* folder". Это означает, что в результате компиляции бинарные файлы собранной программы будут помещены в каталог `install`. Так как мы запускали процесс компиляции из каталога с исходными текстами, то логичнее всего искать каталог в каталоге с исходными текстами (рис. 17.16).

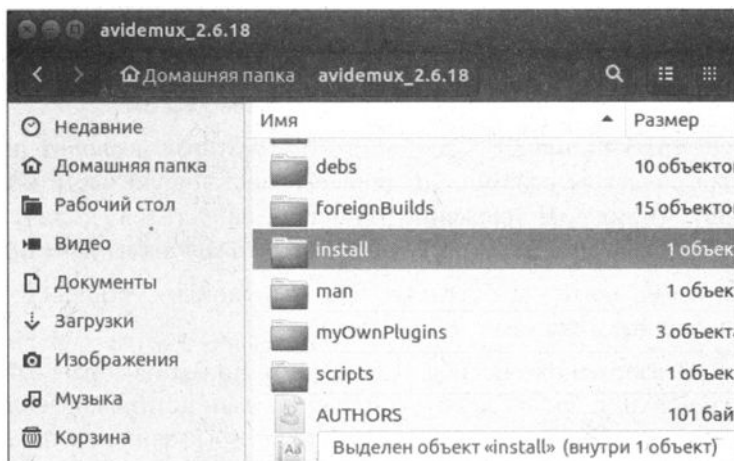


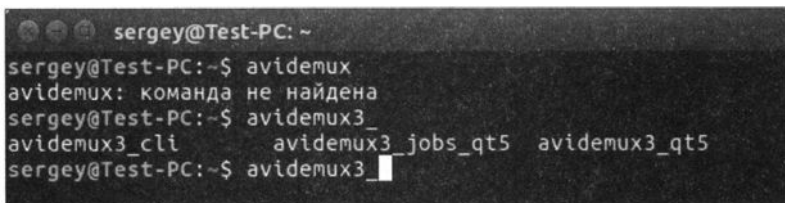
Рис. 17.16. Каталог `install` находится в каталоге с исходными текстами программы

В этом каталоге находится каталог `usr`, в котором имеется четыре подкаталога: `bin`, `include`, `lib` и `share`. Вам не кажется знакомой такая структура каталогов? Правильно, в самом начале знакомства с ОС Ubuntu Linux мы рассматривали структуру файловой системы этой операционной системы и выяснили, что в корневом каталоге `/usr/` обычно располагаются программы, которые устанавливает сам пользователь. Для удобства скопируем содержимое каталога `/home/sergey/avidemux_2.6.18/install/usr/` в корневой каталог `/usr/`:

```
sudo cp -r /home/sergey/avidemux_2.6.18/install/usr/* /usr/
```

Скопировав файлы, вы наверняка задумаетесь над тем, каким же образом вам теперь запускать приложение Avidemux. Так как установка была произведена вручную, то, соответственно, найти данное приложение в главном меню не получится. Если выполнить в Терминале команду `avidemux`, то это не принесет результата, т. к. приложение называется немного по-иному, и вам будет сообщено, что данная команда не найдена. Но мы же точно знаем, что установленное приложение должно быть как-то похоже на *avidemux*. Вот тут нам на выручку приходят полученные ранее знания о "горячих" клавишах Терминала.

Как мы помним, если ввести часть слова в Терминале и три раза нажать клавишу `<Tab>`, то Терминал подскажет нам полное название команд, которые соответствуют заданному слову (рис. 17.17).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ avidemux  
avidemux: команда не найдена  
sergey@Test-PC:~$ avidemux3_  
avidemux3_cli      avidemux3_jobs_qt5  avidemux3_qt5  
sergey@Test-PC:~$ avidemux3_
```

Рис. 17.17. Терминал вывел список команд, которые начинаются на слово `avidemux`

В нашем случае Терминал "знает" только о трех командах:

- ◆ `avidemux3_cli` — запуск приложения в режиме командной строки;
- ◆ `avidemux3_jobs_qt5` — запуск окна приложения со списком заданий;
- ◆ `avidemux3_qt5` — запуск приложения в графическом режиме.

Нас интересует третья команда — `avidemux3_qt5`, которая позволит нам запустить Avidemux в графическом режиме. Допишите недостающую часть команды и нажмите клавишу `<Enter>`. Приложение Avidemux запустится, и для дальнейшего удобства его использования вы можете закрепить его значок на панели запуска.

Можете себя поздравить, мы только что выполнили установку приложения Avidemux из исходных текстов.

Для наглядности изложения способа установки нами был выбран наиболее сложный метод установки, в процессе которой был показан не только сам процесс компиляции и установки приложения, но и ошибки, которые могут возникать в процессе установки приложений из исходных текстов.

При установке приложений из исходных текстов следует в первую очередь опираться на ту информацию, которую предоставляют сами разработчики приложения. Ведь кроме разработчиков, никто так хорошо не ориентируется в собственной программе. Поэтому в первую очередь мы должны читать файл `Readme` или `Install` и, опираясь на приведенные инструкции, производить установку приложения. В нашем случае инструкция по установке находилась на сайте разработчика приложения.

Репозитории в ОС Ubuntu

Устанавливая приложения, мы несколько раз упомянули слово "репозиторий", но до сих пор обходили эту тему стороной. Настало время заполнить пробелы и разобраться в том, что такое репозиторий и для чего он предназначен.

Репозиторий — это специальным образом организованное место хранения пакетов программного обеспечения, предназначенное для конкретного дистрибутива Linux. Это хранилище позволяет упростить процесс установки приложений и обновлений операционной системы.

Во времена медленного и дорогого Интернета роль такого хранилища играли CD/DVD-диски, которые постепенно уступили место онлайн-хранилищам, т. е. современным репозиториям.

Когда прорабатывалась концепция такого хранилища приложений, как репозиторий, то упор делался на простоту и удобство установки нужного программного обеспечения. Простота и удобство использования репозитория складывается из того, что вам предлагаются протестированные программы, которые будут стабильно работать именно в вашем дистрибутиве. Вам не нужно волноваться о разрешении зависимости между пакетами, т. к. все это делается автоматически без вашего вмешательства. Немаловажным достоинством такого решения, как репозиторий, является то, что при выходе новой версии программного обеспечения вы будете проинформированы, и для обновления вам не придется прикладывать огромные усилия, за вас все сделает пакетный менеджер.

На протяжении всей книги мы столько говорили о репозиториях, но до сих пор не имеем представления, где они находятся в системе, "как они выглядят" и как ими управлять.

Мы уже знаем, что в ОС Ubuntu Linux все настройки хранятся в виде обычных текстовых файлов, и поэтому очень удобно производить конфигурацию системы с помощью командной строки. Однако никто не мешает использовать графический режим для управления репозиториями.

В ОС Ubuntu Linux для управления репозиториями в графическом режиме предназначена надстройка **Программы и обновления**, которая вызывается из окна **Параметры системы** либо из главного меню по запросу "*программы и обновления*" (рис. 17.18).

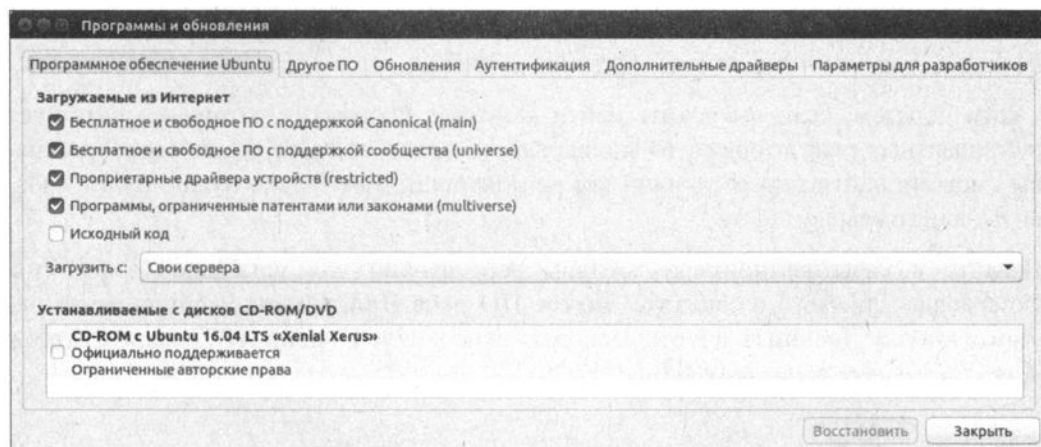


Рис. 17.18. Окно настроек репозитория

На первой вкладке окна **Программное обеспечение Ubuntu** вы можете управлять ветками официальных репозитория (*main*, *universe*, *restricted*, *multiverse* или *исходный код*) и указать ближайшее к вам место загрузки: основной сервер, сервер в РФ или другой сервер.

Следующая вкладка, которая нас интересует, — **Другое ПО** (рис. 17.19).

На этой вкладке отображается весь список репозитория Ubuntu, которые подключены к системе. Это означает, что помимо предустановленных официальных репо-

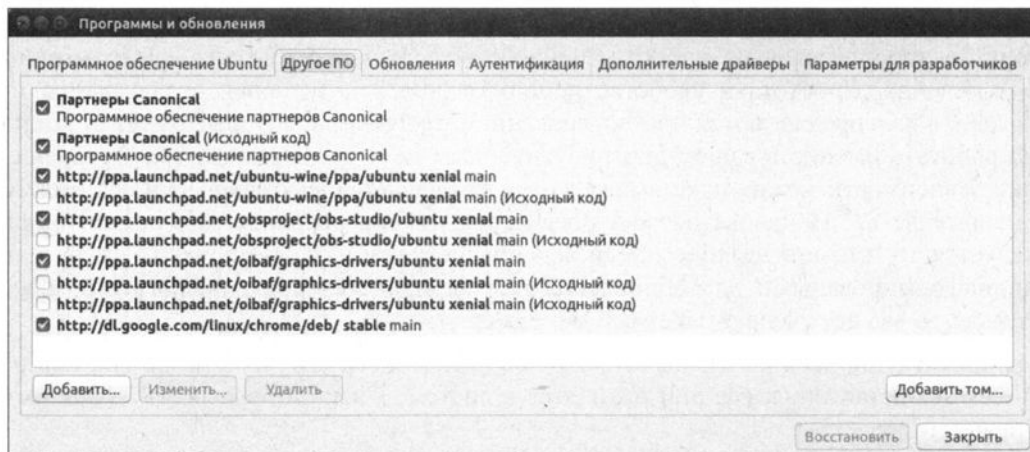


Рис. 17.19. Вкладка со списком репозиториев

зиториев мы можем подключить дополнительный репозиторий, из которого нам нужно получить тот или иной пакет.

Если вы обратили внимание, то адреса репозиториев начинаются с **http://ppa.launchpad.net**. Это связано с тем, что Canonical специально для репозиториев создала отдельный сайт **https://launchpad.net**, на котором любой разработчик программного обеспечения может с легкостью разместить собственный репозиторий. Обычно такие репозитории на сайте **launchpad.net** принято называть PPA (Personal Packages Archive — персональный архив пакетов).

Таким образом, если вы хотите найти какую-то программу, которая отсутствует в стандартных репозиториях, то вполне вероятно, что на сайте **https://launchpad.net** вы сможете найти интересующий вас репозиторий. Вам только нужно будет добавить на него ссылку.

Давайте в качестве примера добавим репозиторий для установки программы Notepadqq. Для этого на вкладке **Другое ПО** окна **Программы и обновления** нажмите кнопку **Добавить** и в открывшемся окне в поле **Строка АРТ** введите строку `ppa:notepadqq-team/notepadqq` (рис. 17.20).

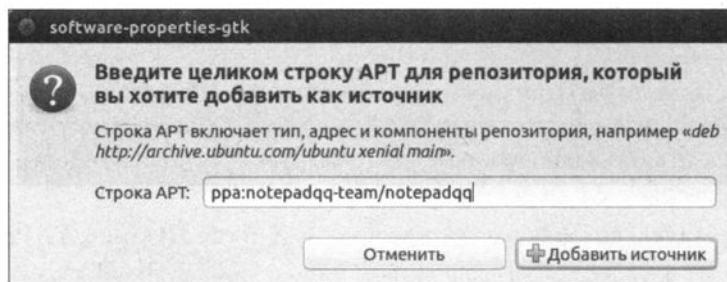


Рис. 17.20. Добавление нового репозитория

После этого нажмите кнопку **Добавить источник** и подтвердите добавление источника, введя свой пароль. Среди списка репозитория у вас появится два репозитория персонального архива пакетов (PPA) (рис. 17.21).

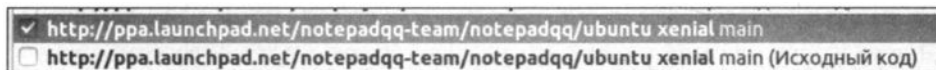


Рис. 17.21. Два новых источника репозитория

Нажмите кнопку **Заккрыть**. Появится информационное окно, сообщающее о том, что список источников программ и их обновлений изменился, и требуется его обновить (рис. 17.22).

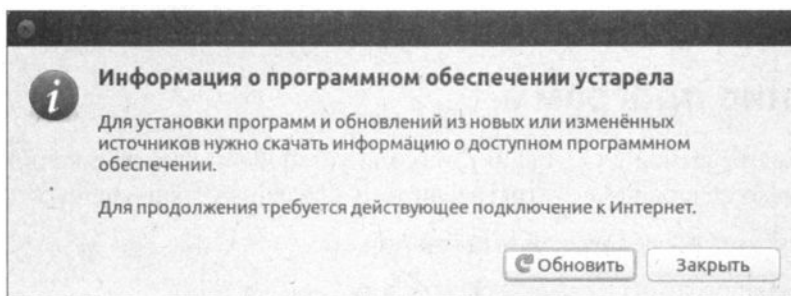


Рис. 17.22. Обновление списка PPA

Теперь установить программу Notepadqq можно следующей командой:

```
sudo apt install notepadqq
```

Если не обновлять список пакетов, то данная попытка установки программы завершится неудачей из-за того, что пакетный менеджер еще не знает об обновлении списка PPA.

Сейчас мы рассмотрели возможность добавления нового репозитория в графическом режиме, однако будет более правильным решением добавить новый источник с помощью Терминала следующей командой:

```
sudo add-apt-repository ppa:webupd8team/atom
```

Команда `add-apt-repository` позволяет добавлять новые источники PPA в файл `sources.list`, который расположен в каталоге `/etc/apt/`. Для подтверждения намерения добавить новый источник нажмите клавишу `<Enter>`, и новый источник будет добавлен.

Данная команда позволяет для редактора Atom добавить репозиторий, адрес которого выглядит следующим образом: **ppa:webupd8team/atom**. Как видите, добавление репозитория в режиме командной строки практически ничем не отличается от графического режима.

Как вы уже догадались, программа `apt` не подозревает о том, что изменились списки источников, и поэтому ей нужно "помочь", т. е. обновить список источников:

```
sudo apt update
```

Обновления списка источников программных пакетов выполнено, и мы можем приступить к установке программы Atom:

```
sudo apt install atom
```

Таким способом выполняется добавление репозитория для той или иной программы. Для удаления ненужного репозитория выполните следующую команду:

```
sudo apt-add-repository -r ppa:notepadqq-team/notepadqq
```

После этого не забудьте обновить список источников.

Таким нехитрым способом производится редактирование списка репозитория. Не забывайте удалять ненужные вам репозитории, т. к. наличие в системе неиспользуемых репозитория — плохой тон.

Удаление программ

За все время изучения ОС Ubuntu Linux мы установили большое количество программ, однако до сих пор не затрагивали тему удаления установленных программ.

Существует два способа удаления программ:

- ♦ удаление программы в графическом режиме;
- ♦ удаление программы в Терминале.

Самый простой способ — это, конечно же, графический: пару щелчков мыши, и программа удалена. Начнем с него. Итак, все вновь установленные приложения отображаются в Менеджере приложений Ubuntu на вкладке **Установлено** (рис. 17.23).

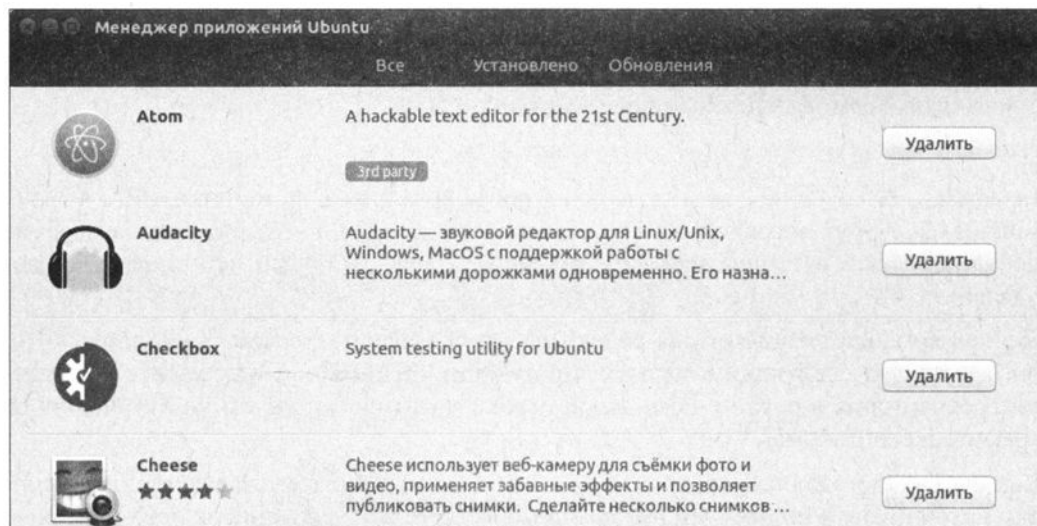


Рис. 17.23. Список установленных приложений в ОС Ubuntu

ПРИМЕЧАНИЕ

Обратите внимание на то, что вновь устанавливаемые приложения отображаются в этом списке не сразу, а только после перезагрузки системы.

Для удаления ненужного приложения достаточно нажать кнопку **Удалить** и подтвердить свое намерение, введя пароль. Проще простого.

А теперь поговорим о том, как можно удалить ненужное приложение с помощью командной строки, т. е. Терминала.

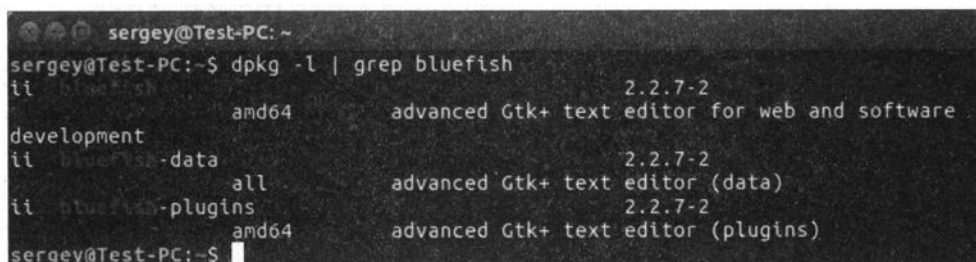
Итак, для начала нам нужно узнать точное название удаляемой программы. Для этого получим список установленных приложений:

```
dpkg -l
```

Данная команда выведет список всех установленных пакетов с указанием имени пакета, версии, архитектуры и краткого описания. Однако даже в только что установленной операционной системе этот список будет довольно объемным. Поэтому давайте уточним запрос:

```
dpkg -l | grep bluefish
```

Теперь запрос получится более конкретизированным (рис. 17.24).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ dpkg -l | grep bluefish  
ii bluefish 2.2.7-2  
amd64 advanced Gtk+ text editor for web and software  
development  
ii bluefish-data 2.2.7-2  
all advanced Gtk+ text editor (data)  
ii bluefish-plugins 2.2.7-2  
amd64 advanced Gtk+ text editor (plugins)  
sergey@Test-PC:~$
```

Рис. 17.24. Уточнение запроса вывода списка установленных пакетов

Зная точное название пакета, вы можете его удалить:

```
sudo apt remove bluefish
```

Обратите внимание, что при установке программы Bluefish было установлено три пакета: bluefish, bluefish-data и bluefish-plugins. Всех их нам придется удалять по очереди, поэтому гораздо удобнее будет выполнить объединенную команду удаления пакетов:

```
sudo apt remove bluefish bluefish-data bluefish-plugins
```

После выполнения данной команды все три указанных пакета будут удалены. Чтобы убедиться в удалении всех пакетов, снова выполните уже знакомую нам команду:

```
dpkg -l | grep bluefish
```

Выводы

Изучив данную главу, мы получили очень полезные практические навыки по установке и удалению программ в ОС Ubuntu Linux. Полученные умения нам пригодятся на протяжении всей работы с этой операционной системой, т. к. установка и удаление программ является основой работы в любой операционной системе.

Мы познакомились с репозиториями, научились их добавлять как в графическом режиме, так и в режиме командной строки. Мы также узнали, что такое зависимости при установке и удалении программ.

ГЛАВА 18



Установка дополнительного программного обеспечения

Дополнительное ПО в Ubuntu

В ОС Ubuntu Linux по умолчанию доступно практически все "из коробки". Это значит, что, установив систему, вы сразу же можете приступить к работе за компьютером, не прибегая к дополнительной настройке и установке программ. Однако для комфортной и привычной работы в этой операционной системе нам все же придется прибегать к установке дополнительных программ.

Не секрет, что каждый из нас привык пользоваться той или иной программой и не всегда привычные нам программы уже предустановлены в операционной системе. Поэтому мы устанавливаем наш любимый проигрыватель, браузер, понравившийся FTP-клиент и т. д.

Учесть потребности каждого пользователя крайне сложно, по этой причине разработчики ОС Ubuntu включили в свой дистрибутив только самые необходимые приложения, оставив вам право выбора установки любого понравившегося программного обеспечения.

Дополнительные медиамодули

В связи с тем, что программное обеспечение находится под защитой авторских прав и лицензионного соглашения, то последнее не всегда совпадает с "философией" ОС Ubuntu и открытостью ее программного кода. По этой причине в ОС Ubuntu по умолчанию не входят системные библиотеки, позволяющие проигрывать популярные форматы мультимедийных файлов, например, такие как AVI. Однако это не значит, что вы не сможете больше воспроизводить такие файлы. При этом вам совсем не обязательно искать в Интернете необходимый кодек и долго изучать инструкцию по его установке. ОС Ubuntu прекрасна тем, что при открытии такого файла она сама подсказывает, какую системную библиотеку необходимо установить. Вам остается только сделать пару щелчков мышью, и необходимый набор системных библиотек будет установлен.

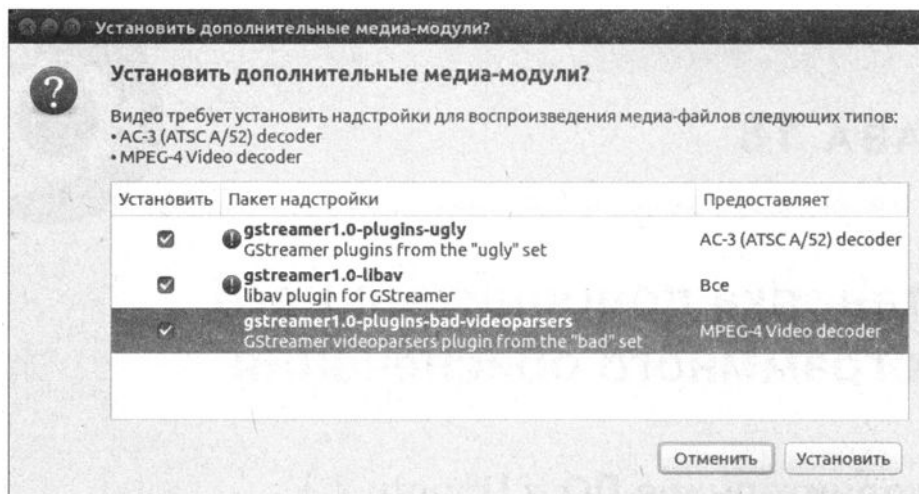


Рис. 18.1. Видеопроигрыватель предлагает установить дополнительные медиамодули

В качестве примера, запустите любой видеофайл с расширением avi или mp4, и медиаплеер предложит вам установить дополнительные медиамодули (рис. 18.1).

Выделите необходимые вам модули и нажмите кнопку **Установить**. Появится уведомление о том, что будут установлены дополнительные программы (рис. 18.2).

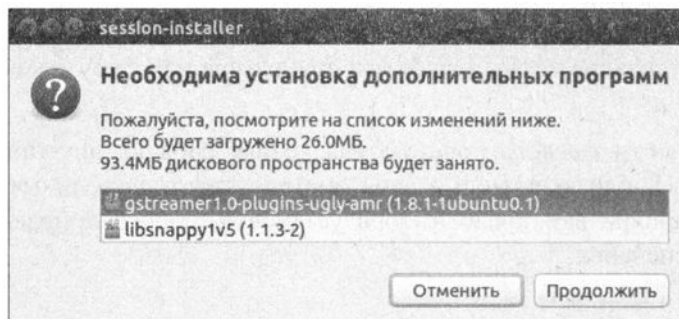


Рис. 18.2. Требуется установка дополнительных программ

Нажмите кнопку **Продолжить** и для подтверждения установки введите пароль вашей учетной записи. Начнется процесс установки отмеченных пакетов (рис. 18.3).

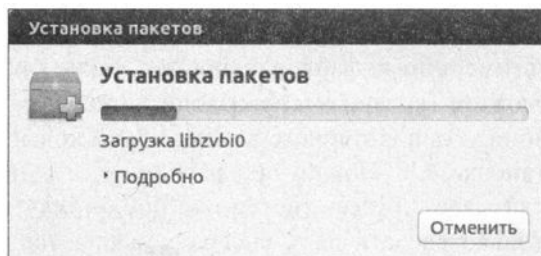


Рис. 18.3. Процесс установки отмеченных пакетов

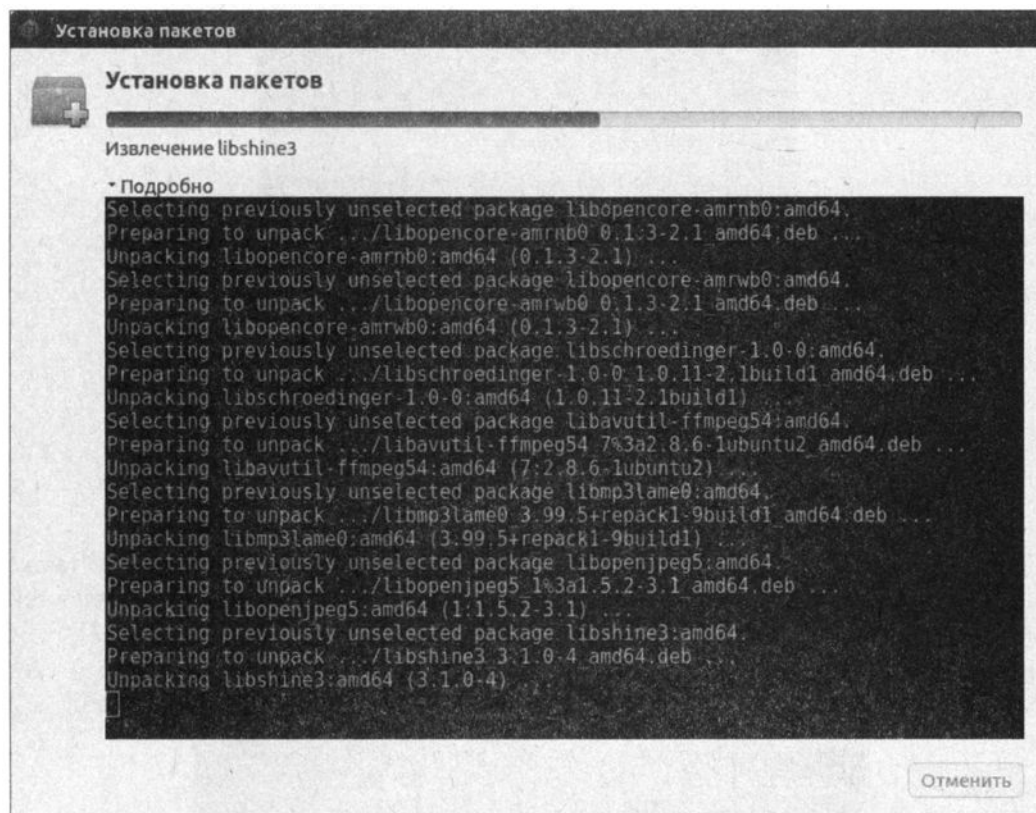


Рис. 18.4. Подробности установки необходимых медиамодулей

Для просмотра более детальной информации по установке нажмите на стрелочку разворачивающегося списка **Подробно** (рис. 18.4).

По окончании процесса установки вы сможете воспроизвести нужные вам мультимедийные файлы.

Мультимедийные проигрыватели

В стандартную поставку ОС Ubuntu входят два проигрывателя:

- ◆ Rhythmbox — проигрыватель для воспроизведения аудиофайлов;
- ◆ Totem — проигрыватель для воспроизведения видеофайлов.

Оба этих проигрывателя отлично справляются с возложенными на них задачами, однако многим пользователям этого функционала бывает недостаточно, и тогда приходится искать альтернативные мультимедийные проигрыватели. Далеко ходить не надо — все необходимое есть в Менеджере приложений Ubuntu. Так, пользователям, привыкшим к аудиопроигрывателю Winamp, по душе придется проигрыватель музыкальных файлов Audacious (рис. 18.5).

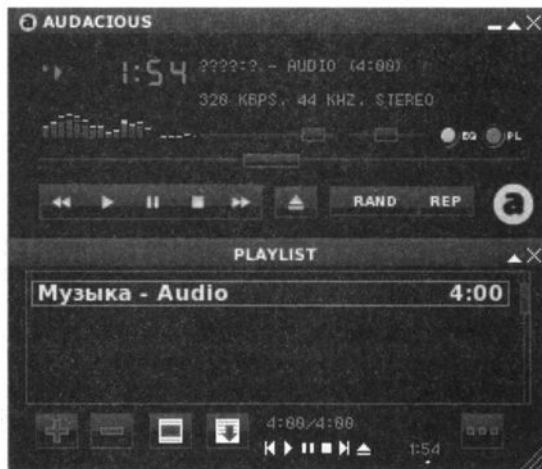


Рис. 18.5. Аудиопроигрыватель Audacious

Проигрыватель Audacious позволяет устанавливать обложки Winamp, а также допускает установку дополнительных плагинов для расширения возможностей данного аудиоплеера.

Для воспроизведения видеофайлов существует видеоплеер VLC (рис. 18.6).

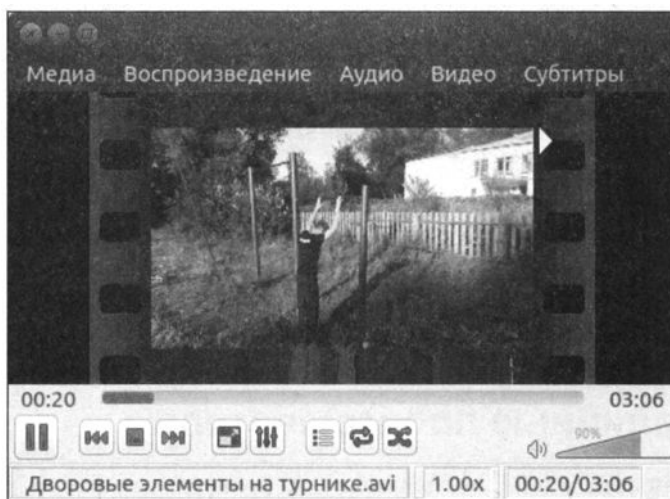


Рис. 18.6. Видеоплеер VLC

Данный видеоплеер поддерживает практически все популярные форматы и имеет собственные системные библиотеки для их воспроизведения. Это означает, что данное программное обеспечение абсолютно свободное, и, пользуясь данным проигрывателем, вы не нарушаете никаких авторских прав.

Проигрыватель VLC имеет множество настроек, и при желании вы сможете сконфигурировать его работу, исходя из личных предпочтений, и сохранить эти настройки для дальнейшего использования.

Мы рассмотрели всего лишь два проигрывателя, но на этом список доступных проигрывателей не заканчивается, т. к. Менеджер приложений Ubuntu постоянно дополняется новым ПО, и вы при желании сможете выбрать и установить любой понравившийся вам проигрыватель.

Web-браузеры

По умолчанию в ОС Ubuntu установлен браузер Firefox, но тем, кто привык к работе в браузере Google Chrome, браузер Firefox может показаться несколько неудобным. У вас есть два варианта решения данной проблемы:

- ♦ загрузить с официального сайта и установить Web-браузер Google Chrome;
- ♦ установить в Менеджере приложений Ubuntu Web-браузер Chromium.

Первый вариант является наиболее предпочтительным, и мы его уже рассматривали. Второй вариант больше подойдет тем, кто стремится пользоваться всем новым, т. к. именно в браузере Chromium первыми появляются экспериментальные новшества, которые уже затем переносятся в браузер Google Chrome.

BitTorrent-клиент

Многим из нас часто приходится загружать объемные файлы, и в силу тех или иных обстоятельств не всегда это удается сделать за один раз. Например, у вас имеется медленное интернет-соединение, и вы просто физически не можете скачать необходимый файл за один день, а оставлять включенным на ночь компьютер не всегда хочется. Благо есть такие программы, которые позволяют приостановить загрузку и возобновить ее в нужный для вас момент. Так, это умеет делать BitTorrent-клиент под названием qBittorrent (рис. 18.7).

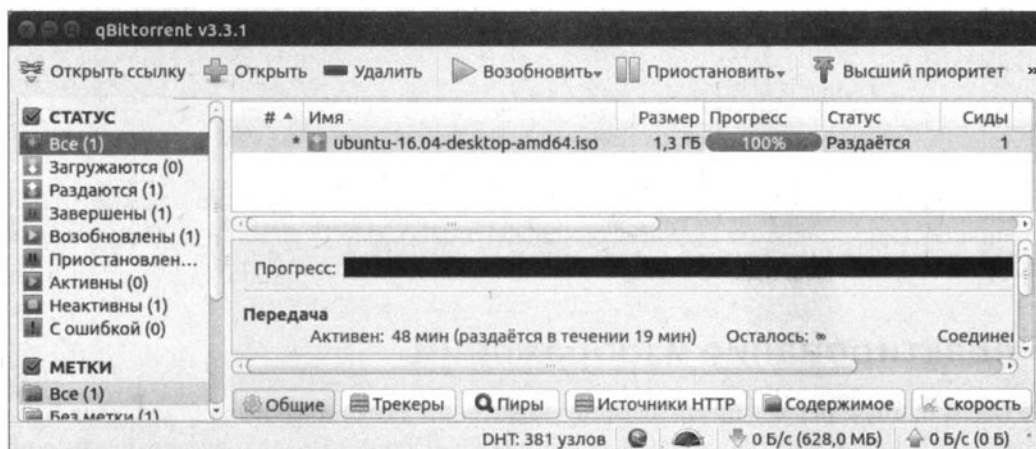


Рис. 18.7. BitTorrent-клиент qBittorrent

Разумеется, что qBittorrent — это не просто менеджер загрузок, а полноценный клиент для работы в пиринговых сетях, и предназначен он в первую очередь для обмена файлами через Интернет.

qBittorrent также доступен для загрузки в Менеджере приложений Ubuntu. Обратите внимание, что в данной программе отсутствует навязчивая реклама, которая зачастую встроена в подобные BitTorrent-клиенты.

Файловый менеджер

Файловым менеджером мы привыкли пользоваться еще со времен Norton Commander и DOS Navigator. Это уже потом пошли современные графические файловые менеджеры, до сих пор помогающие нам в решении тех задач, с которыми не справляется системный файловый менеджер.

Самым удачным решением для ОС Ubuntu является двухпанельный файловый менеджер Double Commander (рис. 18.8). Он не только обеспечивает более удобную работу с файлами, но и способен работать с протоколом FTP. Функционал Double Commander можно расширить за счет установки дополнительных плагинов.

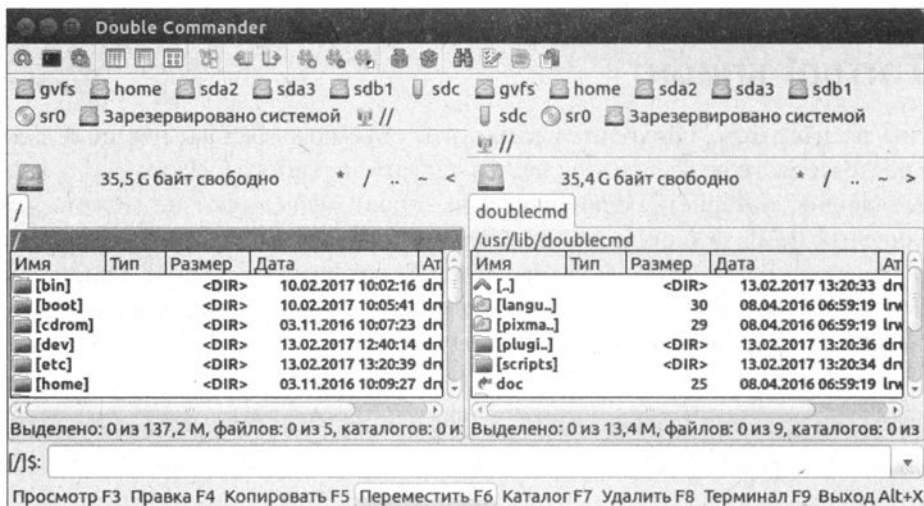


Рис. 18.8. Файловый менеджер Double Commander

По своему функционалу Double Commander не уступает программе Total Commander, но при этом Double Commander полностью бесплатен и доступен для загрузки в Менеджере приложений Ubuntu.

Редактирование изображений

В ОС Windows для редактирования изображений мы привыкли пользоваться программой Adobe Photoshop, однако ее нет для ОС Ubuntu. Правда, расстраиваться не стоит, т. к. существует не менее полезная программа, под названием GIMP.

Редактор изображений GIMP уже предустановлен в ОС Ubuntu.

В случае если вам необходим простой и несложный редактор изображений, то на помощь придет программа под названием Pinta, которая ничуть не сложнее, чем встроенный в ОС Windows редактор Paint. Редактор Pinta также доступен в Менеджере приложений Ubuntu.

Запись CD/DVD-дисков

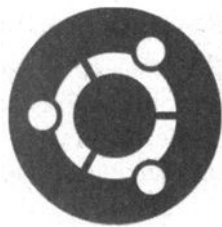
Хоть и CD/DVD-диски уходят в прошлое, тем не менее, иногда нам приходится прибегать к их записи. В ОС Windows для записи дисков многие используют программу Nero. В ОС Ubuntu Linux это программа Brasero. Она является довольно функциональной программой и имеет простой и понятный графический интерфейс. Установив программу Brasero с помощью Менеджера приложений Ubuntu, вы без труда сможете записать CD- или DVD-диск, а также создать его копию и сохранить его образ на вашем компьютере.

Выводы

Изучив эту главу, мы узнали, что для ОС Ubuntu существует множество программ, которые могут заменить предустановленное ПО в этой операционной системе. Описать все имеющиеся программы не представляется возможным и лишено смысла. Вы без труда сами сможете подобрать и установить интересующую вас программу.

Приятным достоинством программ для ОС Ubuntu является то, что они относятся к свободному программному обеспечению, и вы можете сколько угодно пользоваться данными программами, не нарушая при этом лицензионное соглашение.

ГЛАВА 19



Запуск Windows-приложений в ОС Linux

Wine для запуска приложений ОС Windows

В ОС Ubuntu Linux существует возможность запускать приложения, которые предназначены для работы в операционной системе Windows. Данную возможность предоставляет приложение Wine.

Wine (акроним Wine Is Not an Emulator; Wine — не эмулятор) — приложение ОС семейства Linux, которое является неким слоем совместимости с Windows API. Для обеспечения совместимости приложение Wine задействует системные библиотеки Windows, что позволяет запустить приложение Windows в "неродной ОС".

Использование Wine позволит запустить в ОС Ubuntu приложения, которые вам необходимы для работы в ОС Linux, но существуют только для ОС Windows.

Сейчас многим может показаться, что все приложения, предназначенные для ОС Windows, можно запускать в другой операционной системе, однако это не совсем так. Хотя Wine и является некой платформой для запуска Windows-приложений, но, тем не менее, это не гарантирует, что то или иное Windows-приложение будет работать под ОС Ubuntu Linux. Все проблемы совместимости связаны из-за закрытости ОС Windows, что не позволяет в полной мере обеспечить совместимость с Wine. Некоторые приложения используют недокументированные функции ОС Windows, а это уже усложняет задачу по обеспечению полной совместимости. Тем не менее, Wine поддерживает работу большинства популярных Windows-приложений, и разработчики Wine активно работают в этом направлении.

Установка Wine

В стандартных репозиториях ОС Ubuntu имеется доступная для вашего дистрибутива версия Wine. Для установки Wine из стандартного репозитория выполните следующую команду:

```
sudo apt-get install wine
```

После ввода пароля пользователя и подтверждения установки запустится процесс установки. Это займет некоторое время, все зависит от пропускной способности вашего интернет-соединения.

На запрос установки пакета шрифтов нажмите клавишу <Tab> для того, чтобы выделить кнопку **ОК**, а затем клавишу <Enter>, чтобы принять лицензионное соглашение (рис. 19.1).

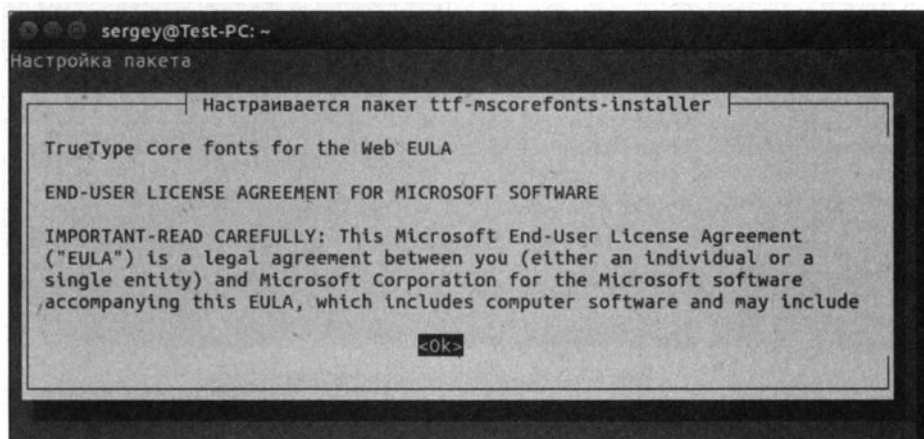


Рис. 19.1. Запрос на принятие лицензионного соглашения

После этого подтвердите установку пакета шрифтов ttf-mscorefonts-installer (рис. 19.2).

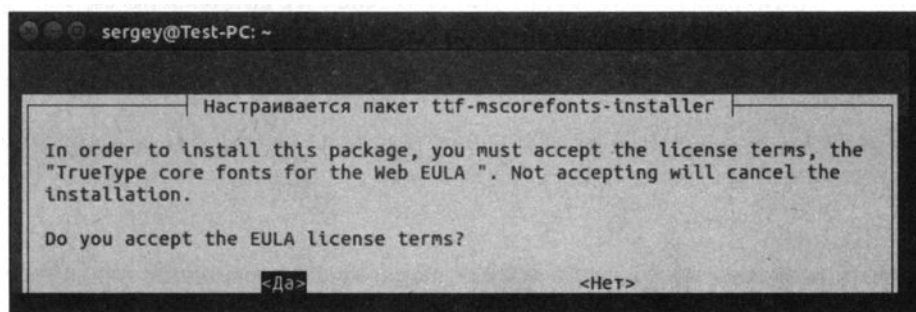


Рис. 19.2. Установка пакета шрифтов ttf-mscorefonts-installer

О завершении установки свидетельствует окно приветствия, готовое к вводу команд (рис. 19.3).

Сейчас мы рассмотрели установку из стандартного репозитория, но при этом следует учитывать, что проект Wine развивается довольно быстро и новые версии выходят очень часто. В стандартном репозитории может содержаться уже неактуальная версия Wine, не поддерживающая нужную вам программу.

Для того чтобы установить актуальную версию Wine, нужно подключить официальный репозиторий Wine:

```
sudo add-apt-repository ppa:ubuntu-wine/ppa
```

```

sergey@Test-PC: ~
Настраивается пакет libxslt1.1:i386 (1.1.28-2.1) ...
Настраивается пакет libxt6:i386 (1:1.1.5-0ubuntu1) ...
Настраивается пакет p7zip (9.20.1-dfsg.1-4.2) ...
Настраивается пакет ttf-wqy-microhei (0.2.0-beta-2) ...
Настраивается пакет winetricks (0.0+20141009+svn1208-2ubuntu1) ...
Настраивается пакет gnome-exe-thumbnailer (0.9.3-2) ...
Настраивается пакет libosmesa6:i386 (11.2.0-1ubuntu2.2) ...
Настраивается пакет libosmesa6:amd64 (11.2.0-1ubuntu2.2) ...
Настраивается пакет p11-kit-modules:i386 (0.23.2-5-ubuntu16.04.1) ...
Настраивается пакет wine1.6-i386:i386 (1:1.6.2-0ubuntu14) ...
Настраивается пакет odbcinstdebian2:amd64 (2.3.1-4.1) ...
Настраивается пакет odbcinst (2.3.1-4.1) ...
Настраивается пакет unixodbc (2.3.1-4.1) ...
Настраивается пакет wine1.6-amd64 (1:1.6.2-0ubuntu14) ...
Настраивается пакет wine1.6 (1:1.6.2-0ubuntu14) ...
Настраивается пакет wine (1:1.6.2-0ubuntu14) ...
Обрабатываются триггеры для libc-bin (2.23-0ubuntu4) ...
sergey@Test-PC:~$

```

Рис. 19.3. Установка Wine завершена

На запрос Терминала введите свой пароль учетной записи и для добавления репозитория нажмите клавишу <Enter>. После этого нам необходимо обновить список пакетов:

```
sudo apt-get update
```

Этим мы обновим список пакетов и сможем установить актуальную версию Wine, выполнив уже знакомую нам команду:

```
sudo apt-get install wine
```

На этом процесс установки Wine заканчивается, и мы переходим непосредственно к настройке самой программы Wine. Необходимо открыть окно **Настройки Wine**:

```
winecfg
```

Окно настроек можно также запустить из главного меню, выполнив поиск по слову *Wine* и выбрав в результатах поиска **Настройки Wine**. Откроется окно настроек (рис. 19.4).

При запуске соответствующей настройки в вашем домашнем каталоге будет создан каталог `/.wine/`, в котором появятся все необходимые для работы файлы и каталоги. Иными словами, в этом каталоге будет создана примерная структура организации файловой иерархии, которая необходима для корректной работы Windows-приложений.

В окне настроек Wine можно дополнительно настроить различные параметры, например, указать версию ОС, под которую будет эмулироваться работа приложений в Wine. Остальные параметры можно оставить по умолчанию, т. к. их редактирование в большинстве случаев не требуется.

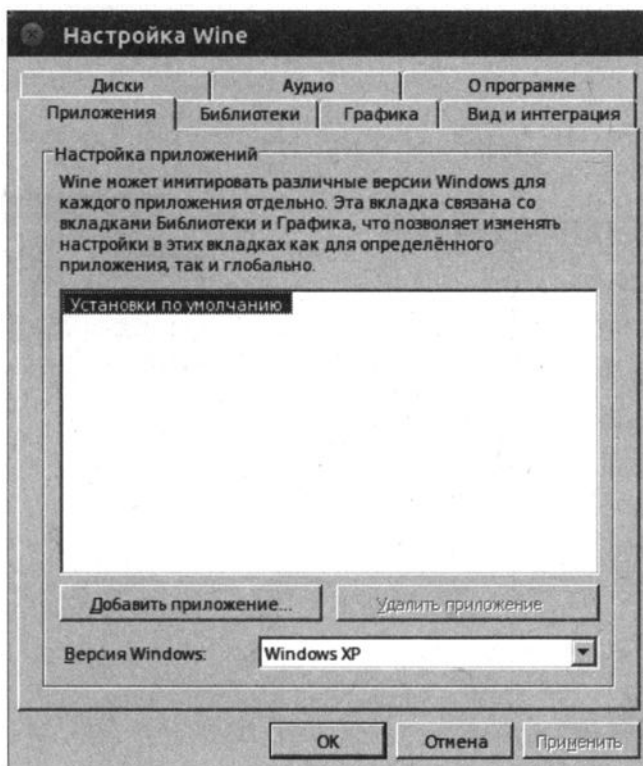


Рис. 19.4. Окно настроек Wine

Установка и запуск Windows-приложений

Вас наверно давно уже мучает вопрос: как устанавливать Windows-приложения в ОС Ubuntu Linux? Сейчас мы рассмотрим этот процесс на конкретном примере. Давайте установим карты 2ГИС. Для начала нам нужно зайти на сайт 2ГИС: <http://info.2gis.ru>. Сайт автоматически попытается определить ваше местоположение. В случае если местоположение определить не удалось или оно определено не верно, выберите правильное местоположение и щелкните по ссылке **Продукты**, а затем **Загрузки** (рис. 19.5).

Выберите версию для Linux, нажав на соответствующую ссылку.

Обратите внимание, что разработчики для удобства привели инструкции по установке карт 2ГИС.

Итак, для начала скачайте файл по ссылке **Оболочка 2ГИС** и сохраните его, например, в домашнем каталоге. Файл с базой данных скачивать не обязательно, т. к. саму базу можно загрузить в программе.

ПРИМЕЧАНИЕ

На момент написания этих строк доступной версией оболочки была версия 3.16.3.0, соответственно, файл имел название 2GISShell-3.16.3.0.orig.zip.



Рис. 19.5. Раздел загрузки карт 2ГИС

Теперь необходимо распаковать загруженный файл. Чтобы вручную не вводить название файла, давайте поступим следующим образом: откройте Терминал и выполните команду `ls` (рис. 19.6).

Как видите, первым в списке находится наш загруженный файл. Выделите его название и нажмите сочетание клавиш `<Ctrl>+<Shift>+<C>` для того, чтобы скопировать его название.



Рис. 19.6. Просмотр файлов в домашнем каталоге

ПРИМЕЧАНИЕ

Вставка скопированной строки производится нажатием клавиш <Ctrl>+<Shift>+<V>.

Выполните следующую команду:

```
unzip 2GISShell-3.16.3.0.orig.zip
```

В домашнем каталоге будет создан каталог 2gis с вложенным каталогом 3.0, в котором и будут находиться файлы карт 2ГИС.

Разумеется, все то же самое можно было сделать с помощью графического файлового менеджера Nautilus.

Теперь нам нужно запустить только что разархивированный файл с программой 2ГИС. Для этого в Терминале выполните следующую команду:

```
wine ~/2gis/3.0/grym.exe
```

Запустится Центр обновлений 2ГИС, в котором необходимо выбрать нужные пакеты для загрузки (рис. 19.7).

Установите флажок **Оболочка 2ГИС**, а потом флажок напротив нужного вам города и нажмите кнопку **Установить**.



Рис. 19.7. Окно центра обновлений

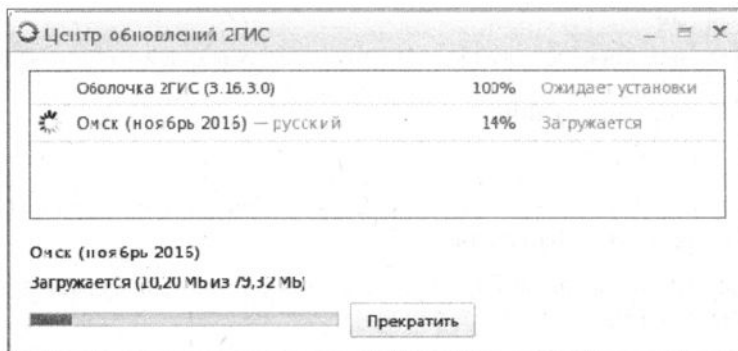


Рис. 19.8. Загрузка выбранных пакетов

Начнется загрузка, а затем установка выбранных вами пакетов (рис. 19.8).

Дождавшись окончания процесса, перезапустите программу 2ГИС.

На этом установка 2ГИС завершена.

Для запуска приложения 2ГИС из Терминала используйте следующую команду:

```
wine ~/2gis/3.0/grym.exe
```

Стоит отметить, что при установке программы 2ГИС на рабочем столе создается соответствующий ярлык, щелкнув по которому, вы сможете запустить установленную нами программу. Если вам по душе запускать установленную программу щелчком мыши, то перейдите в каталог `~/2gis/3.0/` и дважды щелкните левой кнопкой мыши по файлу `grym.exe`.

Установка других Windows-приложений выполняется аналогичным образом. В случае возникновения проблем, связанных с той или иной программой, следует поискать решение в Интернете, т. к. с большей долей вероятности можно рассчитывать на то, что решение вашей проблемы уже есть.

Выводы

Мы научились производить установку Wine, которая является некой программной платформой совместимости с Windows API, что позволяет запускать приложения, предназначенные для работы в ОС Windows.

Мы также научились осуществлять запуск установленных Windows-приложений в ОС Ubuntu Linux.

ГЛАВА 20



Работа с архивами в ОС Ubuntu

Об архивах в ОС Ubuntu

В ОС Ubuntu, как и в любой другой операционной системе, используются архивы. Архивы предназначены не только для уменьшения объема занимаемого файла, но и в большей степени для удобства распространения нескольких файлов. Согласитесь, ведь гораздо удобнее упаковать сразу несколько файлов и выложить в Интернете получившийся архив, нежели выкладывать их по одному.

Давным-давно, когда объем жесткого диска был слишком мал, для экономии места на нем практически все файлы архивировали и хранили их таким образом. Однако это было не самым удобным способом, но за неимением лучшего хранение файлов в архиве было наиболее логичным методом.

Таким образом, архивы всегда были и будут в любой операционной системе.

Менеджер архивов

ОС Ubuntu по умолчанию поддерживает практически все известные нам типы архивов. Для удобства пользования архивами в графическом режиме в системе присутствует программа Менеджер архивов (рис. 20.1).

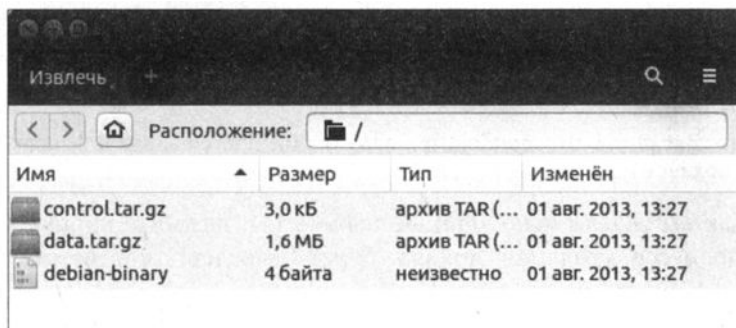


Рис. 20.1. Программа Менеджер архивов

Данная программа позволяет работать с различными типами архивов в ОС Ubuntu Linux. С помощью Менеджера архивов вы сможете создавать новые архивы, просматривать и изменять содержимое уже существующих архивов, извлекать файлы из архива.

Менеджер архивов позволяет работать с двумя типами архивов:

- ◆ несжатые типы архивов: ar, iso и tar;
- ◆ сжатые типы архивов: jar, rar, tar.gz, tgz, tar.bz, tbz, tar.bz2, tbz2, tar.lzo, tzo, tar.7z, tar.xz, cab, cbz, zip и zoo.

Можно расширить функции Менеджера архивов, если установить дополнительные модули. Однако на практике заложенного функционала Менеджера архивов вполне хватает для нормальной работы и, как правило, установка дополнительного ПО не требуется.

Работать с Менеджером архивов просто. Так, для создания архива из каталога или файла необходимо щелкнуть по объекту правой кнопкой мыши и в контекстном меню выбрать пункт **Создать архив**. Откроется диалоговое окно создания нового архива (рис. 20.2).

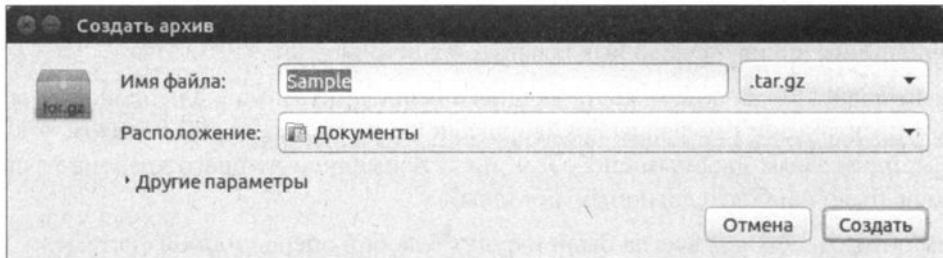


Рис. 20.2. Создание архива

В поле ввода **Имя файла** вы можете задать имя создаваемого архива, а в раскрывающемся списке рядом выбрать формат архива. В выпадающем списке **Расположение** указывается место, куда будет помещен создаваемый архив.

Вы также можете задать и другие параметры, такие как установка пароля для архива, шифрование списка файлов или разделение создаваемого архива строго по определенному размеру. Эти опции доступны в раскрывающемся списке **Другие параметры**.

ПРИМЕЧАНИЕ

Обратите внимание, что дополнительные опции доступны не для всех типов создаваемых архивов.

После того как вы задали необходимые параметры, нажмите кнопку **Создать**. По окончании процесса создания архива будет выведено соответствующее окно (рис. 20.3).

Для просмотра содержимого архива дважды щелкните левой кнопкой мыши по нужному архиву, и Менеджер архивов отобразит его содержимое.

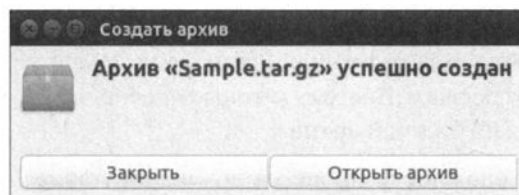


Рис. 20.3. Архив был успешно создан

Извлечение данных из архива в графическом режиме ничуть не сложнее, чем создание архива. Для извлечения файлов или каталогов из архива щелкните по нему правой кнопкой мыши и в контекстном меню выберите пункт **Извлечь сюда**. Архив будет распакован. То же самое можно проделать и из открытого окна Менеджера архивов, если на панели инструментов нажать кнопку **Извлечь** или выбрать пункт меню **Файл**, а затем выбрать пункт меню **Распаковать файлы**.

Как видите, в работе с архивами в графическом режиме нет ничего сложного. А теперь поговорим на тему того, какие бывают типы архивов.

Несжатые и сжатые архивы

Под словом "архив" мы привыкли понимать какие-либо файлы, которые содержатся в нем в сжатом виде. Однако в мире ОС Ubuntu Linux слово "архив" еще не означает, что в нем содержатся *сжатые файлы*. Так, существуют форматы архивов под названием "несжатые архивы". Что это значит? Давайте разберемся.

Перед тем как начинать знакомство с ОС Ubuntu, мы загрузили образ операционной системы, который имел формат ISO. Если вы обратили внимание, то у этого образа был такой же размер, как и впоследствии у записанного на DVD-диск. Это наиболее яркий пример *несжатого архива*. В таком архиве файлы не сжимаются, а просто создается несжатый архив, в который помещаются выбранные файлы и каталоги. Если такой образ в формате ISO был бы сжатым, то перед тем как записывать его на DVD-диск, нам бы потребовалось для начала его распаковать. Согласитесь, ведь это не совсем удобно. Именно для удобства были придуманы несжатые архивы.

В операционных системах семейства Linux наиболее распространенным несжатым форматом является архив tar. Сам архиватор tar по сути не является архиватором, т. к. при создании архива он не использует сжатие файлов. И только после того как архиватор tar создаст архив, в дело вступает другой архиватор, например, gzip, который уже сжимает созданный архив. Поэтому при создании сжатого архива имя файла состоит из двух расширений: tar.gz.

Вы сразу спросите: почему архиватор tar сразу не создает сжатый архив? Дело в том, что архиватор tar при создании архива сохраняет структуру файловой системы, а также некоторые атрибуты, такие как права доступа. И только после этого полученный файл сжимается другим архиватором.

Напрашивается еще один логичный вопрос: почему бы сразу не использовать архиватор gzip? Разгадка кроется в том, что многие архиваторы не умеют упаковывать

сразу несколько файлов, т. к. они могут работать с одним входным потоком данных, т. е. обрабатывать за раз только один файл. Вот и получается, что данные сначала собираются в несжатый архив и только после этого, из несжатого архива, другой архиватор создает сжатый архив.

При всей кажущейся сложности процесса архивации такие "двойные" архивы позволяют создать точную структуру файловой системы с сохранением атрибутов файлов и каталогов.

Исторически сложилось, что для создания архива требуется как минимум два архиватора. Отсюда и пошли такие понятия, как *сжатый* и *несжатый* архивы.

А теперь поговорим о самих архиваторах и научимся работать с ними с помощью Терминала.

Архиватор tar

Архиватор tar (англ. *tape archive* — архив на ленте) — это наиболее распространенный архиватор в семействе ОС Linux. Он стал стандартом де-факто в дистрибутивах на базе ОС Linux.

Данный архиватор позволяет архивировать файлы и каталоги, обновлять архивы, распаковывать их с сохранением файловой структуры со всеми атрибутами.

Изначально, архиватор tar был предназначен для создания архивов на лентах, отсюда и его соответствующее название — *tape archive*. Данный архиватор позволял создавать архив на любом устройстве, будь то дискета или обычный жесткий диск. Однако по умолчанию, если не указывать параметр `-f`, архиватор tar будет пытаться создать архив на ленточном устройстве, а именно на устройстве `/dev/rmt0`. Разумеется, таким устройством уже никто не пользуется, и поэтому мы будем всегда указывать ключ `-f`.

Современная версия архиватора tar мало чем отличается от предыдущих версий этого архиватора.

Синтаксис программы tar следующий:

```
tar [ключ] [файл]
```

Параметр `[файл]` принимает имя одного либо нескольких файлов, которые должны быть архивированы. Этот параметр также может принимать и имена каталогов.

Вкратце рассмотрим основные значения, которые принимает параметр `[ключ]` (табл. 20.1).

Мы привели только основные параметры команды tar. Более детальную информацию о ней можно получить по команде справки: `tar -help`.

Теперь попробуем создать архив из нашего каталога `/home/sergey/sample/`. Для этого выполните в Терминале следующую команду:

```
tar -cf new_archive.tar sample/
```

Таблица 20.1. Значения параметра [ключ] команды tar

Сокращенный вариант	Полный вариант	Описание
-A	--catenate, --concatenate	Позволяет присоединить tar-файлы к уже существующему архиву
-b	--blocking-factor= <i>размер_блока</i>	Позволяет разбить создаваемый архив на блоки, равные указанному значению <i>размер_блока</i>
-c	--create	Позволяет создать новый архив
-C	--directory= <i>каталог</i>	Позволяет извлечь содержимое в указанный каталог
	--delete	Позволяет удалить ненужный объект из архива
-f	--file= <i>архив</i>	Позволяет создать архив в виде файла на диске. Задаёт имя архива, который нужно создать или извлечь
-j	--bzip2	Позволяет пропустить созданный архив через архиватор bzip2. К имени файла добавляется расширение .bz2
-J	--xz	Позволяет пропустить созданный архив через архиватор xz
-k	--keep-old-files	Позволяет обновить архив без перезаписи уже существующих файлов
	--overwrite	Позволяет перезаписать существующие файлы при извлечении их из архива
-p	--preserve-permissions, --same-permissions	Позволяет извлечь информацию о правах доступа к файлу
-r	--append	Позволяет добавить файлы в конец архива
-t	--list	Позволяет вывести список содержимого архива
-u	--update	Позволяет обновить в архиве файлы на более новые
-v	--verbose	Позволяет вывести подробную информацию об обрабатываемых файлах
-x	--extract, --get	Позволяет извлечь файлы из архива
-z	--gzip, --gunzip, --ungzip	Позволяет пропустить созданный архив через архиватор gzip. К имени файла добавляется расширение .gz
-Z	--compress, --uncompress	Позволяет пропустить созданный архив через архиватор compress

Так как мы хотим создать архив, то указываем ключ `-c`. Ключ `-f` означает, что мы хотим создать файл, а не копию на ленточном устройстве.

ПРИМЕЧАНИЕ

Если указать полный путь к каталогу, например `~/sample/`, то архив будет создан с сохранением иерархии файловой системы, т. е. будут включены каталоги `home` и `sergey`.

Опции `-c` и `-f` являются обязательными и указываются именно в этом порядке, а не наоборот.

ПРИМЕЧАНИЕ

Для того чтобы легче запомнить порядок опций, запомните два слова: *create file*. Собственно говоря, именно эти два слова и означают опции `-cf`.

Итак, передав команде `tar` ключ создания файла архива, мы указываем имя создаваемого архива. В нашем случае, это `new_archive.tar`. Последним параметром указываем каталог, который должен быть архивирован.

После выполнения данной команды в домашнем каталоге будет создан несжатый архив `new_archive.tar`.

Для извлечения файлов из архива служит опция `-x` (от англ. *extract* — извлечь). Для извлечения файлов в текущий каталог выполните команду:

```
tar -xf new_archive.tar
```

Для контроля процесса спешности выполнения процесса создания архива или извлечения используйте ключ `-v`:

```
tar -xvf new_archive.tar
```

Ключ `-v` должен быть *перед* ключом `-f`. Почему так, а не после ключа `-f`? Дело в том, что архиватор применяет ключи именно в том порядке, в котором они указаны. Таким образом, если указать ключ вывода подробной информации последним, то и выводить будет нечего, т. к. файл архива уже будет создан. Вот такое получается исключение из правила "create file".

Сейчас мы научились создавать несжатые архивы, но это не значит, что сжатый архив придется делать в два этапа. Чтобы создать сжатый архив, достаточно указать один параметр, который будет указывать на тип архива. Например, параметр `-z` создаст сжатый `gzip`-архив, а `-j` на выходе даст `bzip2`-архив.

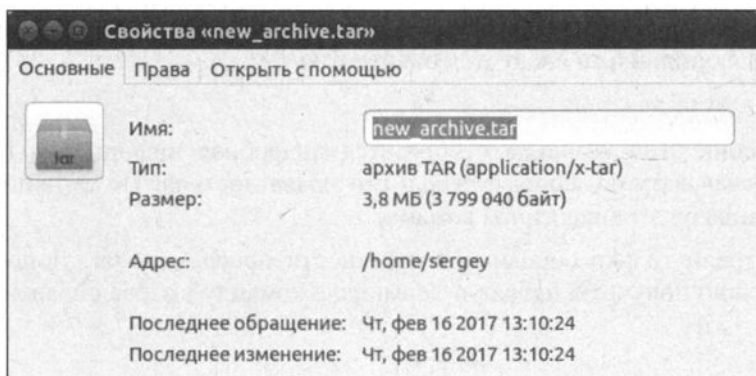
Давайте продемонстрируем создание сжатого `gzip`-архива одной командой:

```
tar -cvzf new_archive.tar.gz sample/
```

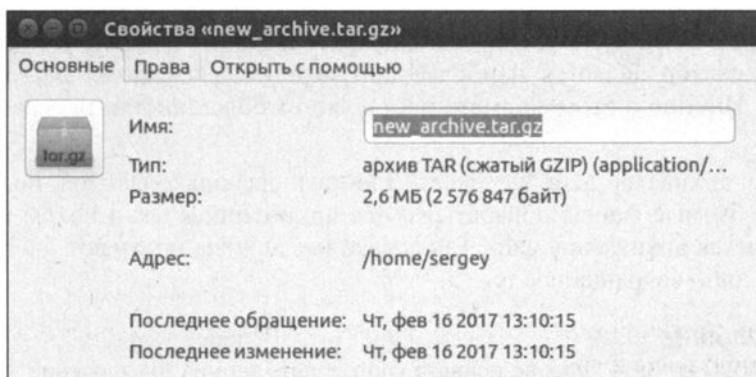
После выполнения данной команды вы получите сжатый архив. Для сравнения взгляните на рис. 20.4.

Обратите внимание на размер получившихся файлов.

Теперь проясним один момент: почему в имени файла при создании архива мы указываем все два расширения — `tar.gz`. Разумеется, можно не указывать расширение, и архив все равно будет создан, но это будет не совсем правильным решением. Лучше следовать тем правилам, которые были выработаны годами, и, согласитесь,



а



б

Рис. 20.4. Свойства несжатого (а) и сжатого (б) архивов

ведь гораздо удобнее понять тип архива, просто взглянув на его расширение, а не гадать, что перед нами. Из таких мелочей, как указание типа архива, и складывается удобство использования ОС Ubuntu.

Извлечение файлов из сжатого архива ничем не отличается от создания такого архива:

```
tar -xvf new_archive.tar.gz -C ~/new_directory/
```

В этом примере мы извлекаем сжатый архив `new_archive.tar.gz` в каталог `~/new_directory/`, добавляя перед ним параметр `-C`, который означает, что мы намерены извлечь содержимое в указанный каталог.

ПРИМЕЧАНИЕ

Каталог назначения для извлекаемых из архива файлов уже должен существовать. В противном случае команда извлечения файлов завершится ошибкой.

Если необходимо просто просмотреть содержимое архива, то достаточно выполнить следующую команду:

```
tar -tf new_archive.tar.gz
```

Данная команда выведет только список файлов архива, но если вам нужна более детальная информация, то следует добавить ключ `-v`:

```
tar -tvf new_archive.tar.gz
```

По выполнении этой команды отобразится подробная информация по каждому файлу, включая дату создания файла и его права доступа. По функциональности данная команда будет аналогична команде `ls -l`.

Мы рассмотрели только основные возможности программы `tar`. Дополнительные сведения можно получить, набрав в Терминале команду вызова справочной информации: `tar -help`.

Архиватор `gzip`

Архиватор `gzip` (англ. GNU Zip) — это программа сжатия и восстановления файлов. Данный архиватор является наиболее популярным для сжатия данных и имеет формат `GZ`. Именно в этом формате поставляется большинство программ в исходных текстах.

Сам по себе, архиватор `gzip` "не умеет" сжимать несколько файлов, поэтому перед сжатием требуемые файлы обрабатываются архиватором `tar`, а потом полученный файл передается архиватору `gzip`. Так созданные архивы получают двойное расширение `tar.gz` либо сокращенное `tgz`.

ПРИМЕЧАНИЕ

В мире ОС на базе Linux не принято употреблять термин "расширение" в отношении к файлам. Гораздо правильнее будет сказать "*суффикс*".

Зачем нужно использовать архиватор `gzip` отдельно от архиватора `tar`? Представьте ситуацию, когда у вас имеется несжатый архив, а вам его нужно выложить в Интернете или отправить по электронной почте. При этом у вас медленное подключение к сети, и вам дорого обходится каждый бит информации. Не отправлять же теперь несжатый архив, когда есть возможность сжать его? Верно. Такой архив можно пропустить через архиватор `gzip` и на выходе получить файл на порядок меньше размером.

Синтаксис команды `gzip` следующий:

```
gzip [ключ] [файл]
```

Параметр `[файл]` принимает имя файла, который нужно сжать.

Вкратце рассмотрим основные значения, которые принимает параметр `[ключ]` (табл. 20.2).

Давайте научимся работать с этой командой. Итак, у нас есть несжатый архив `new_archive.tar`, который мы хотим сжать. Для этого выполним следующую команду:

```
gzip -k new_archive.tar
```

В этом примере мы использовали ключ `-k`, который позволяет сохранить несжатый файл. Если выполнить приведенную выше команду без указания этого ключа, то после выполнения сжатия файл `new_archive.tar` будет удален.

Таблица 20.2. Значения параметра [ключ] команды *gzip*

Сокращенный вариант	Полный вариант	Описание
-a	--ascii	Позволяет архиватору использовать режим текста ASCII, при котором символы конца строки конвертируются в стандарт данной операционной системы
-c	--stdout	Позволяет вывести полученный архив в стандартный поток вывода, например в окно Терминала
-d	--decompress	Позволяет распаковать архив
-f	--force	Позволяет производить сжатие или распаковку архива даже в том случае, если на этот архив есть ссылки или такой архив уже существует. Сжатие будет происходить даже при чтении данных из архива
-k	--keep	Позволяет не удалять обработанные (сжатые) файлы
-l	--list	Позволяет выводить информацию об обрабатываемом файле, такую как размер сжатого файла, размер оригинального файла, коэффициент сжатия и имя оригинального файла
-n	--no-name	Позволяет при сжатии не сохранять исходное имя и время создания файла
-N	--name	Позволяет сохранить или восстановить оригинальное имя и время создания файла
-q	--quiet	Позволяет подавить выдачу предупреждающих сообщений
-r	--recursive	Позволяет рекурсивно обработать каталоги, когда задан шаблон имени файла
-S	--suffix=суффикс	Позволяет задать требуемый суффикс, отличный от .gz
-t	--test	Позволяет провести тест на проверку целостности сжатого архива
-v	--verbose	Позволяет задать подробный вывод информации об обрабатываемых файлах
	--rsyncable	Позволяет выполнить архивацию, используя протокол rsync
-1	--fast	Позволяет задать самую высокую скорость сжатия. В таком режиме сжатие данных незначительное. Если параметр не указан, уровень сжатия равен 6
-9	--best	Позволяет задать самый высокий коэффициент сжатия. При таком режиме сжатие данных занимает значительное время. Если параметр не указан, уровень сжатия равен 6

Обратите внимание, что нам даже не пришлось указывать суффикс сжатого архива, он добавился автоматически.

Теперь извлечем содержимое полученного архива:

```
gzip -kd new_archive.tar.gz
```

После выполнения данной команды мы получим несжатый архив `new_archive.tar`.

Мы рассмотрели лишь сжатие и распаковку с помощью архиватора `gzip`. Однако этих знаний, как правило, достаточно. В любом случае вы без труда самостоятельно сможете применить и другие ключи к команде `gzip`.

Архиватор bzip2

Помимо архиватора `gzip` существует еще и *архиватор bzip2*, который несколько медленнее `gzip`, но имеет более высокую степень сжатия. Данный архиватор при работе добавляет суффикс `.bz2`. За счет более высокой степени сжатия архиватор `bzip2` уступает по скорости другим архиваторам.

Синтаксис команды `bzip2` следующий:

```
bzip2 [ключ] [файл]
```

Параметр `[файл]` принимает имя файла, который должен подвергнуться сжатию.

Команда `bzip2` имеет практически те же самые параметры, что и команда `gzip`. Поэтому отдельно их расписывать не имеет смысла.

Для примера давайте выполним сжатие файла `new_archive.tar`:

```
bzip2 -k new_archive.tar
```

После выполнения команды в домашнем каталоге будет создан сжатый архив `new_archive.tar.bz2`.

Как видите, использование данной команды ничуть не сложнее, чем команда `gzip`.

Прочие архиваторы

В своей работе с ОС Ubuntu Linux вы иногда будете сталкиваться и с такими форматами архивов, как ZIP, RAR, 7Z и прочими не столь распространенными форматами архивов. Возможно, вам и не потребуется их создавать, но извлечь их содержимое вам придется, хотя бы по той причине, что в Интернете самым распространенным форматом архива являются ZIP и RAR.

Допустим, вам прислали файл в формате ZIP, и вам нужно извлечь его содержимое. Искать для этой цели компьютер с ОС Windows — не самое разумное решение. Однако отчаиваться не стоит, т. к. разработчики операционной системы предусмотрели возможность работы с ZIP-файлами. Для этих целей в ОС Ubuntu уже имеется команда `unzip`, которая позволяет извлечь содержимое ZIP-архива:

```
unzip archive.zip
```

Данная команда позволит извлечь содержимое архива в текущий каталог. Более подробную информацию о команде `unzip` можно узнать, вызвав справочную информацию: `man unzip`.

Для создания архива в формате ZIP предназначена одноименная команда `zip`:

```
zip -r zip_archive sample
```

Данная команда создаст ZIP-архив `zip_archive.zip` с содержимым каталога `~/sample/`. Параметр `-r` в данном случае указывает на то, что указанный каталог будет обработан рекурсивно, т. е. в архив попадут также все вложенные каталоги и файлы.

Приводить все параметры данной команды мы не будем по той причине, что использование всех ее возможностей вам вряд ли когда-нибудь понадобится, а посмотреть справочную информацию вы сможете в любой момент, выполнив команду `man zip`.

Второй тип архива, с которым возможно придется столкнуться, — RAR-архив. Пользователи ОС Windows точно о нем знают, хотя бы потому, что разработал данный формат архива наш соотечественник Евгений Рошал.

Для того чтобы работать с архивом в формате RAR, вам придется установить данный архиватор:

```
sudo apt install rar
```

После подтверждения установки будет установлен пакет `rar`, который предоставит вам возможность работы с RAR-архивами. Однако программа `rar` является триальной (пробной), и вам будет доступен не весь функционал. Поэтому если вам необходимо всего лишь извлечь данные из архива, то наилучшим решением будет установка программы `unrar`:

```
sudo apt install unrar
```

После установки программы `unrar` вы сможете извлечь RAR-архив, выполнив следующую команду:

```
unrar x rar_archive.rar
```

Данная команда позволяет извлечь содержимое архива `rar_archive.rar` в текущий каталог. В данном случае в домашний каталог пользователя.

ПРИМЕЧАНИЕ

Перед параметром `x` не ставится символ дефиса.

Более детально узнать о командах `rar` и `unrar` вы можете из справочной информации, выполнив команду `man rar` или `man unrar` соответственно.

Не менее распространенным форматом архива является 7Z-архив, который имеет суффикс `7z`. Для извлечения файлов из таких архивов вам не потребуется установка дополнительных архиваторов, т. к. данный формат поддерживается программой `7za`, которая предустановлена в ОС Ubuntu. Таким образом, для извлечения архива в формате 7Z выполните следующую команду:

```
7za x 7z_archive.7z
```

Данная команда извлечет все файлы с сохранением структуры каталогов, т. к. мы указываем параметр `x` без символа дефиса перед этим параметром. После параметра введено имя архива, который нужно извлечь.

В своей практике вы наверняка будете встречать различные "экзотические" типы архивов. О том, как узнать, к какому типу принадлежит тот или иной файл, мы поговорим чуть ниже.

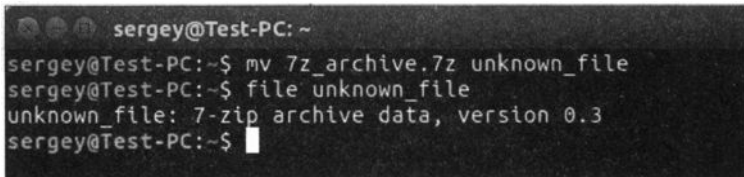
Неизвестный тип файла

Очень часто бывает так, что у файла либо отсутствует расширение (суффикс), либо оно не соответствует действительности. В связи с этим очень сложно определить, какой именно формат файла перед нами. Но даже из этой, казалось бы, безвыходной ситуации есть выход: команда `file`.

Команда `file` позволяет узнать формат неизвестного нам файла. Давайте в качестве примера узнаем формат файла `7z_archive.7z`, предварительно переименовав его в `unknown_file`:

```
mv 7z_archive.7z unknown_file
file unknown_file
```

Как видим, команда `file` смогла без труда распознать, какой перед нами тип файла (рис. 20.5).



```
sergey@Test-PC: ~
sergey@Test-PC:~$ mv 7z_archive.7z unknown_file
sergey@Test-PC:~$ file unknown_file
unknown_file: 7-zip archive data, version 0.3
sergey@Test-PC:~$
```

Рис. 20.5. Команда `file` распознала формат файла

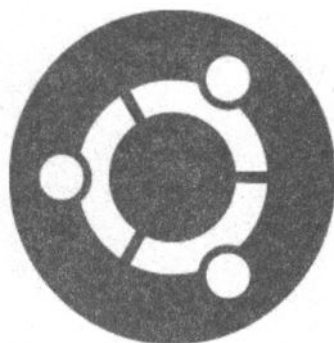
Теперь, после получения строки "7-zip archive data", мы уже точно можем сказать, что перед нами 7Z-архив, и для его извлечения мы применим соответствующую команду.

Полный список параметров команды `file` вы сможете узнать в справочной системе, выполнив команду `man file`.

Выводы

Мы познакомились с основными типами используемых в ОС Ubuntu архивов. Узнали, что в ОС Ubuntu уже есть удобный графический архиватор под названием Менеджер архивов, и научились основным операциям по работе с архивами: распаковывать и создавать архивы.

Помимо этого мы научились работать с архивами в режиме командной строки и познакомились с такими понятиями, как *сжатый* и *несжатый* архивы. Мы научились создавать и извлекать такие архивы, а также определять заранее неизвестный тип архива.

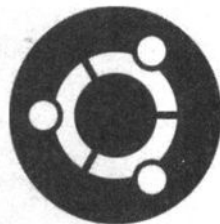


ЧАСТЬ III

Ubuntu для опытного пользователя

Глава 21.	Группы пользователей и права доступа
Глава 22.	Жесткие и символические ссылки
Глава 23.	Монтирование файловых систем
Глава 24.	Уровни выполнения и цели в ОС Ubuntu
Глава 25.	Службы в ОС Ubuntu
Глава 26.	Управление процессами в ОС Ubuntu
Глава 27.	Проверка работоспособности сети в ОС Ubuntu
Глава 28.	Подключение к удаленному рабочему столу
Глава 29.	Восстановление загрузчика GRUB
Глава 30.	Увеличение разрешения ОС Ubuntu в VirtualBox

ГЛАВА 21



Группы пользователей и права доступа

Пользовательские группы

Как и в любой другой многопользовательской операционной системе, в ОС Ubuntu Linux поддерживается многопользовательский режим. Это означает, что за компьютером может работать несколько человек под разными учетными записями, причем одновременно. Однако, все же необходимо ограничить права того или иного пользователя для того, чтобы он не имел доступа к нежелательной информации.

Согласитесь, обычному менеджеру совсем необязательно иметь доступ к файлам бухгалтерии или, например, к файлам отдела программистов. Но как же быть, если компьютер один, а пользователей много? Для таких целей был придуман механизм разделения прав доступа на так называемые группы пользователей.

Группы пользователей — это механизм разделения учетных записей на определенные группы, например для администраторов компьютера, сотрудников бухгалтерии, менеджеров и т. п.

Данный механизм разделения упрощает управление учетными записями и наделения их соответствующими правами доступа к той или иной информации. Так, системному администратору не требуется каждый раз при создании учетной записи заново назначать те же самые права, которые он назначил похожей учетной записи. Достаточно один раз создать определенную группу и назначить ей права, а затем добавлять в эту группу учетные записи, и права группы автоматически будут присвоены этой учетной записи. Замечательно, не так ли?

Права доступа в Linux

В ОС Ubuntu Linux у любого файла или каталога есть *владелец*, т. е. тот, кто создал этот файл или каталог, а также группа, к которой принадлежит владелец. Например, ваша учетная запись, созданная во время установки операционной системы, принадлежит к нескольким группам (рис. 21.1).

Как видим, в ответ на команду `groups sergey` Терминал выдал принадлежность учетной записи `sergey` к нескольким группам: `sergey`, `adm`, `cdrom`, `sudo`, `dip`, `plugdev`, `lpadmin`, `sambashare`.

```
sergey@Test-PC: ~  
sergey@Test-PC:~$ groups sergey  
sergey : sergey adm cdrom sudo dip plugdev lpadmin sambashare  
sergey@Test-PC:~$
```

Рис. 21.1. Просмотр групп учетной записи

Таким образом, можно узнать, к какой группе принадлежит учетная запись.

Самым простым способом узнать владельца файла и его группу можно, щелкнув по интересующему объекту правой кнопкой мыши, в контекстном меню выбрав пункт **Свойства** и перейдя на вкладку **Права** (рис. 21.2).

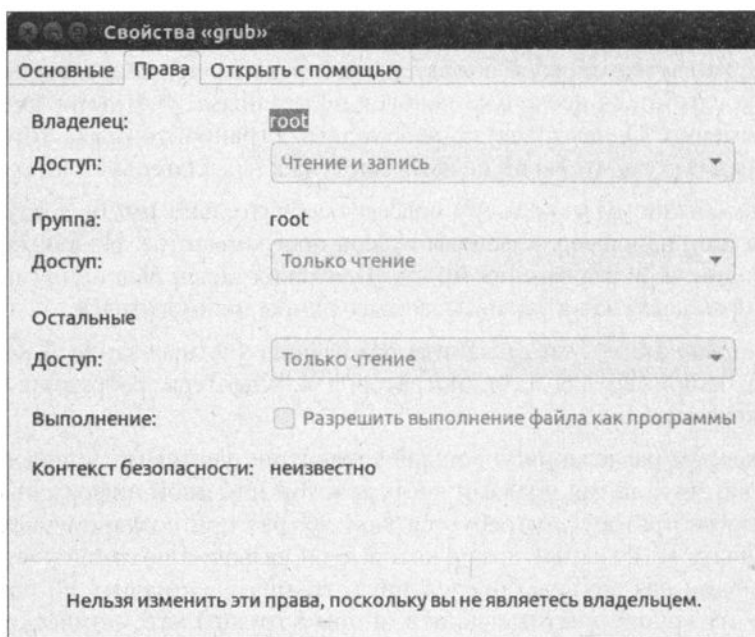


Рис. 21.2. Просмотр сведений о владельце файла

Так мы можем узнать, что у файла grub есть владелец root, который принадлежит к одноименной группе. Обратите внимание, что мы не можем редактировать данный файл, о чем свидетельствуют неактивные списки выбора и надпись "Нельзя изменить эти права, поскольку вы не являетесь владельцем".

Изменять права доступа можно на этой же вкладке при условии того, что у вас на это достаточно прав.

Буквенное представление прав доступа к файлу

Как мы уже знаем, для каждого файла или каталога можно установить права доступа. Права доступа автоматически задаются еще на этапе создания файла или каталога, и этот процесс для нас прозрачен. Система все делает сама, основываясь на правилах задания прав доступа для той или иной пользовательской группы. Но что делать, если вы создали файл и хотите открыть к нему доступ для остальных пользователей компьютера? В ОС Ubuntu для этих целей предусмотрен механизм редактирования прав доступа, с графической реализацией которого вы познакомились, открыв вкладку **Права** свойств файла. Это только самая малая часть того, что предусмотрено разработчиками операционной системы на базе Linux. На деле ОС Ubuntu Linux позволяет производить более гибкую настройку прав доступа, нежели это реализовано в свойствах файла на вкладке **Права**.

В ОС Ubuntu Linux для каждого файла соответствует определенный набор прав доступа, предоставленный в виде 9 битов. Каждые три бита из этого набора определяют права для владельца файла (**user**), группы (**group**) и всех остальных (**other**) (рис. 21.3).

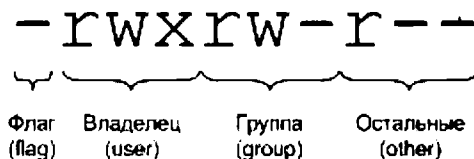


Рис. 21.3. Группы прав доступа

Первый символ (-) — это флаг (*flag*), который указывает на то, что перед нами: файл, каталог и т. п.

Буквенные символы `rw` означают следующее:

- ◆ чтение (read) — r ;
- ◆ запись (write) — w ;
- ◆ выполнение (execute) — x .

Для наглядности выполним в Терминале следующую команду:

```
ls -l /etc/default/grub
```

Сейчас мы не будем вдаваться в подробности синтаксиса самой команды, т. к. нас интересует ее ответ (рис. 21.4).

Нас интересует только первая часть ответа (первые 10 символов): `-rw-r--r--`. Данный ответ условно можно разделить на четыре части (табл. 21.1).

Первый символ, именуемый флагом (тип *flag*), может применять значения, перечисленные в табл. 21.2.

```

sergey@Test-PC: ~
sergey@Test-PC:~$ ls -l /etc/default/grub
-rw-r--r-- 1 root root 1304 июн 28 20:25 /etc/default/grub
sergey@Test-PC:~$

```

Рис. 21.4. Ответ команды `ls -l`Таблица 21.1. Права доступа файла `/etc/default/grub`

Флаг (flag)	Пользователь (user)	Группа (group)	Другие (other)
-	rw-	r--	r--

Таблица 21.2. Значения флага типа `flag`

Флаг	Описание
-	Отсутствие флага. Обычно указывает на то, что перед нами файл
l	Символическая ссылка (symbolic link). Указывает на то, что перед нами ссылка на файл. Символ <code>l</code> — это не цифра 1, а строчная латинская буква L
d	Указывает на то, что перед нами каталог (directory)
b	Блочное устройство (block device). Указывает на то, что перед нами какое-то блочное устройство, например DVD-диск, жесткий диск и т. п.
c	Символьное устройство (character device). Указывает на то, что перед нами символьное устройство, такое как стример, модем и т. п.
p	Канал, устройство fifo (fifo device). Указывает на то, что перед нами тип запоминающего устройства FIFO
s	UNIX-сокеты (unix domain socket). Указывает на то, что перед нами сокет межпроцессорного взаимодействия

Таким образом, в ОС Ubuntu Linux используется три группы флагов (не считая первый флаг типа *flag*, который указывает, что перед нами файл, каталог, блочное устройство и т. п.).

Итак, давайте разберем вывод команды `ls -l /etc/default/grub` по порядку.

Первый символ, а правильное сказать флаг, указывает, что именно перед нами. В нашем случае это символ дефиса (-). Это означает, что перед нами файл.

Первая группа флагов, а именно `rw-`, определяет права доступа владельца файла. В нашем случае разрешено чтение файла (символ `r`) и запись (символ `w`). Выполнение файла запрещено, т. к. файл не является исполняемым. О том, что запрещено выполнение файла, т. е. запуск, свидетельствует знак дефиса (-).

Вторая группа флагов, а именно `r--`, определяет права доступа группы пользователей файла. В данном случае всем пользователям группы разрешено только чтение файла (символ `r`). Запись и выполнение файла запрещены, о чем свидетельствуют следующие после символа `r` два дефиса (`--`).

Третья группа флагов, а именно `r--`, определяет права доступа для прочих пользователей, которые не принадлежат к группе пользователя файла. В нашем случае всем остальным пользователям разрешено только чтение файла (символ `r`), а запись и выполнение запрещены.

ПРИМЕЧАНИЕ

В случае если в правах доступа подряд стоят три символа дефиса (`---`), то при попытке чтения этого файла пользователь увидит сообщение в отказе доступа к файлу.

Теперь перейдем к рассмотрению прав доступа к каталогам.

Буквенное представление прав доступа к каталогам

Как мы уже знаем, символы `rwX` в правах доступа к файлу определяют права того или иного пользователя или группы на *чтение*, *запись* и *выполнение* соответственно. Однако не торопитесь применять это "правило" к каталогам. В отношении их дела обстоят несколько иначе. Так, если для каталога установлены права, например, `rw-`, то это еще не означает, что в данный каталог можно производить запись. Почему? Давайте разберемся вместе.

Итак, у нас в домашнем каталоге пользователя имеется каталог `Sample` с тремя файлами:

- ◆ `counter.php`;
- ◆ `Music file.mp3`;
- ◆ текстовый документ `txt`.

Для родительского каталога этих файлов, т. е. для `Sample`, установлены права доступа `r-----`. Данный набор прав означает, что разрешено только чтение (символ `r`). Наверняка вы подумали, что если вашей учетной записи разрешено чтение, то вы можете открыть данный каталог и просмотреть содержимое файлов, находящихся в нем. Так считаете вы, но ОС Ubuntu Linux считает иначе, и операционная система позволит только выполнить *чтение имен файлов* данного каталога, а при попытке прочитать *содержимое* одного из файлов этого каталога вы получите отказ (рис. 21.5).

"Как же так?!" — подумаете вы и для пущей убедительности выполните команду `ls -l ~/Sample/` (рис. 21.6).

Опять отказ в доступе к файлам внутри каталога `~/Sample/`.

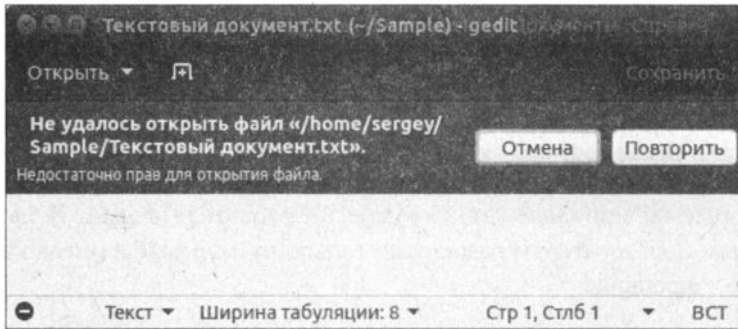


Рис. 21.5. Отказ в доступе к файлу Текстовый документ.txt

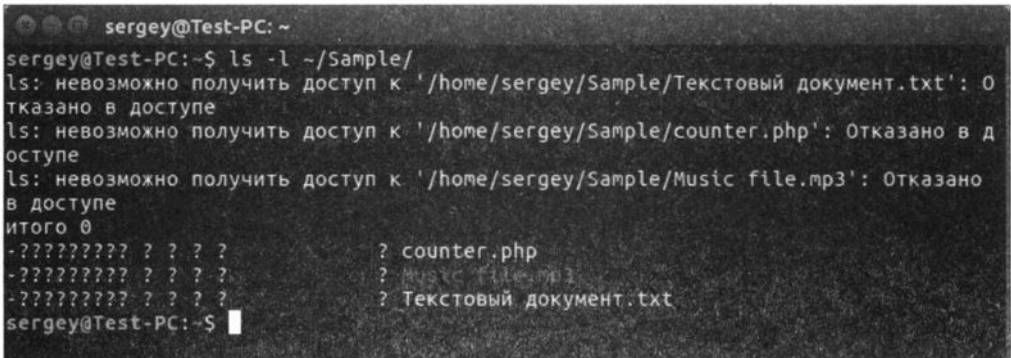


Рис. 21.6. Попытка посмотреть содержимое каталога ~/Sample/

ПРИМЕЧАНИЕ

Не забывайте, что символ ~ (тильда) — это сокращенный вариант указания домашнего каталога текущей учетной записи.

Почему же это происходит? Может быть, права доступа все же неверные, и там вовсе не r-----? Выполним команду `ls -l ~` (рис. 21.7).

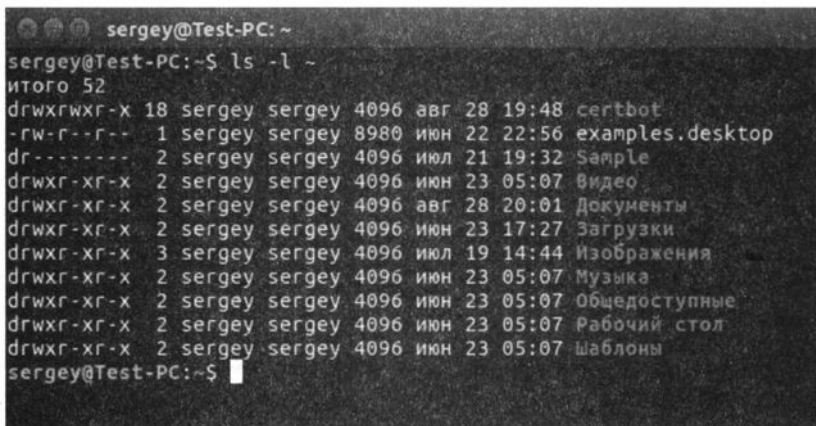


Рис. 21.7. Просмотр прав доступа домашнего каталога текущего пользователя

На четвертой строке вывода команды `ls -l ~` в первой колонке мы видим все те же права `r-----`.

ПРИМЕЧАНИЕ

Не забываем, что первый символ (флаг) `d` в перечислении прав доступа указывает на то, что перед нами каталог, а не файл `Sample`.

Настало время разобраться, в чем же дело и почему в случае с каталогами не действует тот же принцип прав доступа, который был применим к файлам.

Для каталогов флаги `rwX` имеют несколько иной смысл. Так, флаг `r` позволяет только *читать имена файлов* в каталоге, в чем мы уже убедились чуть ранее. Флаг `w` имеет смысл только в сочетании с флагом `x`, что позволит производить манипуляции с файлами, например, создавать, удалять и переименовывать их. Так, маска `rw-` будет равносильна маске `r--`, что означает права только на чтение имен файлов. Поэтому вы также не сможете прочитать содержимое файлов.

Таким образом, флаг `x` предоставляет доступ к самим файлам и их атрибутам, но данный флаг не имеет доступа к именам файлов, поэтому для нормальной работы с содержимым каталога применяются остальные флаги `r` и `w`.

Сейчас вам все сказанное покажется слишком сложным и запутанным, поэтому давайте рассмотрим таблицу прав файлов и каталогов, и все встанет на свои места (табл. 21.3).

Таблица 21.3. Соотношение прав файла и каталога с их маской

Маска (флаги прав)	Права на файл	Права на каталог
---	Полное отсутствие прав	Полное отсутствие прав
--x	Права на выполнение	Доступ к файлам и их атрибутам. Однако нет доступа на чтение имен файлов и на создание, переименование и удаление файлов в каталоге
-w-	Права на запись	Отсутствие прав
-wx	Права на запись и выполнение	Все права, за исключением права на чтение имен файлов. Присутствуют права на чтение, изменение и запуск файлов, а также на создание, удаление и переименование файлов при условии, если на затрагиваемые файлы имеются соответствующие права. Данное правило справедливо лишь при условии, что вам заранее известны имена нужных файлов в данном каталоге. Узнать имена файлов в каталоге невозможно
r--	Права на чтение	Права на чтение имен файлов
r-x	Права на чтение и выполнение	Права на чтение имен файлов, чтение файлов и их атрибутов. Однако нет прав на создание, переименование и удаление файлов в каталоге
rw-	Права на чтение и запись	Права на чтение имен файлов
rwX	Полные права	Полные права

Получается, что для просмотра каталога необходимо как минимум иметь права доступа `r-x`, т. е. права на чтение имен файлов и их атрибутов, но даже и в этом случае вам не удастся производить манипуляции с файлами. Для полноценной работы, как вы понимаете, необходимы полные права на доступ.

Весьма примечателен тот факт, что в режиме доступа `-wx` можно производить любые манипуляции в каталоге, за исключением чтения имен файлов. Это означает, что если вам заранее неизвестны имена файлов в каталоге, то вы не сможете их прочитать, удалить или переименовать. Однако создавать новые файлы в этом каталоге вы сможете без проблем.

Важно понимать отличия прав доступа для файлов и каталогов и в случае возникновения каких-либо проблем с доступом к информации уметь правильно трактовать права доступа и понимать, какие полномочия они предоставляют.

Цифровое представление прав доступа

Так уж сложилось, что в мире Linux удобнее использовать не буквенное, а цифровое представление прав доступа. Одной из причин использования цифрового представления прав доступа является компактность, например, маску `rw-r--r--` удобнее будет записать в виде трех цифр 644. Давайте разберемся, откуда появилось число 644 вместо привычного уже нам буквенного представления прав доступа.

Для примера возьмем права доступа для владельца файла `rw-`. Владелцу разрешено чтение (символ `r`), значит, мысленно записываем цифру 1. Так как разрешена еще и запись, то также мысленно запоминаем еще цифру 1, а вот выполнение запрещено, поэтому запоминаем цифру 0. Получается три числа: 110. Если перевести двоичное число 110 в восьмеричное, то получится число 6.

Делаем то же самое и с оставшейся частью выражения — `r--r--`. Здесь для группы владельца и всех остальных выходят числа 100 и 100. При переводе числа 100 в восьмеричную систему счисления получается 4. Значит, для выражения `r--` эквивалентом будет число 4. Теперь запишем все три числа вместе. Получится 644, т. е. права для владельца файла (число 6), права для группы владельца (число 4) и права для всех остальных (также число 4).

Для перевода из двоичной системы счисления в восьмеричную удобно воспользоваться табл. 21.4.

Таблица 21.4. Системы счисления и права доступа

ОСТ (двоичная)	BIN (восьмеричная)	Mask (маска)
0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--

Таблица 21.4 (окончание)

ОСТ (двоичная)	BIN (восьмеричная)	Mask (маска)
5	101	r-x
6	110	rw-
7	111	rwX

По аналогии переведем маску `rwXrw-r--` из символьного представления в цифровое. Итак, для владельца файла будет число 7, для группы владельца файла число 6, а для всех остальных 4. Получается три цифры — 764. Это и есть права доступа, которые не только удобно запоминать, но и использовать.

Символьные биты SUID, SGID и sticky

В технической литературе, а также на просторах Интернета, вы будете часто наблюдать картину, когда права доступа обозначаются не тремя, а четырьмя цифрами, например 0764. Не пугайтесь при виде этих четырех цифр, т. к. первая цифра — это всего лишь *специальный бит*, который влияет на запуск файла или указывает на владельца объектов в каталоге.

В любой современной операционной системе символьные имена пользователей имеют свои уникальные идентификаторы. Так, пользователь `root` в ОС Ubuntu Linux имеет идентификатор равный нулю, т. е. цифре 0. Числа с 1 по 999 предназначены для системных нужд, а все остальные до числа 65 535 могут быть присвоены пользовательской учетной записи. Таким образом, пользовательская учетная запись `sergey`, созданная при установке ОС Ubuntu, будет иметь идентификатор 1000.

ПРИМЕЧАНИЕ

В разных семействах ОС Linux идентификаторы могут различаться. Так, в ОС Red Hat зарезервированные числа будут находиться в диапазоне от 101 до 499.

Сделано все это для удобства самой ОС, т. к. операционная система на основании уникальных идентификаторов назначает права доступа. Имена учетных записей могут быть одинаковыми, а вот идентификаторы уже будут отличаться. Этим самым ОС "отличает" друг от друга учетные записи.

SUID (Set User Identifier) — бит смены идентификатора пользователя. Смысл этого бита пользователя состоит в том, что когда пользователем запускается какая-либо программа на выполнение, то она получает такие же права, как и сам пользователь, запустивший ее. Однако если установлен бит смены идентификатора пользователя SUID, то программа получит права доступа к файлам и каталогам, которые имеет владелец файла программы. Это значит, что если у программы установлен бит SUID равный 0, то пользователь, запустивший эту программу, получит права пользователя `root` на время выполнения этой программы.

Самый яркий пример — это смена пароля пользователя. Для того чтобы получить доступ к файлу `/etc/passwd`, к которому есть доступ только у пользователя `root` (который и является владельцем файла), необходимо запустить программу `/usr/bin/passwd`, у которой установлен бит идентификатора пользователя `root`. Запустив эту программу, непривилегированный пользователь получает возможность смены своего пароля в "чужом" файле.

SGID (Set group ID) — бит смены идентификатора группы. Работает аналогичным образом, как и бит смены идентификатора пользователя.

Sticky (липучка) — бит сохранения задачи. Данный бит позволяет указать операционной системе на необходимость сохранения ее (задачи) в оперативной памяти даже после завершения работы программы. Это бывает полезным, когда программа часто загружается для выполнения той или иной задачи, и сохранение кода последней в оперативной памяти позволяет существенно экономить время на ее загрузку. В современных реалиях данный флаг используется для каталогов, чтобы защитить в нем файлы. Из такого каталога пользователь сможет удалить лишь те файлы, владельцем которых он является.

Специальные права, т. е. использование символьных битов, нужны крайне редко. Давайте на конкретном примере посмотрим, как используются символьные биты. Выполните в Терминале команду `ls -l /usr/bin/passwd` (рис. 21.8).

```

Терминал  Файл  Правка  Вид  Поиск  Терминал  Справка
sergey@Test-PC:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 54256 map 29 15:25 /usr/bin/passwd
sergey@Test-PC:~$

```

Рис. 21.8. Просмотр символьных битов

Как мы видим, в первой колонке, так называемом триплете прав (`-rwsr-xr-x`), у прав доступа владельца файла появился непонятный символ `s` вместо символа `x`. Значит, для этой программы установлен бит SUID, и эта программа будет выполнена от имени пользователя `root`, а не от имени того, кто запустил эту программу.

На первый взгляд все несколько сложно, но в то же время и понятно: есть символ `s`, значит, установлен бит SUID. Но как понять, установлен ли символ `x`? Посмотрим на табл. 21.5, и все встанет на свои места.

Таблица 21.5. Соответствие символьных битов с буквенным и цифровым эквивалентами

Символьные биты	Установлен символ <code>x</code>	Не установлен символ <code>x</code>	Цифровой эквивалент
Бит SUID	<code>s</code>	<code>S</code>	4
Бит SGID	<code>s</code>	<code>S</code>	2
Бит Sticky	<code>t</code>	<code>T</code>	1

Как мы видим, если в группе прав, например, владельца файла, установлен флаг *x*, то вместо него будет символ *s*, в противном случае (когда флаг *x* отсутствует) вместо него будет заглавная буква *S*. В цифровом эквиваленте вместо нуля будет цифра 4.

В случае с SGID ситуация аналогичная, за исключением цифрового эквивалента. При установке бита SGID для группы пользователя значение будет равным 2.

При установке sticky-бита символ *x* заменяется строчной буквой *s*, в противном случае будет стоять заглавная буква *S*. Цифровой эквивалент в случае установки бита sticky будет равен 1.

ПРИМЕЧАНИЕ

Стоит помнить, что символьные биты SUID, SGID и sticky задают права для владельца файла, группы пользователя и для всех остальных соответственно.

Таким образом, при правах доступа 4555 (символьное представление *-r-sr-xr-x*) каждый пользователь получает право читать и выполнять файл с правами владельца файла.

Стоит отметить, что при установке SGID для каталога все создаваемые файлы в этом каталоге будут принадлежать к той же группе, что и сам каталог. Установка SUID для каталога не имеет смысла.

Что касается применения sticky-бита к каталогам, то стоит еще раз упомянуть, что удалять или переименовывать файлы в таких каталогах смогут лишь владельцы файлов либо пользователь root.

И напоследок приведем полную таблицу соответствия символьных битов числовым значениям (табл. 21.6).

Таблица 21.6. Соответствие флага символьным битам

Флаг (flag)	Бит SUID	Бит SGID	Бит sticky
0	—	—	—
1	—	—	+
2	—	+	—
3	—	+	+
4	+	—	—
5	+	—	+
6	+	+	—
7	+	+	+

Исходя из этой таблицы, мы сможем правильно задать права доступа к файлу или каталогу, применяя флаг к правам доступа. Вероятнее всего, вам очень редко придется прибегать к использованию данной таблицы, но знать эти основы не помешает.

ПРИМЕЧАНИЕ

Не используйте символьные биты без особой на то надобности, т. к. неправильная их установка может повлечь брешь в системе безопасности рабочей станции. Естественно, что на домашнем компьютере использование символьных битов не принесет большого вреда, о чем нельзя сказать про "боевой" сервер.

Права доступа по умолчанию

Для каталогов максимальные права доступа равны маске 777, а для файлов — 666. Однако ОС Ubuntu Linux по умолчанию создает каталоги и файлы с отличной маской, т. е. для созданного каталога права доступа не будут равны 777, а новым файлам не будут присвоены права 666.

Продemonстрируем это на примере. Для этого последовательно выполните следующие команды:

```
touch test.txt
mkdir testdir
```

В результате в домашнем каталоге вашей учетной записи будут созданы пустой файл `test.txt` и каталог `testdir`, причем с правами по умолчанию. Посмотрим, чему будут равны права доступа, установленные по умолчанию. Выполните следующую команду:

```
ls -l
```

Нас интересуют только две строчки:

```
drwxrwxr-x  2 sergey sergey 4096 окт  5 16:21 testdir
-rw-rw-r--  1 sergey sergey   0 окт  5 16:21 test.txt
```

Теперь, если символьное представление прав доступа перевести в цифровое, то для файла права доступа будут равны 664, а для каталога 775. Почему же получилось именно так? Дело в том, что при создании файла или каталога учитывается пользовательская маска.

Пользовательская маска — это способ задания прав доступа, при котором происходит вычитание заданного числа от числовой маски прав доступа по умолчанию.

В ОС Ubuntu Linux для просмотра и изменения маски прав предназначена команда `umask` (от англ. *user file creation mode mask* — маска режима создания пользовательских файлов). Синтаксис этой команды следующий:

```
umask [маска]
```

Параметр `[маска]` позволяет задать новую маску прав, которая будет учитываться при создании файла или каталога.

При установке ОС "с нуля" маска прав доступа равна 0002 (либо 022 в случае трех-битного обозначения маски). Это легко выяснить, если выполнить команду `umask` без параметров.

Для того чтобы задать новую пользовательскую маску, достаточно выполнить следующую команду:

```
umask 0077
```

В результате создаваемые файлы и каталоги будут рассчитываться с учетом новой пользовательской маски. Однако тут есть одно "но": данная маска будет действовать до первой перезагрузки операционной системы, после чего маска снова будет равна 0002. Для того чтобы не происходило "обнуление" пользовательской маски, добавьте требуемое значение, например `umask 0077`, в файл `.profile`, который находится в домашнем каталоге вашей учетной записи.

Теперь посмотрим, как происходит расчет прав доступа с учетом пользовательской маски.

Для файла расчет маски происходит по следующему правилу. Берется значение прав доступа по умолчанию и отнимается пользовательская маска, т. е. $666 - 002$. В итоге получится число 664. Когда пользовательская маска равна 0022, формула будет выглядеть так: $666 - 0022 = 644$. Теперь понятно, откуда взялось число 664 у только что созданного файла `test.txt`?

Для каталогов расчет аналогичный: $777 - 002 = 775$. Когда пользовательская маска равна 0022, формула будет выглядеть так: $777 - 0022 = 755$.

Это самый легкий и понятный способ расчета прав, но ситуация меняется, когда пользовательская маска равна значению 077. Вот тут уже возникает путаница. Давайте посмотрим на примере.

Итак, нам известно, что пользовательская маска равна 077, значит, формула расчета прав для каталогов будет такой: $777 - 077 = 700$. С этим тоже все понятно, но с файлами ситуация обстоит иначе. Для файлов в нашем случае формула будет выглядеть следующим образом: $666 - 077 = 600$. По правилам математики должно получиться число 589, но почему же в этом примере стоит 600?

Дело в том, что при расчете пользовательской маски и прав доступа вычитание происходит *поразрядно*, т. е. первая цифра прав по умолчанию отнимается первой цифры в маске. Таким образом, получается $6 - 0 = 6$. Затем то же самое продлевается со вторым и третьим числами. И вот тут кроется разгадка: в случае когда результат вычитания получается *отрицательным*, число *заменяется нулем*. Таким образом, $6 - 7$ будет равно 0, а не -1 . Отсюда у нас получилось число 600, а не 589.

Манипуляция пользовательской маской бывает полезной, когда за компьютером работает несколько пользователей, и крайне нежелательно, чтобы они имели доступ к файлам или к папкам, созданным другими пользователями. Установив один раз пользовательскую маску, вам больше не придется каждый раз при создании файла менять его права доступа.

Смена прав

Теперь, когда мы уже знаем о том, что такое права доступа, и понимаем не только как они обозначаются, но и читаются, нам остается лишь научиться их редактировать.

В ОС Ubuntu Linux за редактирование прав доступа отвечает программа `chmod` (от англ. *change mode*), которая расположена в каталоге `/bin/`.

Используется следующий синтаксис программы `chmod`:

```
chmod [ссылки] [оператор] [режимы] файл ...
```

Параметр `[ссылки]` определяет пользователей, к которым будут применяться права доступа, и может принимать значения, перечисленные в табл. 21.7.

Таблица 21.7. Параметр `[ссылки]`

Значение параметра <code>[ссылки]</code>	Класс пользователей	Описание
u	user	Владелец файла
g	group	Группа пользователей файла
o	other	Остальные пользователи
a	all	Все пользователи

Параметр `[оператор]` определяет ту операцию, которая будет передана программе `chmod`, и принимает значения, перечисленные в табл. 21.8.

Таблица 21.8. Параметр `[оператор]`

Параметр <code>[оператор]</code>	Описание
+	Добавить права
-	Удалить права
=	Установить права

Параметр `[режимы]` определяет, какие именно права будут добавлены или удалены, и принимает значения, перечисленные в табл. 21.9.

Таблица 21.9. Параметр `[режимы]`

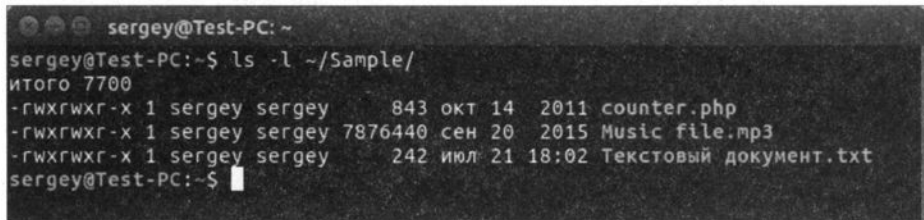
Параметр <code>[режимы]</code>	Название параметра	Описание
r	read	Добавление прав на чтение файла и содержимого каталога
w	write	Добавление прав записи в файл или каталог
x	execute	Добавление прав на выполнение файла или чтение содержимого каталога
X	special execute	Добавление прав на выполнение файла, если он является каталогом или уже имеет право на выполнение
s	setuid/setgid	Добавление атрибутов SUID или SGID, позволяющих запустить файл на выполнение с правами владельца файла (SUID) или группы (SGID)

Таблица 21.9 (окончание)

Параметр [режимы]	Название параметра	Описание
t	sticky	Добавление атрибута t для каталогов, который наделяет правами удаления файлов в этом каталоге только владельца этого файла

Параметр *файл* позволяет указать файл (либо файлы), которого коснутся изменения прав доступа.

В качестве примера у нас будет выступать все тот же каталог Sample, расположенный в домашнем каталоге нашей учетной записи. Откройте Терминал и выполните команду `ls -l ~/Sample/`. В ответ на это программа `ls` вернет примерно следующий результат (рис. 21.9).



```

sergey@Test-PC: ~
sergey@Test-PC:~$ ls -l ~/Sample/
итого 7700
-rwxrwxr-x 1 sergey sergey      843 окт 14   2011 counter.php
-rwxrwxr-x 1 sergey sergey 7876440 сен 20   2015 Music file.mp3
-rwxrwxr-x 1 sergey sergey      242 июл 21 18:02 Текстовый документ.txt
sergey@Test-PC:~$

```

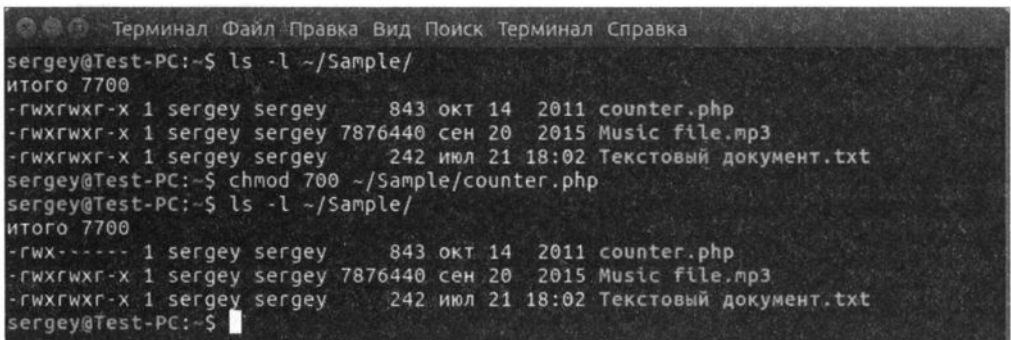
Рис. 21.9. Просмотр прав доступа файлов в каталоге Sample

Допустим, мы по какой-то причине хотим запретить группе пользователя и всем остальным просмотр, запись и выполнение файла `counter.php`, т. е. сменить права с `-rwxrwxr-x` на `-rwx-----`. Для этого намного удобнее воспользоваться числовым представлением прав доступа. Как нам уже известно, представление `-rwx-----` в числовом эквиваленте будет равно 700.

Для изменения прав доступа к файлу `counter.php` выполните команду

```
chmod 700 ~/Sample/counter.php
```

Убедимся, все ли верно мы сделали. Для этого выполним уже известную нам команду `ls -l ~/Sample/` (рис. 21.10).



```

Терминал Файл Правка Вид Поиск Терминал Справка
sergey@Test-PC:~$ ls -l ~/Sample/
итого 7700
-rwxrwxr-x 1 sergey sergey      843 окт 14   2011 counter.php
-rwxrwxr-x 1 sergey sergey 7876440 сен 20   2015 Music file.mp3
-rwxrwxr-x 1 sergey sergey      242 июл 21 18:02 Текстовый документ.txt
sergey@Test-PC:~$ chmod 700 ~/Sample/counter.php
sergey@Test-PC:~$ ls -l ~/Sample/
итого 7700
-rwx----- 1 sergey sergey      843 окт 14   2011 counter.php
-rwxrwxr-x 1 sergey sergey 7876440 сен 20   2015 Music file.mp3
-rwxrwxr-x 1 sergey sergey      242 июл 21 18:02 Текстовый документ.txt
sergey@Test-PC:~$

```

Рис. 21.10. Права доступа к файлу counter.php изменились

Сейчас мы задали права к файлу с помощью цифрового представления прав доступа, но помимо него можно задавать права и с помощью символического (буквенного) представления. Сейчас мы это продемонстрируем.

Давайте вернем группе пользователя права на чтение файла `counter.php`. Для этого выполните в Терминале команду

```
chmod g+r ~/Sample/counter.php
```

Теперь права доступа примут вид `-rwxr-----`.

Возможности программы `chmod` на этом не заканчиваются. Более детальную информацию по работе с этой программой можно получить, выполнив команду `man chmod`.

Смена владельца файла

Иногда возникает ситуация, когда требуется не просто изменить права доступа на файл, но и сменить владельца файла. Для этих целей в ОС Ubuntu Linux предусмотрена программа `chown` (от англ. *change owner*). Синтаксис ее прост:

```
chown пользователь файл
```

где:

- ♦ *пользователь* — это имя пользователя, которому вы собираетесь делегировать права;
- ♦ *файл* — сам файл, права на который вы собираетесь передать другому пользователю.

Имейте в виду, что после смены владельца файла у вас не будет доступа к этому файлу, если только ваша учетная запись не наделена полными административными правами.

Смена атрибутов файла

ОС Ubuntu Linux предоставляет инструмент по работе с атрибутами файлов на "родных" для этой ОС файловых системах. Для смены атрибутов предназначена программа `chattr` (от англ. *change attribute*), синтаксис которой выглядит следующим образом:

```
chattr [-RV] [-v версия] [атрибуты] [файлы...]
```

Необязательный параметр `[-RV]` является ключом POSIX и может принимать значения, перечисленные в табл. 21.10.

Параметр `[атрибуты]` служит для добавления, снятия или установки новых атрибутов и может принимать значения из табл. 21.11.

Для установки перечисленных выше атрибутов имеются специальные операторы (табл. 21.12).

Таблица 21.10. Параметр [-RV] и ключ [-v]

Ключ	Описание
-R	Рекурсивное изменение атрибутов каталога и их содержимого
-V	Вывод полной информации и версии программы <code>chattr</code>
-v	Установка номера версии или генерации файла

Таблица 21.11. Параметр [атрибуты]

Значение	Описание
A	Позволяет не обновлять время последнего доступа к файлу. При этом параметре не происходит обновление записи <code>atime</code>
S	Синхронное обновление файлов. Внесенные изменения синхронно записываются на диск
D	Синхронное обновление каталогов. Внесенные изменения синхронно записываются на диск
a	Разрешает лишь добавлять записи. Установить или снять этот атрибут может только пользователь <code>root</code>
c	Сжатие файла. Файлы с установленным параметром сжатия будут автоматически сжиматься (упаковываться) на диске средствами самой ОС
d	Отмена архивации. Файлы с установленным параметром не будут попадать в список резервного копирования программой <code>dump</code>
i	Параметр немодифицируемого файла. Файлы с установленным параметром становятся полностью немодифицируемыми и не могут быть удалены, переименованы или подвергаться редактированию. Установить или снять этот атрибут может лишь пользователь <code>root</code>
s	Параметр безопасного удаления. При удалении файла с установленным атрибутом будет выполнено обнуление его блоков с обратной их записью на диск
T	Параметр вершины иерархии каталога. Каталог с установленным атрибутом будет считаться расположенным на вершине иерархии каталогов
t	Параметр запрета слияния в конце файла. Файлы с установленным параметром не будут иметь в конце блока дописанных частичных фрагментов других файлов, если такую запись в конце блока поддерживает файловая система
u	Параметр неудаляемого файла. Файлы с таким параметром при удалении физически остаются на диске, т. е. все их содержимое остается на диске не тронутым, что позволяет без труда восстановить такие файлы в дальнейшем

Таблица 21.12. Операторы изменения атрибутов

Оператор	Описание
+	Добавление указанных атрибутов к существующим
-	Снятие (удаление) указанных атрибутов
=	Установка только указанных атрибутов

После перечисления операторов и параметров указывается сам файл/файлы в параметре [файлы...]. В качестве примера давайте для файла counter.php установим параметр неудаляемого файла. Для этого выполните следующую команду:

```
chattr +u ~/Sample/counter.php
```

Она добавит параметр `u` к требуемому файлу.

Просмотр атрибутов

После установки атрибутов файла возникает резонный вопрос: как же просмотреть эти атрибуты?

Для просмотра атрибутов в ОС Ubuntu Linux предусмотрен соответствующий инструмент, который называется `lsattr` (от англ. *list attribute* — просмотреть атрибуты). Это обычная программа, предназначенная для вывода списка установленных атрибутов файла или каталога.

Синтаксис команды `lsattr` выглядит следующим образом:

```
lsattr [-RVadv] [файлы...]
```

Параметр `[-RVadv]` позволяет задать дополнительные опции при запросе атрибута файла. Этот параметр может принимать значения, перечисленные в табл. 21.13.

Таблица 21.13. Параметр `[-RVadv]`

Ключ	Описание
<code>-R</code>	Рекурсивно выводит атрибуты каталогов и их содержимого
<code>-V</code>	Выводит версию программы
<code>-a</code>	Выводит информацию обо всех файлах в каталогах, включая скрытые файлы, чьи имена начинаются с точки (.)
<code>-d</code>	Отображает имена каталогов, также как и имена файлов, вместо отображения содержимого каталогов
<code>-v</code>	Выводит версию или номер поколения файла

Вторым параметром указывается сам файл (файлы), атрибуты которого требуется просмотреть. Если не указывать первый параметр, а передать команде только имя файла, то будут возвращены лишь атрибуты файла.

Теперь, чтобы просмотреть установленные атрибуты у нашего файла counter.php, нужно выполнить команду `lsattr ~/Sample/counter.php`, которая выведет список всех установленных атрибутов файла (рис. 21.11).

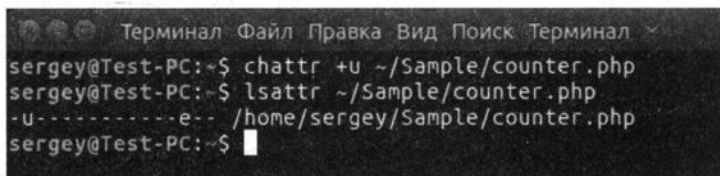


Рис. 21.11. Просмотр атрибутов файла

Как мы можем наблюдать, в первой колонке ответа команды (-u-----e--) появился символ u. Это означает, что файл помечен как неудаляемый.

ПРИМЕЧАНИЕ

Атрибут e говорит о том, что этот файл использует так называемые *экстенты*, т. е. свойство файловой системы ext4, при котором данные файла могут быть расположены в нескольких логических блоках дискового пространства либо храниться в одном таком блоке, равном размеру самого файла.

Выводы

Мы рассмотрели довольно сложную тему, касающуюся групп пользователей и их прав. Мы узнали, что в ОС Ubuntu Linux права могут иметь символьное и цифровое обозначения, а также научились их изменять и добавлять, переназначать владельца файла.

Мы разобрались в том, что собой представляют символьные биты и для чего они предназначены.

ГЛАВА 22



Жесткие и символические ссылки

Ссылки в ОС Linux

В ОС Ubuntu Linux вся информация на жестком диске хранится в виде набора блока данных, которые в зависимости от типа файловой системы могут быть расположены в разных секторах жесткого диска. Отсюда возникает вопрос: где хранится информация о владельце файла, дате создания файла и сведения, если файл в прямом смысле разбросан по всему жесткому диску?

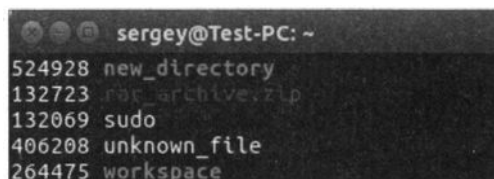
Дело в том, что при создании файловой системы часть ее объема выделяется под индексные дескрипторы — *inode*.

Индексный дескриптор inode (от англ. *index node* — индексный узел, произносится как *айнод* либо *инод*) — это структура данных, присущая UNIX-совместимым файловым системам, в которой хранится метаданные о файле. Так, каждый *inode* имеет свой порядковый номер, который идентифицирует объект в файловой системе и содержит такую информацию, как имя владельца файла, права доступа к файлу, время последнего обращения к файлу, размер файла и т. п. Однако существует определенное число таких дескрипторов, которое соответствует максимальному количеству файлов, допускаемое определенной файловой системой.

Посмотреть номер индексного дескриптора мы можем, выполнив следующую команду:

```
ls -li
```

В результате в первой колонке вывода мы получим тот самый номер индексного дескриптора (рис. 22.1).

A terminal window with a dark background and light text. The prompt is 'sergey@Test-PC: ~'. The output of the 'ls -li' command is shown, with the first column being the inode number and the second column being the file name.

```
sergey@Test-PC: ~  
524928 new_directory  
132723 rat_archive.zip  
132069 sudo  
406208 unknown_file  
264475 workspace
```

Рис. 22.1. Номер индексного дескриптора

Таким образом, у файла `sudo` номер индексного дескриптора равен 132069. Простыми словами, `inode` — это порядковый номер файла, который является уникальным в пределах отдельной файловой системы.

Как мы уже знаем, каталог в ОС Ubuntu Linux представляет собой специальный файл, который содержит записи произвольной длины в зависимости от длины имени файла и количества расположенных в нем файлов, и называется *записью каталога*. Каждая запись каталога содержит номер индексного дескриптора файла (`inode`), размера записи, имени файла и его длину в блоках. Помимо перечисленных данных, в записи каталога могут находиться дополнительные записи. Именно такие записи зачастую и являются ссылками.

Ссылки — это дополнительные записи каталога, которые позволяют обращаться к файлам или каталогам по нескольким именам и из разных каталогов. Такие записи ссылок бывают двух типов:

- ◆ жесткие ссылки;
- ◆ символические ссылки.

Об этих двух типах ссылок мы сейчас и поговорим.

Типы ссылок

Для начала вспомним такое понятие в ОС Windows, как ярлыки. Они предназначены для удобства доступа к файлу или каталогу из любого расположения, будь то рабочий стол или папка с личными документами. И где бы ни был сам ярлык, вы можете сослаться на файл из любого удобного для вас места. По сути, *ярлык* и есть *ссылка*, про которую мы говорили чуть выше. Различие только в том, что в ОС Ubuntu Linux ссылка имеет намного больше функций, чем ярлык в ОС Windows. И кроме того, как мы уже сказали, в ОС Ubuntu Linux имеется два типа ссылок: жесткая и символическая ссылки.

Символическая ссылка (`symlink`) — это запись каталога, указывающая на имя объекта с другим индексным дескриптором `inode` и хранящая только имя файла, а не его содержимое. По своей сути, символическая ссылка очень похожа на простой ярлык в ОС Windows, содержащий путь к файлу, для которого он создан. При открытии символической ссылки открывается целевой файл или каталог. Если удалить символическую ссылку, то сам файл останется нетронутым. Но и при удалении целевого файла символическая ссылка останется нетронутой и будет указывать на несуществующий объект.

ПРИМЕЧАНИЕ

Иногда символические ссылки называют *мягкими ссылками*, однако это не совсем правильно и вносит некоторую путаницу для начинающих пользователей. Поэтому используйте корректное название — символические ссылки.

Символические ссылки могут ссылаться как на файлы, так и на каталоги, при удалении или переименовании которых сами символические ссылки становятся недей-

ствительными. Права доступа и собственно номер индексного дескриптора `inode` будет отличаться от исходного файла. Такое бывает очень полезным, когда вам необходимо поделиться каким-либо файлом с несколькими пользователями и учесть, что права доступа этих пользователей к файлу должны быть разными. При изменении прав доступа самого исходного файла права доступа символической ссылки остаются прежними.

Отличительной особенностью символической ссылки является то, что она может указывать на объекты, которые находятся на другом разделе жесткого диска.

Таким образом, символическая ссылка задействует только базовые возможности файловой системы, что нельзя сказать про жесткие ссылки.

Жесткая ссылка (hard link) — это запись каталога, указывающая на дескриптор `inode` для определенного файла. Таким образом, жесткая ссылка может указывать на файл лишь в пределах одной файловой системы и не может указывать на каталог.

Для жестких ссылок совершенно не имеют значения имя файла и его месторасположение, т. к. она ссылается на дескриптор файла.

Жесткие ссылки имеют те же разрешения на файл, что и сам файл. Жестких ссылок на файл может быть сколько угодно, однако для любого файла существует как минимум одна жесткая ссылка, указывающая на сам файл, т. е. на его индексный дескриптор `inode`. Удалить файл, имеющий жесткие ссылки, возможно только тогда, когда будет удалена последняя жесткая ссылка, и в тот момент, когда удаляется последняя жесткая ссылка, указывающая на дескриптор файла, удаляется сам файл. При этом счетчик ссылок сбрасывается в значение 0.

Создание ссылок

Для создания ссылок в ОС Ubuntu Linux предназначена команда `ln` (от англ. *link* — ссылка). Данная команда позволяет создавать как жесткие, так и символические ссылки. По умолчанию создаются жесткие ссылки. Для создания символической ссылки достаточно указать ключ `--symbolic`.

Синтаксис этой команды следующий:

```
ln [ключ] файл_источник имя_ссылки
```

Параметр `файл_источник` принимает имя файла либо каталога, для которого должна быть создана ссылка. Параметр `имя_ссылки` задает имя для создаваемой ссылки.

Рассмотрим основные значения, которые принимает параметр `[ключ]` (табл. 22.1).

Мы рассмотрели необходимые параметры команды `ln`, и теперь наступило время поближе познакомиться со ссылками.

Таблица 22.1. Значения параметра [ключ] команды `ln`

Сокращенный вариант	Полный вариант	Описание
	<code>--backup[=контроль]</code>	<p>Позволяет создать резервную копию для каждого целевого файла в соответствии с заданным аргументом <code>[=контроль]</code>. Допустимые значения этого аргумента следующие:</p> <ul style="list-style-type: none"> • <code>none, off</code> — отключение создания резервных копий; • <code>numbered, t</code> — создание нумерованных копий; • <code>existing, nil</code> — при существовании нумерованных резервных копий будет продолжено создание нумерованных копий. В противном случае будут созданы простые копии; • <code>simple, never</code> — создание простых копий
<code>-b</code>		Аналогичен ключу <code>--backup</code> , только не позволяет принимать аргумент
<code>-d, -F</code>	<code>--directory</code>	Позволяет пользователю <code>root</code> создавать жесткие ссылки на каталоги. При этом с большей долей вероятности действие завершится ошибкой из-за системных ограничений
<code>-f</code>	<code>--force</code>	Позволяет удалять существующие целевые файлы
<code>-i</code>	<code>--interactive</code>	Позволяет запросить подтверждение перед удалением ссылки
<code>-L</code>	<code>--logical</code>	Позволяет разыменовывать символическую ссылку, т. е. получить объект по ссылке
<code>-n</code>	<code>--no-dereference</code>	Позволяет рассматривать <code>имя_ссылки</code> как обычный файл, даже если это символическая ссылка на каталог
<code>-P</code>	<code>--physical</code>	Позволяет создать жесткую ссылку непосредственно на саму символическую ссылку
<code>-r</code>	<code>--relative</code>	Позволяет создать символическую ссылку относительно ее расположения
<code>-s</code>	<code>--symbolic</code>	Позволяет создать символическую ссылку вместо жесткой ссылки
<code>-S</code>	<code>--suffix=суффикс</code>	Позволяет переопределить обычный суффикс резервных копий на тот, что задан в аргументе <code>суффикс</code> . По умолчанию суффикс резервных копий — это <code>~</code>
<code>-t</code>	<code>--target-directory=каталог</code>	Позволяет указать <code>каталог</code> , в котором должны быть созданы указанные ссылки
<code>-T</code>	<code>--no-target-directory</code>	Позволяет всегда считать <code>имя_ссылки</code> обычным файлом

Работа с символическими ссылками

Для начала рассмотрим создание символических ссылок, но перед этим для удобства в домашнем каталоге создадим каталог `links` и сделаем его текущим:

```
mkdir links
cd links
```

Создадим файл `sample_symlink` с произвольным текстом:

```
echo 'Произвольный текст' > sample_symlink
```

Данной командой мы создали файл `~/links/sample_symlink` с текстом "Произвольный текст". Остается только создать символическую ссылку на этот файл:

```
ln -s sample_symlink my_symlinks
```

После выполнения данной команды в каталоге `~/links/` будет создана символическая ссылка `my_symlinks` для файла `sample_symlink` (рис. 22.2).

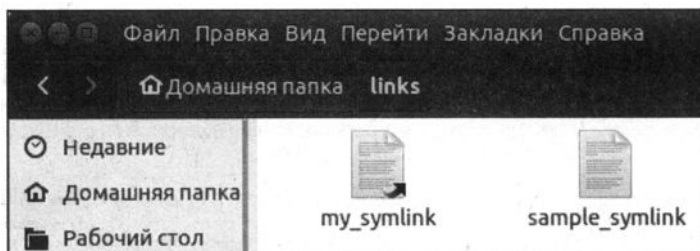


Рис. 22.2. Символическая ссылка `my_symlink`

Убедимся в том, что по символической ссылке доступен просмотр содержимого файла `sample_symlink`:

```
cat my_symlink
```

В Терминале отобразится содержимое файла `sample_symlink`. Если у вас все еще остались сомнения в том, действительно ли перед нами символическая ссылка, выполните следующую команду:

```
ls -li
```

Ключ `-i` в данной команде позволяет вывести `inode` (индексный дескриптор файла) для каждого файла в каталоге. В результате выполнения данной команды вы увидите следующее (рис. 22.3).

```
sergey@Test-PC: ~/links
sergey@Test-PC:~/links$ ls -li
итого 4
265161 lrwxrwxrwx 1 sergey sergey 14 map 13 13:49 my_symlink -> sample_symlink
265160 -rw-rw-r-- 1 sergey sergey 36 map 13 13:49 sample_symlink
sergey@Test-PC:~/links$
```

Рис. 22.3. Просмотр содержимого каталога `~/links/`

Нас интересует третья строка вывода Терминала. В первом столбце вывода число 265161 означает номер дескриптора файла, тот самый inode. Это и есть символическая ссылка. Обратите внимание, что индексный дескриптор ссылки и самого файла отличается на одну лишь цифру, а это уже означает, что перед нами два файла.

Во втором столбце вывода информации о символической ссылке мы видим уже знакомые права доступа пользователей, но первым символом является буква l, которая как раз и указывает на то, что перед нами ссылка. Кроме этого, права доступа символической ссылки и самого файла различаются.

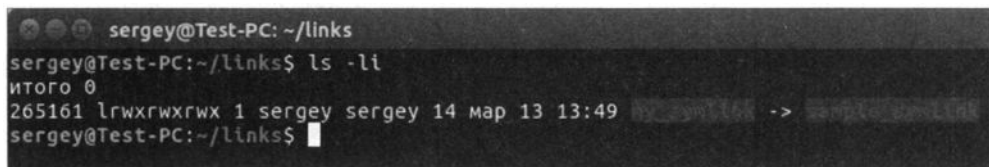
Третьим столбцом значится счетчик жестких ссылок на данный файл. В нашем случае у данного файла только одна жесткая ссылка, которая указывает на сам файл. Счетчик ссылок справедлив только для жестких ссылок и не распространяется на символические ссылки.

Последним столбцом следует строка "my_symlink -> sample_symlink", которая сообщает, что ссылка my_symlink указывает на файл sample_symlink. Указателем в данном случае выступают символы ->.

Попробуем удалить исходный файл sample_symlink и вывести информацию о содержимом каталога ~/links/.

```
rm sample_symlink
ls -li
```

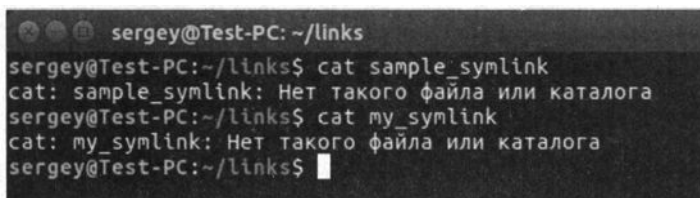
После выполнения этих двух команд мы увидим следующую картину (рис. 22.4).



```
sergey@Test-PC: ~/links
sergey@Test-PC:~/links$ ls -li
итого 0
265161 lrwxrwxrwx 1 sergey sergey 14 мар 13 13:49 my_symlink -> sample_symlink
sergey@Test-PC:~/links$
```

Рис. 22.4. Ссылка указывает на несуществующий файл

Обратите внимание, что Терминал выделил красным цветом имя символической ссылки, словно подсказывая нам, что она указывает на несуществующий файл. Если теперь попытаться выполнить команду просмотра содержимого файла или ссылки, то Терминал выведет ошибку о том, что запрашиваемый файл не существует (рис. 22.5).



```
sergey@Test-PC: ~/links
sergey@Test-PC:~/links$ cat sample_symlink
cat: sample_symlink: Нет такого файла или каталога
sergey@Test-PC:~/links$ cat my_symlink
cat: my_symlink: Нет такого файла или каталога
sergey@Test-PC:~/links$
```

Рис. 22.5. Запрашиваемого файла sample_symlink не существует

Таким образом, мы убедились, что символическая ссылка не содержит информацию из файла, на который она указывает, и при удалении исходного файла ссылка будет указывать на несуществующий файл. Стоит отметить, что при удалении символической ссылки исходный файл остается нетронутым, ровно, как и его содержимое.

Нерабочую символическую ссылку можно удалить за ненадобностью:

```
rm my_symlink
```

Работа с жесткими ссылками

Теперь продемонстрируем работу с жесткими ссылками. Для начала создадим файл `sample_hardlink`, в который поместим некоторый текст:

```
cd links
echo 'Произвольный текст' > sample_hardlink
```

Теперь создадим саму жесткую ссылку:

```
ln sample_hardlink my_hardlink
```

Для создания жесткой ссылки мы передали команде `ln` имя файла источника (`sample_hardlink`) и вторым параметром указали имя создаваемой жесткой ссылки.

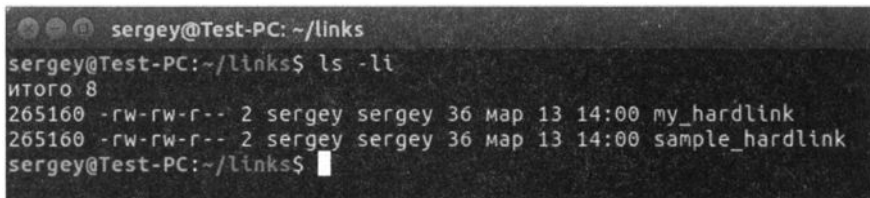
Теперь, если просмотреть содержимое файла по жесткой ссылке, будет выведено содержимое файла `sample_hardlink`:

```
cat my_hardlink
```

Пока ничего необычного в работе жесткой ссылки мы не увидели, и на первый взгляд кажется, что жесткая ссылка ничем не отличается от символической ссылки. Однако это не так. Давайте выполним следующую команду:

```
ls -li
```

Обратите внимание на индексный дескриптор файла (рис. 22.6).



```
sergey@Test-PC: ~/links
sergey@Test-PC:~/links$ ls -li
итого 8
265160 -rw-rw-r-- 2 sergey sergey 36 мар 13 14:00 my_hardlink
265160 -rw-rw-r-- 2 sergey sergey 36 мар 13 14:00 sample_hardlink
sergey@Test-PC:~/links$
```

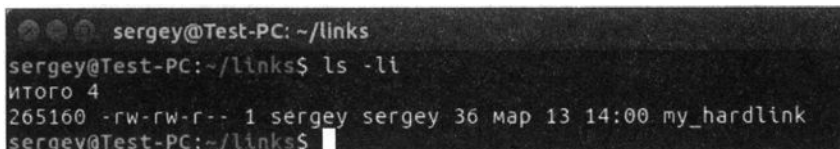
Рис. 22.6. Жесткая ссылка на файл `sample_hardlink`

Как мы видим, у файла `sample_hardlink` и жесткой ссылки `my_hardlink` совпадают индексные дескрипторы файла. Права доступа также одинаковые. В третьей колонке значится цифра 2, которая говорит нам о том, что для данного файла существует две жесткие ссылки: `sample_hardlink` и `my_hardlink`.

Что будет, если удалить исходный файл? Давайте на примере и узнаем:

```
rm sample_hardlink
cat my_hardlink
```

Выполнив эти две команды, мы убедимся в том, что даже после удаления исходного файла содержимое файла `sample_hardlink` остается доступным для чтения. Для этого просмотрим индексный дескриптор жесткой ссылки `my_hardlink` (рис. 22.7).



```
sergey@Test-PC: ~/links
sergey@Test-PC:~/links$ ls -li
итого 4
265160 -rw-rw-r-- 1 sergey sergey 36 map 13 14:00 my_hardlink
sergey@Test-PC:~/links$
```

Рис. 22.7. Жесткая ссылка указывает на тот же inode

Обратите внимание, что inode остался прежним, а счетчик ссылок стал равен 1. Теперь мы сами лично убедились в том, что при удалении жесткой ссылки оставшаяся ссылка по-прежнему указывает на нужный участок диска с данными. Только после удаления последней ссылки удалится сам файл, а счетчик ссылок будет обнулен.

Выводы

Мы познакомились со ссылками в ОС Ubuntu Linux и узнали о том, что они бывают двух типов: символическими и жесткими. Мы также узнали, что каждый файл имеет свой уникальный индексный дескриптор — inode, который не может повторяться в пределах одной файловой системы.

Мы научились создавать символические ссылки, а также жесткие ссылки. После этого на конкретных примерах мы рассмотрели, как работать с ними и отличать их типы.

ГЛАВА 23



Монтирование файловых систем

Точка монтирования

Когда мы только начинали знакомиться с файловой системой ОС Ubuntu Linux, мы узнали, что при создании все разделы монтируются в подкаталоги корня файловой системы. Корневой каталог, как известно, начинается с символа прямого слеша /. То есть, в отличие от ОС Windows, все разделы монтируются только как подкаталоги, а не как отдельные устройства, например, диски C:, D: и т. д. Таким образом, каждый монтируемый каталог или устройство имеет так называемую точку монтирования.

Точка монтирования (mount point) — это каталог в ОС Ubuntu Linux, принадлежащий корневой файловой системе, к которому подключают либо от которого отключают разделы диска во время работы операционной системы.

ОС Ubuntu Linux для удобства пользователя автоматически монтирует подключаемые устройства. Это означает, что вам не нужно самостоятельно выполнять команду монтирования подключенного устройства. Однако автомонтирование доступно не во всех Linux-подобных операционных системах. Так, в серверных версиях операционных систем автомонтирование считается небезопасным и выполняется только системным администратором. Правда, это вовсе не означает, что вы теперь не сможете воспользоваться возможностями автомонтирования подключаемых устройств. Вопрос только в том, целесообразно и безопасно ли автомонтирование подключаемых устройств на рабочем сервере?

Совсем по-иному дела обстоят с домашними компьютерами, для которых автомонтирование устройств просто необходимо, поскольку за день мы подключаем несколько устройств и по несколько раз. Именно по этой причине в ОС Ubuntu решено было включить автомонтирование по умолчанию.

Монтирование устройств в ОС Ubuntu

В ОС Ubuntu монтирование разделов диска осуществляют исходя из настроек файла /etc/fstab, в котором хранятся настройки монтирования разделов жесткого диска, включая файл подкачки.

При каждой загрузке операционная система считывает настройки из этого файла и на основании полученной информации монтирует все перечисленные разделы в указанные точки монтирования (рис. 23.1).

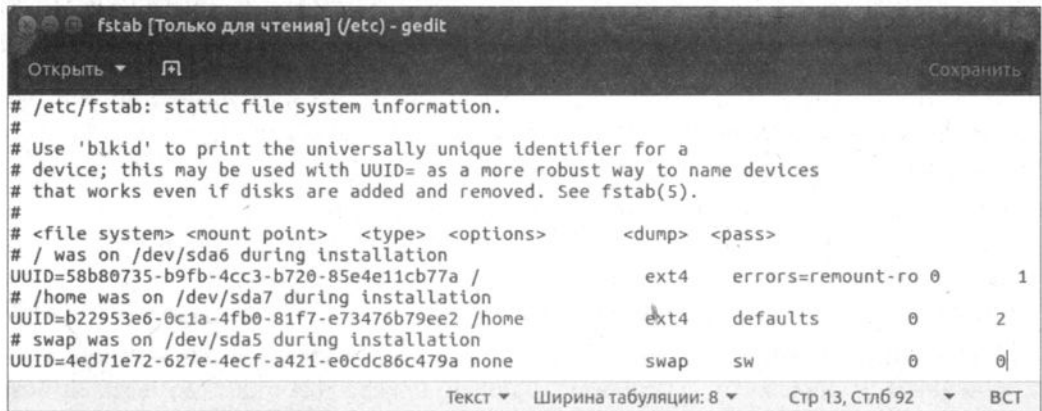


Рис. 23.1. Точки монтирования в файле /etc/fstab

Кроме этого файла, операционная система не имеет никаких других файлов, в которых бы задавались иные точки монтирования.

Все подключаемые USB-устройства хранения информации, как правило, автоматически монтируются в каталог /media/ с именем, соответствующим метке устройства. Операционная система Ubuntu сама определяет тип файловой системы устройства и самостоятельно выделяет точку монтирования для подключенного устройства. Все это справедливо для устройств, которые подключены в то время, когда операционная система уже загружена. Однако в случае подключения устройства до момента загрузки операционной системы устройство будет смонтировано только при первом обращении к нему.

Возникает резонный вопрос: как быть в том случае, когда требуется монтировать устройство при загрузке операционной системы? Здесь нам на помощь приходит уже знакомый файл /etc/fstab, в котором нужно правильно прописать настройки монтирования, и устройство будет автоматически монтироваться при каждой загрузке ОС, но об этом мы поговорим чуть позже, а сейчас рассмотрим способы наименования блочных устройств.

Способы постоянных именований

Существует несколько способов именования блочных устройств:

- ◆ наименование посредством UUID;
- ◆ наименование посредством имен устройств;
- ◆ наименование посредством меток устройств (labels);
- ◆ наименование посредством Hardware ID.

Любое монтируемое устройство всегда уникально, даже в том случае, если вы найдете два одинаковых устройства с одними и теми же характеристиками. Например, вы купили два одинаковых USB-накопителя одного и того же производителя, которые даже внешне выглядят одинаково, но это только на первый взгляд. Каждое устройство имеет свой уникальный идентификатор UUID.

UUID (Universally Unique Identifier — универсальный уникальный идентификатор) — это уникальный и постоянный буквенно-числовой идентификатор, предназначенный для идентификации подключаемого оборудования, а точнее, раздела с файловой системой.

Присвоение уникального идентификатора скрыто от наших глаз и происходит на уровне ядра операционной системы. Так, уникальный идентификатор может быть присвоен даже разделу файла подкачки `swap`. Если переустановить операционную систему, то раздел получит другой UUID.

Наименование блочных устройств посредством UUID является наиболее распространенным и надежным способом. Данный способ работает во всех Linux-подобных дистрибутивах.

Вторым, более легким способом является наименование устройств посредством *имен устройств*. Как вы помните, в ОС Ubuntu Linux имена устройств имеют буквенное обозначение, например, первый жесткий диск имеет имя `sda`, а второй жесткий диск получит имя `sdb` и т. д. При этом первый раздел на первом жестком диске получит имя `sda1`, а второй — `sda2`. Если вы захотите поменять местами первый и второй жесткие диски, то изменятся также имена устройств. Данный способ является устаревшим и не рекомендуется к применению.

Третий способ наименования по своей сути напоминает наименование посредством имен устройств, только на этот раз вместо имен устройств используются метки устройств (*labels*).

Метка устройства (*labels*) — это короткое и удобное для запоминания слово, которое обычно задается разделу диска при его создании. Например, для раздела `/dev/sda5`, на котором у нас расположен раздел файла подкачки, мы можем задать метку `swap`, а для раздела `/dev/sda6` — метку `root`. Задание меток никак не повлияет на работу операционной системы — это нужно только для вашего удобства.

Данный метод наименования удобен своей простотой и заменяет устаревший способ наименования посредством имен устройств.

Последний метод наименования — *наименование посредством Hardware ID*. Данный метод на первый взгляд напоминает наименование посредством UUID, но это только на первый взгляд. Дело в том, что, как и в первом случае, при наименовании устройств задействуются цифробуквенные значения. Эти цифробуквенные значения называются *аппаратным серийным номером*, и они задаются заводом-производителем. Данные значения остаются неизменными даже при переустановке операционной системы, и это дает некоторое удобство. Так, настроив однажды монтирование такого устройства, вы сможете сохранить данные настройки для резервного восстановления, например, для быстрого восстановления настроек после краха или переустановки операционной системы.

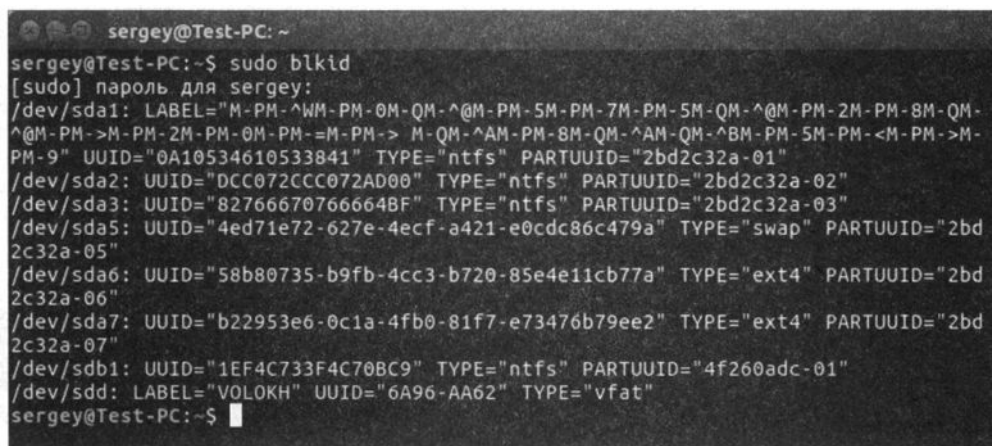
Теперь перейдем к более детальному рассмотрению способов именований блочных устройств.

Именованное посредством UUID

Так как универсальный уникальный идентификатор диска зависит от конкретной файловой системы, то для начала необходимо получить этот атрибут:

```
sudo blkid
```

После выполнения данной команды будет получен примерно следующий результат (рис. 23.2).



```
sergey@Test-PC: ~
sergey@Test-PC:~$ sudo blkid
[sudo] пароль для sergey:
/dev/sda1: LABEL="M-PM-^WM-PM-0M-QM-^@M-PM-5M-PM-7M-PM-5M-QM-^@M-PM-2M-PM-8M-QM-^@M-PM->M-PM-2M-PM-0M-PM-=M-PM-> M-QM-^AM-PM-8M-QM-^AM-QM-^BM-PM-5M-PM-<M-PM->M-PM-9" UUID="0A10534610533841" TYPE="ntfs" PARTUUID="2bd2c32a-01"
/dev/sda2: UUID="DCC072CCC072AD00" TYPE="ntfs" PARTUUID="2bd2c32a-02"
/dev/sda3: UUID="827666707666648F" TYPE="ntfs" PARTUUID="2bd2c32a-03"
/dev/sda5: UUID="4ed71e72-627e-4ecf-a421-e0cdc86c479a" TYPE="swap" PARTUUID="2bd2c32a-05"
/dev/sda6: UUID="58b80735-b9fb-4cc3-b720-85e4e11cb77a" TYPE="ext4" PARTUUID="2bd2c32a-06"
/dev/sda7: UUID="b22953e6-0c1a-4fb0-81f7-e73476b79ee2" TYPE="ext4" PARTUUID="2bd2c32a-07"
/dev/sdb1: UUID="1EF4C733F4C70BC9" TYPE="ntfs" PARTUUID="4f260adc-01"
/dev/sdd: LABEL="VOLOKH" UUID="6A96-AA62" TYPE="vfat"
sergey@Test-PC:~$
```

Рис. 23.2. Получение UUID подключенных устройств

Для примера рассмотрим следующую строку:

```
/dev/sda6: UUID="58b80735-b9fb-4cc3-b720-85e4e11cb77a" TYPE="ext4"
PARTUUID="2bd2c32a-06"
```

Данная строка состоит из четырех колонок:

- ♦ /dev/sda6: — имя раздела;
- ♦ UUID="58b80735-b9fb-4cc3-b720-85e4e11cb77a" — уникальный идентификатор UUID;
- ♦ TYPE="ext4" — тип раздела, т. е. его файловая система;
- ♦ PARTUUID="2bd2c32a-06" — уникальный идентификатор раздела в пределах диска.

Если вы сейчас обратите внимание на последнюю строку, то увидите следующую колонку:

```
LABEL="VOLOKH"
```

Она и является той самой меткой (label), которую мы рассматривали в третьем способе наименования блочных устройств.

Здесь нас больше всего интересует вторая колонка, которая показывает нам уникальный идентификатор UUID. Разумеется, данный раздел не требует монтирова-

ния, т. к. он уже монтируется при загрузке операционной системы, и поэтому данная строка была показана только в качестве примера.

Именованное посредством имен устройств

Стоит напомнить, что данный способ именования является устаревшим и не рекомендуется к использованию. Однако о нем стоит знать, т. к. к нему нередко прибегают при восстановлении сбоев на машинах с устаревшей Linux-подобной операционной системой.

Итак, чтобы получить имена устройств, точнее, имена разделов, выполним следующую команду:

```
sudo fdisk -u -l
```

Данная команда выведет подробное описание всех блочных разделов памяти, которые задействованы в данный момент. Вывод может показаться вам слишком большим, поэтому давайте сократим его, уточнив, какой именно диск нас интересует:

```
sudo fdisk -u -l /dev/sda
```

В данном случае мы указываем команде `fdisk`, что нас интересует устройство `/dev/sda`. Результат выполнения команды будет не таким длинным (рис. 23.3).

```

sergey@Test-PC: ~
sergey@Test-PC:~$ sudo fdisk -u -l /dev/sda
[sudo] пароль для sergey:
Диск /dev/sda: 298,1 GiB, 320072933376 байтов, 625142448 секторов
Единицы измерения: секторов из 1 * 512 = 512 байтов
Размер сектора (логический/физический): 512 байт / 512 байт
I/O size (minimum/optimal): 512 bytes / 512 bytes
Тип метки диска: dos
Идентификатор диска: 0x2bd2c32a

Устр-во   Загрузочный   Start  Конеч  Секторы   Size Id Тип
/dev/sda1 *      2048    206847    204800    100M  7 HPFS/NTFS/exFAT
/dev/sda2        206848  163842047  163635200    78G  7 HPFS/NTFS/exFAT
/dev/sda3      163842048  471042047  307200000  146,5G  7 HPFS/NTFS/exFAT
/dev/sda4      471044094  625141759  154097666    73,5G  5 Расширенный
/dev/sda5      471044096  480260095    9216000    4,4G  82 Linux swap / Sola
/dev/sda6      480262144  582662534  102400391    48,8G  83 Linux
/dev/sda7      582664192  625141759  42477568    20,3G  83 Linux
sergey@Test-PC:~$
  
```

Рис. 23.3. Вывод информации о наименованиях разделов жесткого диска

Вывод команды `fdisk` состоит из восьми колонок, но нас интересует только первая колонка **Устр-во**, которая перечисляет имена разделов, присутствующих в операционной системе. При таком способе монтировании используются имена устройств, которые перечислены в первой колонке вывода команды.

Именованное посредством меток устройств

Современные файловые системы позволяют задавать метку (label) раздела. Однако это не обязательно, и метку задают в основном для удобства.

```

sergey@Test-PC: ~
sergey@Test-PC:~$ sudo blkid
[sudo] пароль для sergey:
/dev/sda1: LABEL="М-РМ-^WM-PM-0M-QM-^@M-PM-5M-PM-7M-PM-5M-QM-^@M-PM-2M-PM-8M-QM-^@M-PM->M-PM-2M-PM-0M-PM-=M-PM-> M-QM-^AM-PM-8M-QM-^AM-QM-^BM-PM-5M-PM-<M-PM->M-PM-9" UUID="0A10534610533841" TYPE="ntfs" PARTUUID="2bd2c32a-01"
/dev/sda2: UUID="DCC072CCCC072AD00" TYPE="ntfs" PARTUUID="2bd2c32a-02"
/dev/sda3: UUID="82766670766664BF" TYPE="ntfs" PARTUUID="2bd2c32a-03"
/dev/sda5: UUID="4ed71e72-627e-4ecf-a421-e0cdc86c479a" TYPE="swap" PARTUUID="2bd2c32a-05"
/dev/sda6: UUID="58b80735-b9fb-4cc3-b720-85e4e11cb77a" TYPE="ext4" PARTUUID="2bd2c32a-06"
/dev/sda7: UUID="b22953e6-0c1a-4fb0-81f7-e73476b79ee2" TYPE="ext4" PARTUUID="2bd2c32a-07"
/dev/sdb1: UUID="1EF4C733F4C70BC9" TYPE="ntfs" PARTUUID="4f260adc-01"
/dev/sdd: LABEL="VOLOKH" UUID="6A96-AA62" TYPE="vfat"
sergey@Test-PC:~$

```

Рис. 23.4. Вывод информации о подключенных устройствах

Выполните знакомую нам команду `sudo blkid` и внимательно посмотрите результат вывода (рис. 23.4).

Обратите внимание на первую строку вывода и на последнюю. В обеих строках присутствует параметр `LABEL`, который и содержит заданную метку. Так, в последней строке у нас задана метка `VOLOKH`, а в первой строке какие-то непонятные цифробуквенные значения. Дело в том, что данная метка задана на русском языке ОС Windows и содержит значение "Зарезервировано системой", т. е. метка задана в другой кодировке, и программа `blkid` не смогла ее корректно обработать.

Если вы все еще сомневаетесь, запустите редактор разделов GParted и убедитесь в этом лично (рис. 23.5).

ПРИМЕЧАНИЕ

В только что установленной операционной системе Ubuntu редактор разделов GParted отсутствует, и его требуется установить вручную, выполнив следующую команду: `sudo apt install gparted`.

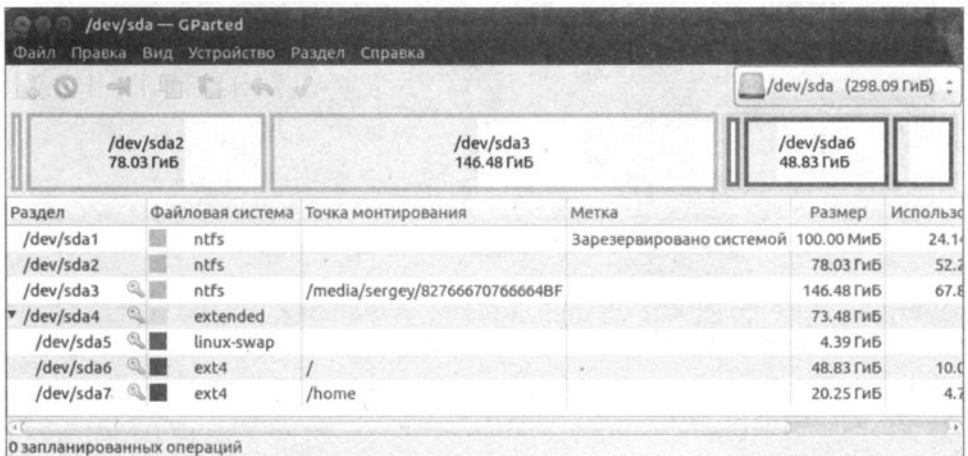


Рис. 23.5. Редактор разделов GParted

В столбце **Метка** перечислены все заданные метки. Стоит иметь в виду, что подключенные устройства отображаются по отдельности. Для переключения отображения другого диска выберите нужный диск в раскрывающемся списке на панели инструментов.

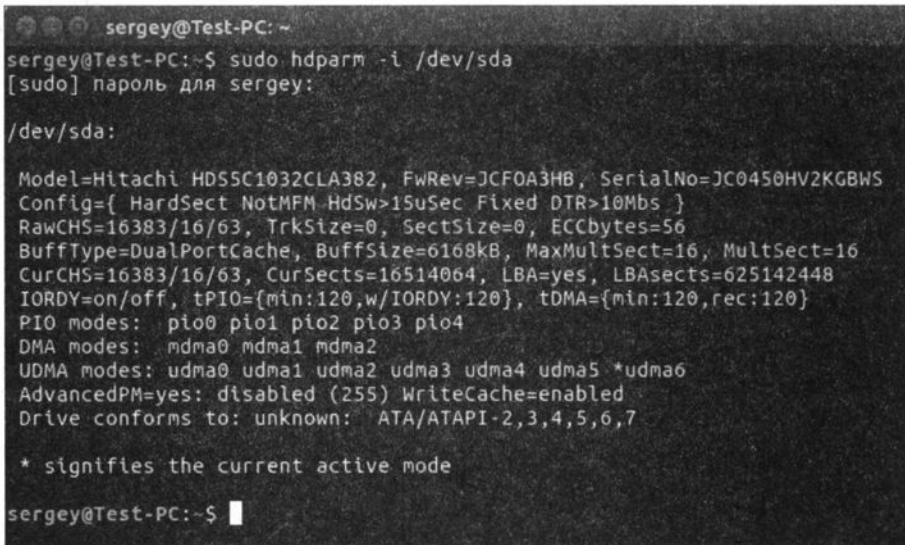
Использование меток при монтировании очень помогает в работе, однако также имеет свои недостатки. Так, вы уже убедились в том, что метки, заданные на русском языке в одной операционной системе, не всегда корректно обрабатываются в другой системе. Метки могут быть повторяющимися, и в виду этого будет возникать путаница при монтировании устройств.

Именованное посредством Hardware ID

Каждому устройству еще на заводе присваивается уникальный аппаратный серийный номер, который остается неизменным на протяжении всей работы этого устройства. Узнать серийный номер устройства, например диска `sda`, можно следующей командой:

```
sudo hdparm -i /dev/sda
```

По выполнении команды вы получите не только серийный номер, но и модель устройства, а также другую техническую информацию (рис. 23.6).



```
sergey@Test-PC: ~
sergey@Test-PC:~$ sudo hdparm -i /dev/sda
[sudo] пароль для sergey:

/dev/sda:

Model=Hitachi HDS5C1032CLA382, FwRev=JCFOA3HB, SerialNo=JC0450HV2KGBWS
Config={ HardSect NotMFM HdSw>15uSec Fixed DTR>10Mbs }
RawCHS=16383/16/63, TrkSize=0, SectSize=0, ECCbytes=56
BuffType=DualPortCache, BuffSize=6168kB, MaxMultSect=16, MultSect=16
CurCHS=16383/16/63, CurSects=16514064, LBA=yes, LBAsects=625142448
IORDY=on/off, tPIO={min:120,w/IORDY:120}, tDMA={min:120,rec:120}
PIO modes: pio0 pio1 pio2 pio3 pio4
DMA modes: mdma0 mdma1 mdma2
UDMA modes: udma0 udma1 udma2 udma3 udma4 udma5 *udma6
AdvancedPM=yes: disabled (255) WriteCache=enabled
Drive conforms to: unknown: ATA/ATAPI-2,3,4,5,6,7

* signifies the current active mode

sergey@Test-PC:~$
```

Рис. 23.6. Получение информации о жестком диске

Параметр `SerialNo` содержит серийный номер устройства, однако серийный номер принадлежит диску в целом.

А теперь посмотрим, как операционная система "видит" разделы диска:

```
ls -l /dev/disk/by-id/
```

По выполнении данной команды мы увидим примерно такую картину (рис. 23.7).

```
sergey@Test-PC: ~  
sergey@Test-PC:~$ ls -l /dev/disk/by-id/  
итого 0  
lrwxrwxrwx 1 root root 9 map 24 09:20 ata-Hitachi_HDS5C1032CLA382_JC0450HV2KGBWS -> ../../sda  
lrwxrwxrwx 1 root root 10 map 24 09:20 ata-Hitachi_HDS5C1032CLA382_JC0450HV2KGBWS-part1 -> ../../sda1  
lrwxrwxrwx 1 root root 10 map 24 09:20 ata-Hitachi_HDS5C1032CLA382_JC0450HV2KGBWS-part2 -> ../../sda2  
lrwxrwxrwx 1 root root 10 map 24 09:20 ata-Hitachi_HDS5C1032CLA382_JC0450HV2KGBWS-part3 -> ../../sda3  
lrwxrwxrwx 1 root root 10 map 24 09:20 ata-Hitachi_HDS5C1032CLA382_JC0450HV2KGBWS-part4 -> ../../sda4  
lrwxrwxrwx 1 root root 10 map 24 09:20 ata-Hitachi_HDS5C1032CLA382_JC0450HV2KGBWS-part5 -> ../../sda5  
lrwxrwxrwx 1 root root 10 map 24 09:20 ata-Hitachi_HDS5C1032CLA382_JC0450HV2KGBWS-part6 -> ../../sda6  
lrwxrwxrwx 1 root root 10 map 24 09:20 ata-Hitachi_HDS5C1032CLA382_JC0450HV2KGBWS-part7 -> ../../sda7
```

Рис. 23.7. Получение информации по разделам жесткого диска

Обратите внимание, что для каждого созданного раздела операционная система сгенерировала идентификационный номер на основе серийного номера устройства, интерфейса подключения, наименования производителя и номера раздела. Так, вместо простого серийного номера JC0450HV2KGBWS, диск получил длинный идентификационный номер ata-Hitachi_HDS5C1032CLA382_JC0450HV2KGBWS. Помимо этого, каждый раздел получил суффикс -part1, -part2 и т. д.

Универсальный метод получения информации о блочных устройствах

Мы рассмотрели четыре способа именования блочных устройств, и каждый метод был описан в отдельности. Есть еще один универсальный способ получения необходимой информации с помощью команды:

```
ls -lR /dev/disk
```

Данная команда позволит вывести информацию по всем доступным блочным устройствам. Результаты выполнения данной команды практически ничем не отличаются от результатов выполнения тех команд, которые мы рассматривали при описании каждого метода именования в отдельности. Вы вправе пользоваться любой удобной и понравившейся вам командой. Знание о том, как получить детальную информацию о блочном устройстве, никогда не бывает лишним и может выручить вас в нужный момент.

Автоматическое монтирование

Теперь, когда мы уже знаем о способах именования блочных устройств, настало время познакомиться с автоматическим монтированием этих устройств. За автоматическое монтирование устройств отвечает файл /etc/fstab. Для его редактирования нужны права root, поэтому запускаем его редактирование с соответствующими правами:

```
gksudo gedit /etc/fstab
```

Файл /etc/fstab небольшой, и в нем перечисляются диски или разделы, которые должны быть смонтированы во время загрузки операционной системы.

Строки, начинающиеся со знака решетки #, являются комментариями и игнорируются операционной системой.

На каждый раздел или диск отводится одна строчка, которая состоит из шести колонок:

- ◆ `<file system>` — наименование раздела или диска;
- ◆ `<mount point>` — точка монтирования;
- ◆ `<type>` — тип файловой системы;
- ◆ `<options>` — дополнительные параметры монтирования;
- ◆ `<dump>` — указание утилите `dump` для создания резервных копий данных файловой системы;
- ◆ `<pass>` — указание утилите `fsck` для проверки целостности файловой системы.

Первый столбец `<file system>` указывает на физическое место размещения файловой системы, по которому операционная система будет искать конкретный раздел или диск для монтирования. В этом столбце мы можем указать UUID, метку устройства, путь или серийный номер.

Второй столбец `<mount point>` задает точку монтирования, т. е. каталог, к которому будет примонтирован требуемый раздел или диск.

Третий столбец `<type>` принимает в качестве аргумента тип поддерживаемой файловой системы, например, `ext2`, `ext3`, `ext4`, `btrfs`, `reiserfs`, `xfs`, `jfs`, `smbfs`, `iso9660`, `vfat`, `ntfs`, `swap` или `auto`. Если в качестве аргумента указано значение `auto`, то операционная система попытается самостоятельно определить тип файловой системы. Данная опция весьма полезна при автоматическом монтировании компакт-дисков.

Четвертый столбец `<options>` задает параметры монтирования файловой системы. В табл. 23.1 перечислены основные принимаемые параметры данного столбца.

Таблица 23.1. Параметры столбца `<options>`

Значение	Описание
<code>auto</code>	Позволяет файловой системе монтироваться автоматически либо после выполнения в Терминале команды <code>mount -a</code>
<code>noauto</code>	Позволяет монтировать файловую систему не автоматически, а по запросу пользователя
<code>defaults</code>	Позволяет монтировать файловую систему со значениями по умолчанию, которые соответствуют набору опций: <code>auto</code> , <code>async</code> , <code>exec</code> , <code>dev</code> , <code>rw</code> , <code>suid</code>
<code>dev</code>	Позволяет на монтируемой файловой системе создать файлы устройств
<code>nodev</code>	Позволяет запретить создание файлов устройств на монтируемой файловой системе
<code>exec</code>	Позволяет исполнять бинарные файлы, расположенные на монтируемом разделе или диске
<code>noexec</code>	Позволяет запретить исполнение бинарных файлов. Допустимо использовать на несистемных разделах. В противном случае, эта настройка приведет к неработоспособности ОС

Таблица 23.1 (окончание)

Значение	Описание
suid	Позволяет записать в формате файла устройства Linux в каталоге /dev/, например /dev/sda1, или с помощью уникального идентификатора UUID, тогда формат записи будет таким UUID="XXXX-XXXX-XXXX-XXXX", также возможна запись с помощью метки, например, LABEL=home. Файловая система указывает, в какой файловой системе нужно монтировать это устройство, например, ext4, ext3, ext2, btrfs. Точка монтирования — куда нужно примонтировать это устройство, например, /home, /boot, /mnt. Параметр позволяет выполнять операции с suid - и sgid -битами
nosuid	Позволяет запретить выполнение операций с SUID- и SGID-битами
sync	Позволяет задать синхронное выполнение операций ввода-вывода
async	Позволяет асинхронно выполнять операции ввода-вывода
user	Разрешает монтирование файловой системы любому пользователю
nouser	Разрешает монтирование файловой системы только пользователю root
ro	Позволяет монтировать файловую систему только для чтения
rw	Позволяет смонтировать файловую систему для чтения и записи
atime	Позволяет включить информацию о последнем времени доступа при каждом чтении файла
noatime	Позволяет отключить запись информации о последнем времени доступа к файлу

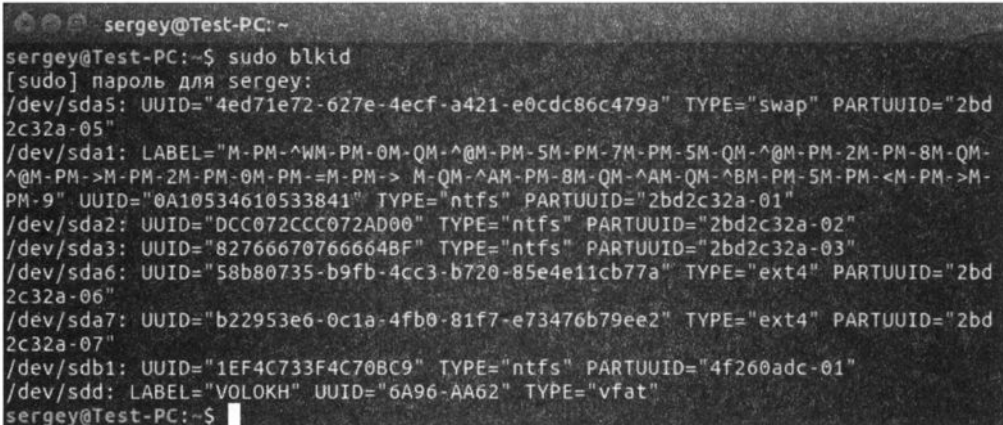
В табл. 23.1 перечислены лишь основные параметры монтирования, которые может принимать столбец `<options>`. Со всеми возможными опциями монтирования вы можете ознакомиться, выполнив команду `man mount`.

Пятый столбец `<dump>` предназначен для одноименной утилиты, которая позволяет делать резервную копию. Возможные значения — 0 или 1. Если установлено значение 1, то утилита `dump` делает резервную копию монтируемой файловой системы. Утилита `dump` по умолчанию не устанавливается вместе с операционной системой, и поэтому во избежание возможных ошибок рекомендуется установить значение 0.

Последний, шестой столбец `<pass>` предназначен утилите `fsck`, которая считывает заданное в столбце число и на основании этого решает, в каком порядке ей следует проверять монтируемую файловую систему. Возможные значения: 0, 1 либо 2. При установке значения 0 файловая система не проверяется утилитой `fsck`. Значение 1 присваивается только корневой файловой системе, остальные файловые системы могут иметь приоритет, равный значению 2.

Теперь, когда мы ознакомились с форматом файла `/etc/fstab`, настало время применить полученные знания на практике. Давайте для начала научимся автоматическому монтированию USB-устройства. Для этого подключите устройство к компьютеру и выполните следующую команду:

```
sudo blkid
```



```

sergey@Test-PC: ~
sergey@Test-PC:~$ sudo blkid
[sudo] пароль для sergey:
/dev/sda5: UUID="4ed71e72-627e-4ecf-a421-e0cdc86c479a" TYPE="swap" PARTUUID="2bd2c32a-05"
/dev/sda1: LABEL="М-PM-^WM-PM-0M-QM-^@M-PM-5M-PM-7M-PM-5M-QM-^@M-PM-2M-PM-8M-QM-^@M-PM->M-PM-2M-PM-0M-PM-=M-PM-> M-QM-^AM-PM-8M-QM-^AM-QM-^BM-PM-5M-PM-<M-PM->M-PM-9" UUID="0A10534610533841" TYPE="ntfs" PARTUUID="2bd2c32a-01"
/dev/sda2: UUID="DCC072CCC072AD00" TYPE="ntfs" PARTUUID="2bd2c32a-02"
/dev/sda3: UUID="827666707666648F" TYPE="ntfs" PARTUUID="2bd2c32a-03"
/dev/sda6: UUID="58b80735-b9fb-4cc3-b720-85e4e11cb77a" TYPE="ext4" PARTUUID="2bd2c32a-06"
/dev/sda7: UUID="b22953e6-0c1a-4fb0-81f7-e73476b79ee2" TYPE="ext4" PARTUUID="2bd2c32a-07"
/dev/sdb1: UUID="1EF4C733F4C708C9" TYPE="ntfs" PARTUUID="4f260adc-01"
/dev/sdd: LABEL="VOLOKH" UUID="6A96-AA62" TYPE="vfat"
sergey@Test-PC:~$

```

Рис. 23.8. Вывод списка подключенных накопителей

Программа `blkid` покажет все подключенные блочные устройства вместе с их UUID. Последним в списке будет наш USB-накопитель (рис. 23.8).

В нашем случае, подключенный накопитель имеет имя `/dev/sdd`. Создадим для этого устройства каталог, к которому будет монтироваться файловая система:

```
sudo mkdir /mnt/volokh/
```

Каталог для монтирования создан, осталось только составить запись для файла `/etc/fstab`:

```
UUID=6A96-AA62 /mnt/volokh/ auto defaults 0 2
```

Разберем нашу запись:

- ◆ `6A96-AA62` — UUID диска, который будет монтироваться при загрузке системы;
- ◆ `/mnt/volokh/` — каталог, к которому будет монтироваться файловая система;
- ◆ `auto` — тип файловой системы. В данном случае, файловая система будет распознана автоматически;
- ◆ `defaults` — дополнительные опции. В данном случае, будет использоваться стандартный набор опций, а именно: `auto, async, exec, dev, rw, suid`;
- ◆ `0` — указание утилите `dump` о том, что для данной файловой системы не нужно делать резервных копий;
- ◆ `2` — указание утилите `fsck` о том, что файловая система будет проверяться на наличие ошибок.

Теперь осталось внести изменения в файл `/etc/fstab`. Для этого запустим текстовый редактор с правами `root` в графическом режиме:

```
gksudo gedit /etc/fstab
```

В конец файла `/etc/fstab` дописываем нашу строку:

```
UUID=6A96-AA62 /mnt/volokh/ auto defaults 0 2
```

Сохраняем сделанные нами изменения и перезагружаем операционную систему. После загрузки операционной системы в каталог `/mnt/volokh/` будет смонтирована файловая система съемного USB-носителя. Это означает, что на панели запуска теперь будет отсутствовать значок нашего USB-носителя и все файлы будут располагаться в каталоге `/mnt/volokh/`.

ПРИМЕЧАНИЕ

Монтирование файловой системы в определенный каталог не означает, что все данные физически находятся в монтируемом каталоге, т. к. этот каталог является всего лишь точкой монтирования.

Подобным образом вы можете монтировать и логический раздел жесткого диска:

```
UUID=82766670766664BF /mnt/data/ ntfs defaults 0 2
```

ПРИМЕЧАНИЕ

Помните, что раздел для монтирования должен существовать, иначе операционной системе не удастся выполнить монтирование файловой системы.

Для монтирования устройства *по его имени* формат записи будет следующим:

```
/dev/sda3 /mnt/data/ ntfs defaults 0 2
```

Монтирование с использованием *метки устройства* (label) осуществляется так:

```
LABEL=MYDATA /mnt/data/ ntfs defaults 0 2
```

Несколько иначе выполняется автоматическое монтирование файловой системы *по серийному номеру* (Hardware ID). Для начала вам следует выяснить серийный номер раздела, который вы желаете монтировать:

```
ls -l /dev/disk/by-id/
```

В результате выполнения команды мы получим список разделов и их серийные номера (см. рис. 23.7).

В этом выводе списка разделов нас интересует раздел `/dev/sda3`. Полученные номера разделов — это файлы, расположенные в каталоге `/dev/disk/by-id/`. Поэтому для указания требуемого раздела нам необходимо ввести полный путь к файлу раздела, т. е. задать на сам раздел:

```
/dev/disk/by-id/ata-Hitachi_HDS5C1032CLA382_JC0450HV2KGBWS-part3 /mnt/data/  
ntfs defaults 0 2
```

Как видите, нет ничего сложного в автоматическом монтировании устройств.

С автоматическим монтированием мы разобрались, и настало время познакомиться с ручным способом монтирования файловых систем.

Ручное монтирование и размонтирование устройства

Помимо автоматического монтирования файловых систем, нам может понадобиться ручное монтирование, например в ситуации, когда необходимо временно смонтировать USB-накопитель в указанный вами каталог.

Предположим, вы уже знаете имя ваше устройства, пусть это будет `/dev/sdd`, и данное устройство вам необходимо смонтировать в каталог `/mnt/mydoc/`. Для этого выполните следующую команду:

```
sudo mount /dev/sdd /mnt/mydoc/
```

ПРИМЕЧАНИЕ

Помните, что точка монтирования должна существовать, иначе вы получите следующую ошибку: `"mount: mount point /mnt/mydoc/ does not exist"`.

После успешной процедуры монтирования все файлы, расположенные на монтируемом устройстве, будут доступны из каталога `/mnt/mydoc/`.

Для размонтирования (отсоединения) устройства предназначена следующая конструкция:

```
sudo umount /dev/sdd
```

Данная команда позволит размонтировать указанное устройство. Стоит отметить, что вместо указания имени устройства вы можете также задать монтируемый каталог. Так, следующая команда полностью аналогична предыдущей:

```
sudo umount /mnt/mydoc/
```

Если вы обратите внимание, то как при монтировании, так и размонтировании устройства, на панели запуска остается значок устройства. Дело в том, что размонтирование устройства вовсе не означает извлечение устройства. Для извлечения устройства выполните следующую команду:

```
sudo eject /dev/sdd/
```

После этого ваше USB-устройство будет отключено, и вы сможете безопасно его извлечь, не боясь потерять данные.

Выводы

Мы рассмотрели вопрос монтирования дисковых устройств, а также принцип автоматического монтирования устройств, познакомились с методами наименования устройств, научились составлять инструкции для автоматического монтирования данных устройств. Мы обсудили вопрос ручного монтирования файловых систем и их размонтирования.

Полученные нами знания окажутся полезными при повседневной работе с операционной системой Ubuntu Linux.

ГЛАВА 24



Уровни выполнения и цели в ОС Ubuntu

Инициализация системы

В ОС Ubuntu, как и в другой Linux-подобной операционной системе, инициализация системы начинается с загрузки программы *init*. От того, как сконфигурирована эта программа, зависит дальнейшая загрузка операционной системы. Однако обо всем по порядку.

Между нажатием кнопки включения компьютера и полной загрузкой операционной системы проходит не более минуты, но весь рутинный процесс загрузки скрыт от наших глаз. Рассмотрим этот процесс в упрощенном варианте.

Итак, после нажатия кнопки включения компьютера управление на себя берет *BIOS* (Basic Input/Output System — базовая система ввода-вывода), которая производит проверки целостности устройств и, если не найдено ошибок, загружает главную загрузочную запись *MBR* (Master Boot Record). Затем управление передается программе-загрузчику ОС. В нашем случае загрузчиком выступает *GRUB* (GRand Unified Bootloader). Он загружает ядро операционной системы, и управление передается уже ядру. После этого ядро монтирует файловую систему и загружает в память программу *init*. А она, в свою очередь, определяет уровень выполнения и в соответствии с ним загружает операционную систему.

Это всего лишь упрощенное описание процедуры запуска операционной системы, но именно в таком порядке и происходит загрузка ОС.

Уровни выполнения

Уровень выполнения (*runlevel*) — это режим работы операционной системы, который определяет, какие задачи могут выполняться операционной системой в текущем состоянии.

Обычно встречается 7 уровней выполнения, которые нумеруются с нуля и обозначаются числами от 0 и до 6. Работа с операционной системой в тот или иной момент времени означает ее работу на соответствующем уровне выполнения. Так, по умолчанию ОС Ubuntu работает на пятом уровне выполнения (рис. 24.1).

```

sergey@Test-PC: ~
sergey@Test-PC:~$ runlevel
N 5
sergey@Test-PC:~$

```

Рис. 24.1. ОС Ubuntu находится на пятом уровне выполнения

При отключенном графическом режиме ОС Ubuntu будет загружена со значением со вторым уровнем выполнения. В табл. 24.1 приведены все уровни выполнения.

Таблица 24.1. Уровни выполнения в ОС Ubuntu

Уровень выполнения	Описание
0	Позволяет выполнить действия по выключению операционной системы
1	Однопользовательский режим, предназначенный для административных целей по восстановлению работоспособности операционной системы
2	Многопользовательский режим, повторяющий собой третий уровень запуска. На этом режиме обычно работает ОС Ubuntu Server
3	Многопользовательский режим с поддержкой сети
4	Многопользовательский режим. Обычно не используется
5	Многопользовательский режим с поддержкой графического режима
6	Позволяет выполнить действия по перезагрузке операционной системы

Помимо перечисленных семи уровней выполнения, существует еще один уровень, обозначаемый буквой *s* (от англ. *Single*). Данный уровень является синонимом однопользовательского режима, т. е. уровня 1.

Уровень инициализации, т. е. уровень запуска по умолчанию, в ОС Ubuntu указывается в файле `/etc/init/rc-sysinit.conf` (рис. 24.2).

Значение по умолчанию указано в строке `"env DEFAULT_RUNLEVEL=2"`. Данный уровень выполнения срабатывает тогда, когда графическая среда отсутствует. Соот-

```

rc-sysinit.conf [Только для чтения] (/etc/init) - gedit
Открыть Сохранить
stop on runlevel

# Default runlevel, this may be overridden on the kernel command-line
# or by faking an old /etc/inittab entry
env DEFAULT_RUNLEVEL=2

emits runlevel

# There can be no previous runlevel here, but there might be old
Текст Ширина табуляции: 8 Стр 14, Стлб 1 ВСТ

```

Рис. 24.2. Уровень выполнения по умолчанию

ветственно, т. к. мы пользуемся обычной (не серверной) версией ОС Ubuntu, то по умолчанию для этой версии ОС используется пятый уровень выполнения.

После того как в память загрузилась программа `init`, она начинает считывать параметры файла `/etc/init/rc-sysinit.conf`, и уже на основании этих параметров происходит определение уровня запуска.

В зависимости от того, какой уровень запуска выбран, операционная система выполняет сценарии в соответствии со следующим списком (табл. 24.2).

Таблица 24.2. Соответствие уровня выполнения и сценария выполнения

Уровень выполнения	Каталог со сценарием
0	<code>/etc/rc0.d/</code>
1	<code>/etc/rc1.d/</code>
2	<code>/etc/rc2.d/</code>
3	<code>/etc/rc3.d/</code>
4	<code>/etc/rc4.d/</code>
5	<code>/etc/rc5.d/</code>
6	<code>/etc/rc6.d/</code>
S	<code>/etc/rcS.d/</code>

В перечисленных каталогах расположены сценарии, которые отвечают за обработку соответствующего им уровня выполнения. Названия сценариев начинаются с буквы `s` либо с буквы `k`. Сценарии, начинающиеся с буквы `s`, предназначены для запуска (от англ. *startup*), а сценарии, начинающиеся с буквы `k`, — для завершения работы (от англ. *kill*).

После буквенной приставки идут число и название сервиса. Число обозначает порядковый номер сервиса, который должен быть запущен при выполнении этого уровня инициализации.

Тот или иной сценарий, расположенный в перечисленных каталогах, запускает на выполнение сервис, сценарий которого находится в каталоге `/etc/init.d/`. Так, на пятом уровне при выполнении сценария `/etc/rc5.d/S04cups` запустится сценарий `/etc/init.d/cups`.

Современная система уровня запуска

В ранних версиях ОС Ubuntu Linux смена уровня запуска осуществлялась следующей командой:

```
sudo init 3
```

В текущей версии ОС Ubuntu данная команда не работает в виду того, что система инициализации `init` (точнее, система System V `init`) была заменена `upstart`, и после в ОС Ubuntu версии 15.10 появилась новая система инициализации `systemd`.

ПРИМЕЧАНИЕ

Команды `runlevel` и `telinit` оставлены в ОС по соображениям совместимости написанных ранее сценариев для прежней системы инициализации. Вместо этих команд теперь используется команда `systemctl`.

`systemd` — это современный менеджер инициализации и централизованного управления службами операционной системы.

В настоящее время, менеджер инициализации `systemd` установлен по умолчанию в большинстве Linux-подобных дистрибутивов. Он призван заменить собой устаревшие системы инициализации, такие как System V `init` и `upstart`.

Вместо уровней выполнения используются так называемые цели.

Цели (targets) — это альтернатива уровням выполнения, которая вместо номера выполнения использует свое уникальное имя.

Важным отличием *целей* от *уровней выполнения* является то, что в одно и то же время могут быть активными несколько целей, а уровень выполнения мог быть только один. Наглядно различия и сходства концепции уровней выполнения и целей представлены в табл. 24.3.

Таблица 24.3. Уровни выполнения и цели в ОС Ubuntu

Уровень выполнения	Цели	Описание
0	<code>runlevel0.target</code> , <code>poweroff.target</code>	Уровень позволяет выполнить действия по выключению операционной системы
1, S	<code>runlevel1.target</code> , <code>rescue.target</code>	Однопользовательский режим, предназначенный для административных целей по восстановлению работоспособности операционной системы
2	<code>runlevel2.target</code> , <code>multi-user.target</code>	Многопользовательский режим, повторяющий собой третий уровень запуска. На этом режиме обычно работает ОС Ubuntu Server
3	<code>runlevel3.target</code> , <code>multi-user.target</code>	Многопользовательский режим с поддержкой сети
4	<code>runlevel4.target</code> , <code>multi-user.target</code>	Многопользовательский режим. Обычно не используется
5	<code>runlevel5.target</code> , <code>graphical.target</code>	Многопользовательский режим с поддержкой графического режима
6	<code>runlevel6.target</code> , <code>reboot.target</code>	Уровень позволяет выполнить действия по перезагрузке операционной системы
emergency	<code>emergency.target</code>	Уровень позволяет запустить аварийную оболочку операционной системы

Как видите, концепция целей внешне схожа с концепцией уровней выполнения, однако вместо цифр используются имена. Конечно же, от прежней системы инициализации в `systemd` осталась только несколько схожая концепция, но не более

того. Система инициализации `systemd` строилась с учетом максимального быстрого действия, базовыми компонентами которой являются юниты.

Юниты (от англ. *unit* — единица) — это файлы конфигурации, хранящие информацию о службе, устройстве или сожете.

Каждая такая единица, т.е. *юнит*, для своей работы может требовать другие единицы (*Requires*), конфликтовать с другими единицами (*Conflicts*), запускаться ранее других юнитов (*Before*) или после других (*After*) (рис. 24.3).

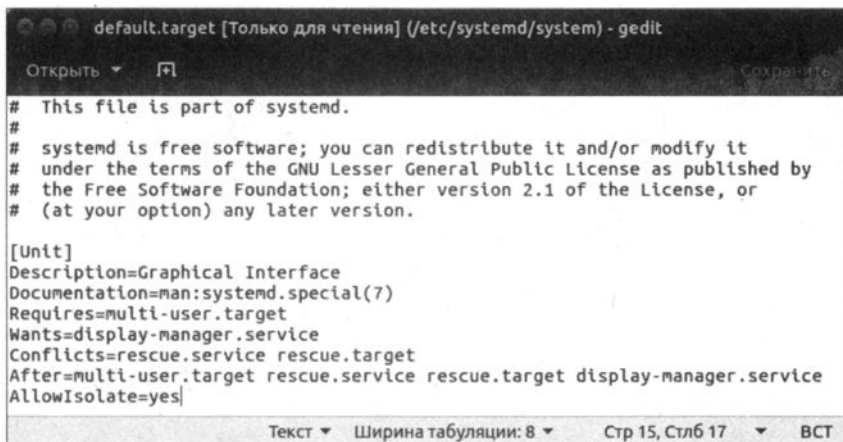


Рис. 24.3. Содержимое цели `/etc/systemd/system/default.target`

Единицы (юниты) хранятся в каталоге `/lib/systemd/system/`. Выяснить расположение юнитов можно следующей командой:

```
pkg-config systemd --variable=systemdsystemunitdir
```

Изменить текущую цель можно следующим образом:

```
sudo systemctl isolate reboot.target
```

Данная команда позволит перезапустить операционную систему. Команда `systemctl` предназначена для управления целями и призвана заменить собой устаревшие команды `runlevel` и `telinit`. Эта команда является некой управляющей надстройкой над `systemd`.

Давайте посмотрим, как можно определить цель, которая загружается по умолчанию:

```
systemctl get-default
```

Данная команда возвращает цель, которая запускается по умолчанию (рис. 24.4).

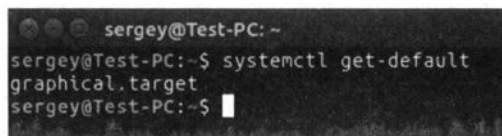


Рис. 24.4. Цель, запускаемая по умолчанию

Для запуска цели, соответствующей уровням выполнения 2, 3 и 4, потребуется создание *именованного целевого юнита*, который обычно создается на основе существующих юнитов и размещается в каталоге `/etc/systemd/`. В каталог `/etc/systemd/ваша_цель_wants` следует включить те службы, которые должны быть запущены в требуемой цели.

Выводы

В этой главе мы вкратце познакомились с уровнями выполнения и целями в ОС Ubuntu Linux. Мы узнали, что уровни выполнения были заменены целями.

Цели в ОС Ubuntu могут быть полезны для проведения работ по восстановлению работоспособности операционной системы либо для диагностики проблем в загрузке ОС.

ГЛАВА 25



Службы в ОС Ubuntu

Службы, сервисы и демоны

В современных многозадачных операционных системах существуют фоновые программы, которые не требуют взаимодействия с пользователем. Такие программы называют службами.

Служба (от англ. *service*) — программа, запускаемая операционной системой и работающая в фоновом режиме без прямого взаимодействия с пользователем.

В UNIX-мире такого понятия, как служба, нет, а вместо него есть *демоны* (от англ. *daemon*). Название службы "*демон*" носит отнюдь не религиозный характер. Одной из версий такого названия является акронимом выражения "**D**isk **A**nd **E**xecution **M**ONitor".

В большинстве Linux-подобных операционных систем службы именуют именно демонами, однако в ОС Ubuntu вместо демонов "прижилось" название "службы".

ПРИМЕЧАНИЕ

Другим, менее распространенным названием служб, является название "сервисы", которое является лишь дословным переводом английского слова *service*.

Сейчас мы не будем вдаваться в подробности происхождения термина "демон", т. к. это подобно спорам на тему: "что было вначале — курица или яйцо?", а лучше подробнее рассмотрим службы в ОС Ubuntu.

Службы в ОС Ubuntu

Для того чтобы централизованно управлять службами операционной системы, нужна система инициализации, которая будет запускать службы в нужный момент и следить за тем, чтобы они работали так, как это было задумано по определению.

В первых версиях ОС Ubuntu Linux управление запуском служб производилось с помощью команд Терминала, в котором требовалось выполнить команду запуска нужной службы. По мере развития концепции ОС Ubuntu Linux на смену устаревшей модели запуска служб пришла система инициализации *systemd*.

Система инициализации *systemd* — это программа инициализации служб операционной системы Linux. Другими словами, *systemd* является первичной службой, которая инициализируется ядром операционной системы и после этого запускает подчиненные ей службы операционной системы.

После запуска той или иной службы система инициализации *systemd* собирает весь вывод служебной информации запущенной службы и сохраняет в определенный лог-файл. В случае если служба аварийно завершила свою работу, система инициализации *systemd* будет пытаться ее перезапустить.

Как нам известно из предыдущей главы, в системе инициализации *systemd* есть такое понятие, как юниты. Повторим, что *юниты* (от англ. *unit* — единица) — это файлы конфигурации, хранящие информацию о службе, устройстве или сокете.

Юниты могут в своей работе взаимодействовать друг с другом. Приведем небольшой список юнитов, которые задействуются в работе службой *systemd*:

- ♦ *service* — программа, которая управляет работой других служб;
- ♦ *socket* — сокет для взаимодействия процессов (определенных службой);
- ♦ *device* — конфигурационный файл, который содержит инструкции для обработки списка устройств;
- ♦ *mount* — файл, содержащий информацию о точке монтирования файловой системы;
- ♦ *automount* — файл, содержащий информацию о списке устройств, которые должны быть смонтированы автоматически;
- ♦ *target* — файл, указывающий на группу служб, которые требуется выполнить этой единице;
- ♦ *snapshot* — файл, указывающий на другие единицы и восстанавливающий состояние ранее запущенных служб.

Мы рассмотрели только основные единицы, которые используются в работе системой инициализации *systemd*. Описывать каждую единицу в отдельности не имеет смысла, мы рассмотрим только *unit service*. Он как раз относится к службам в ОС Ubuntu Linux.

Управление службами в ОС Ubuntu

Для управления службами система инициализации *systemd* использует инструмент *systemctl*.

Systemctl — инструмент системы инициализации *systemd*, предназначенный для управления службами в ОС Ubuntu Linux. Данный инструмент не только разрешает управлять запуском, остановкой и перезагрузкой служб, но и имеет возможность проверки ее состояния, а также позволяет анализировать эффективность загрузки службы.

Синтаксис программы *systemctl* следующий:

```
systemctl [опции] {команда}
```

Вкратце рассмотрим основные значения, которые принимает параметр [опции] команды `systemctl` (табл. 25.1).

Таблица 25.1. Описание опций команды `systemctl`

Сокращенный вариант	Полный вариант	Описание
-a	--all	Позволяет отобразить все запущенные службы, включая остановленные
-q	--quiet	Позволяет вывести минимальную информацию по службам
-t	--type=тип	Позволяет отобразить службы в соответствии с заданным типом
-l	--full	Позволяет отобразить полное имя единицы
	--runtime	Позволяет не сохранять вносимые изменения после перезагрузки системы
	--no-pager	Позволяет отключить постраничный вывод информации
	--state=состояние	Позволяет отсортировать выводимый список служб по заданному состоянию: <ul style="list-style-type: none"> • LOAD — отображение только загруженных служб; • SUB — отображение состояния активности единицы; • ACTIVE — отображение активных служб
-s	--signal=сигнал	Позволяет отправить службе заданный сигнал

ПРИМЕЧАНИЕ

Это лишь небольшой список опций команды `systemctl`. Полный набор параметров можно узнать, выполнив команду `man systemctl`.

Теперь рассмотрим значения, которые принимает параметр {команда} команды `systemctl` (табл. 25.2).

Таблица 25.2. Описание параметров {команда} программы `systemctl`

Команда	Описание
<code>list-units</code>	Позволяет вывести список активных юнитов
<code>list-sockets</code>	Позволяет вывести список загруженных сокетов, упорядоченных по адресу
<code>start NAME</code>	Позволяет запустить службу, имя которой передано в аргументе <i>NAME</i>
<code>stop NAME</code>	Команда позволяет остановить службу, имя которой передано в аргументе <i>NAME</i>

Таблица 25.2 (окончание)

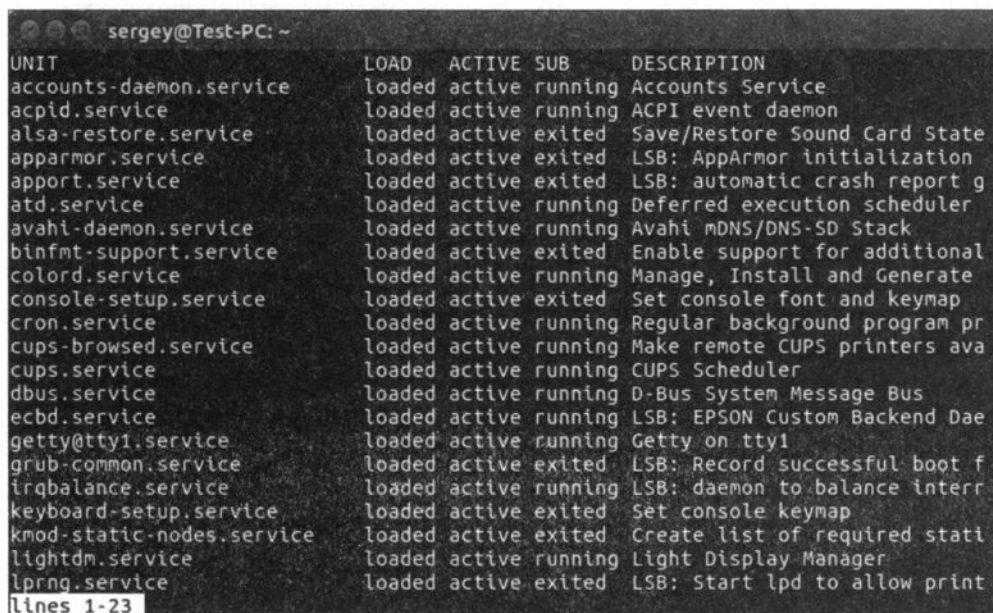
Команда	Описание
<code>reload NAME</code>	Позволяет обновить конфигурации службы, имя которой передано в аргументе <i>NAME</i>
<code>restart NAME</code>	Позволяет перезапустить службу, имя которой передано в аргументе <i>NAME</i>
<code>try-restart NAME</code>	Позволяет перезапустить службу, имя которой передано в аргументе <i>NAME</i> , при условии, что указанная служба активна в данный момент времени
<code>reload-or-restart NAME</code>	Позволяет обновить конфигурацию службы, а затем выполнить ее перезапуск. В случае если обновление конфигурации не предусмотрено, выполняется только перезапуск службы
<code>isolate NAME</code>	Позволяет запустить службу, имя которой передано в аргументе <i>NAME</i> , а все остальные остановить
<code>kill NAME</code>	Позволяет отправить службе сигнал завершения работы, имя которой передано в аргументе <i>NAME</i>
<code>is-active PATTERN</code>	Позволяет проверить, запущена ли указанная служба, имя которой передано в аргументе <i>PATTERN</i>
<code>is-failed PATTERN</code>	Позволяет проверить, не завершилась ли аварийно служба, имя которой передано в аргументе <i>PATTERN</i>
<code>status [PATTERN PID]</code>	Позволяет отобразить состояние службы, имя которой передано в аргументе <i>PATTERN</i> либо идентификатор которой задан в аргументе <i>PID</i> , и ее вывод
<code>show [PATTERN JOB]</code>	Позволяет отобразить параметры управления службой, имя которой передано в аргументе <i>PATTERN</i> , либо отобразить параметры работающих служб
<code>reset-failed [PATTERN]</code>	Команда позволяет перезапустить службы, завершившиеся с ошибкой, имя которых передано в аргументе <i>PATTERN</i>
<code>list-dependencies [NAME]</code>	Позволяет отобразить зависимости службы, имя которой передано в аргументе <i>NAME</i>
<code>list-unit-files [PATTERN]</code>	Позволяет отобразить все файлы, установленные указанной службой
<code>enable NAME</code>	Позволяет активировать запуск службы, имя которой передано в аргументе <i>NAME</i>
<code>disable NAME</code>	Позволяет отключить запуск службы, имя которой передано в аргументе <i>NAME</i>
<code>reenable NAME</code>	Позволяет отключить запуск службы, а затем снова включить
<code>is-enabled NAME</code>	Позволяет отобразить, находится ли указанная служба в автозапуске или нет

Таковы основные команды программы `systemctl`, полный набор которых вы можете узнать, выполнив команду справки `man systemctl`.

Теперь, когда мы познакомились с основными командами системы инициализации, настало время применить полученные знания на практике. Для начала отобразим список запущенных служб:

```
systemctl list-units --type=service
```

Данная конструкция позволит вывести список служб в удобном для восприятия виде (рис. 25.1).



```
sergey@Test-PC: ~
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
accounts-daemon.service            loaded active running Accounts Service
acpid.service                       loaded active running ACPI event daemon
alsa-restore.service               loaded active exited Save/Restore Sound Card State
apparmor.service                   loaded active exited LSB: AppArmor initialization
apport.service                     loaded active exited LSB: automatic crash report g
atd.service                        loaded active running Deferred execution scheduler
avahi-daemon.service               loaded active running Avahi mDNS/DNS-SD Stack
binfmt-support.service             loaded active exited Enable support for additional
colord.service                     loaded active running Manage, Install and Generate
console-setup.service              loaded active exited Set console font and keymap
cron.service                       loaded active running Regular background program pr
cups-browsed.service                loaded active running Make remote CUPS printers ava
cups.service                       loaded active running CUPS Scheduler
dbus.service                       loaded active running D-Bus System Message Bus
ecbd.service                      loaded active running LSB: EPSON Custom Backend Dae
getty@tty1.service                 loaded active running Getty on tty1
grub-common.service                loaded active exited LSB: Record successful boot f
irqbalance.service                 loaded active running LSB: daemon to balance interr
keyboard-setup.service              loaded active exited Set console keymap
kmod-static-nodes.service           loaded active exited Create list of required stati
lightdm.service                    loaded active running Light Display Manager
lprng.service                      loaded active exited LSB: Start lpd to allow print
lines 1-23
```

Рис. 25.1. Список запущенных служб

В полученном результате имеется 5 колонок:

- ◆ UNIT — имя службы (единицы);
- ◆ LOAD — состояние службы. Статус `loaded` означает, что конфигурационный файл службы был успешно запущен;
- ◆ ACTIVE — активность службы. Статус `active` означает, что служба была успешно запущена;
- ◆ SUB — текущее состояние службы. Статус `running` означает, что в данный момент времени служба выполняется, а статус `exited` свидетельствует о том, что служба успешно завершила свою работу;
- ◆ DESCRIPTION — краткое описание службы.

ПРИМЕЧАНИЕ

Навигация по списку служб производится клавишами-стрелками `<↓>` и `<↑>`.

Исходя из полученного вывода, мы можем узнать, что служба `cups.service` в данный момент загружена, о чем свидетельствует ее статус `loaded` в столбце `LOAD`.

В данный момент времени она активна, о чем говорит статус `active` в одноименном столбце.

Если вам требуется отобразить список только запущенных служб, то для этого предназначена следующая конструкция:

```
systemctl list-units --type service --state=running
```

Аналогично производится получение списка служб, чья работа завершилась аварийно:

```
systemctl list-units --type=service --state=failed
```

Получать список служб мы научились, теперь рассмотрим, как производится запуск требуемой службы. В качестве примера запустим службу `cups.service`:

```
sudo systemctl start cups.service
```

В ответ на эту команду Терминал запросит у вас пароль и запустит службу `cups.service`. В случае успешного запуска Терминал не выведет дополнительных сообщений.

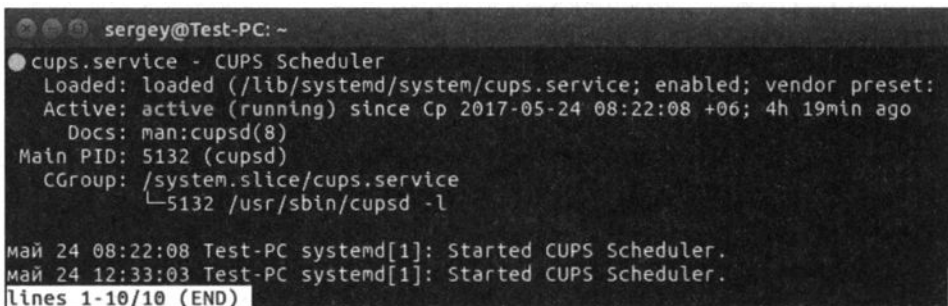
Останавливается служба аналогичным образом:

```
sudo systemctl stop cups.service
```

Если вы сомневаетесь в том, запустилась или остановилась требуемая служба, вы в любой момент времени можете просмотреть ее текущее состояние:

```
sudo systemctl status cups.service
```

В ответ на это программа `systemctl` отобразит не только текущее состояние службы, но также и ее месторасположение, время запуска и другую полезную информацию, которая может пригодиться в случае решения проблем с запуском службы (рис. 25.2).



```
sergey@Test-PC: ~
● cups.service - CUPS Scheduler
   Loaded: loaded (/lib/systemd/system/cups.service; enabled; vendor preset:
   Active: active (running) since Cp 2017-05-24 08:22:08 +06; 4h 19min ago
     Docs: man:cupsd(8)
    Main PID: 5132 (cupsd)
    CGroup: /system.slice/cups.service
            └─5132 /usr/sbin/cupsd -l

май 24 08:22:08 Test-PC systemd[1]: Started CUPS Scheduler.
май 24 12:33:03 Test-PC systemd[1]: Started CUPS Scheduler.
lines 1-10/10 (END)
```

Рис. 25.2. Просмотр текущего состояния службы

Разобраться с остальными параметрами команды `systemctl` не составит особого труда, поэтому перейдем к рассмотрению автозагрузки служб при старте операционной системы.

Автозагрузка служб и отключение служб

Те, кто знаком с принципом работы служб в ОС Windows, знают, что службы могут стартовать вместе с запуском операционной системы, а также запускаться в процессе работы операционной системы, т. е. вручную.

В ОС Ubuntu Linux схожая концепция запуска служб. Таким образом, службы могут иметь 4 состояния:

- ◆ **enabled** — данный статус означает, что служба находится в автозагрузке и будет загружена при старте операционной системы;
- ◆ **disabled** — данный статус означает, что служба не находится в автозагрузке и она не будет запущена при старте операционной системы, однако она может быть запущена вручную;
- ◆ **masked** — данный статус означает, что служба не находится в автозагрузке и не будет запущена при старте операционной системы, однако запустить вручную ее не удастся;
- ◆ **static** — данный статус означает, что служба находится в автозагрузке и будет загружена при старте операционной системы, однако данную службу невозможно отключить.

В качестве примера получим список служб, которые имеют состояние **masked**:

```
systemctl list-unit-files --state=masked
```

Данная команда позволит получить имена служб, запуск которых невозможен.

Представим ситуацию, когда вам потребовалось внести в автозагрузку вашу службу `my.program.service`. В этом случае вам поможет следующая команда:

```
sudo systemctl enable my.program.service
```

Данная команда поместит в автозагрузку требуемую службу, и она будет автоматически запускаться при старте операционной системы.

Для отключения службы используйте следующую команду:

```
sudo systemctl disable my.program.service
```

Разумеется, для выполнения предыдущих команд требуется существование самой службы, иначе Терминал сообщит о том, что данный юнит не был найден и поэтому его запуск невозможен.

Проверить успешность запуска службы можно, выполнив команду:

```
sudo systemctl is-enabled my.program.service
```

Данная команда выведет одну-единственную строчку состояния службы: **enabled**, **disabled**, **masked** или **static**.

Управление питанием

В начале знакомства с ОС Ubuntu мы научились управлять питанием компьютера с помощью команды `shutdown`, которой передавались соответствующие параметры. Однако управление питанием компьютера возможно также и с помощью служб программой `systemctl`.

Рассмотрим значения, которые принимает параметр {команда} команды `systemctl` (табл. 25.3).

Таблица 25.3. Описание параметров управления питанием команды `systemctl`

Значения параметра {команда}	Описание
<code>is-system-running</code>	Позволяет проверить, запущены ли все модули и службы операционной системы
<code>default</code>	Позволяет загрузить операционную систему с целью по умолчанию
<code>rescue</code>	Позволяет загрузить режим, предназначенный для восстановления работоспособности операционной системы
<code>emergency</code>	Позволяет загрузить аварийную оболочку операционной системы для восстановления ее работоспособности
<code>halt</code>	Позволяет остановить работу операционной системы и выключить компьютер
<code>poweroff</code>	Позволяет завершить работу операционной системы и выключить питание компьютера
<code>reboot</code>	Позволяет выполнить перезагрузку операционной системы
<code>kexec</code>	Позволяет завершить работу операционной системы и выполнить ее загрузку с использованием механизма ядра <code>kexec</code> (загрузка ядра операционной системы с пропуском процедуры аппаратной инициализации)
<code>exit</code>	Позволяет выполнить процедуры выхода из пользовательской учетной записи и остановить работу системы
<code>switch-root</code> файловая_система	Позволяет сменить корневую файловую систему на файловую систему, указанную в параметре <i>файловая_система</i>
<code>suspend</code>	Позволяет перейти в ждущий режим
<code>hibernate</code>	Позволяет перейти в спящий режим
<code>hybrid-sleep</code>	Позволяет перевести операционную систему в гибридный сон

Для перезагрузки компьютера достаточно выполнить следующую команду:

```
systemctl reboot
```

Для выключения компьютера предназначена следующая конструкция:

```
systemctl poweroff
```

Как видите, управление питанием компьютера с помощью команды `systemctl` ничуть не сложнее, чем использование команды `shutdown`.

Управление питанием ACPI

ACPI (Advanced Configuration and Power Interface — усовершенствованный интерфейс управления конфигурацией и питанием) — это интерфейс взаимодействия операционной системы, аппаратного обеспечения компьютера и микропрограммного обеспечения материнской платы (BIOS). Данный интерфейс получил свое широкое применение в ноутбуках, т. е. там, где необходимо управлять зарядом батареи, получать ответ на действие при закрытии крышки ноутбука и т. п.

В ОС Ubuntu управлением питанием ACPI занимается система инициализации `systemd`. Для этого в каталоге `/etc/systemd/` существует файл `logind.conf`, который отвечает за настройки ACPI.

Параметры файла `/etc/systemd/logind.conf` имеют следующий синтаксис:

Параметр=значение

Приведем несколько параметров, которые чаще всего используются в ОС, установленной на ноутбуках:

- ◆ `HandlePowerKey` — параметр предназначен для выбора действия при нажатии кнопки питания;
- ◆ `HandleSuspendKey` — параметр предназначен для выбора действия при нажатии кнопки ждущего режима;
- ◆ `HandleHibernateKey` — параметр предназначен для выбора действия при нажатии кнопки спящего режима;
- ◆ `HandleLidSwitch` — параметр предназначен для выбора действия при закрытии крышки ноутбука.

В качестве значений могут быть использованы следующие параметры:

- ◆ `poweroff` — позволяет завершить работу операционной системы и выключить питание;
- ◆ `suspend` — позволяет перейти в ждущий режим;
- ◆ `hibernate` — позволяет перейти в спящий режим;
- ◆ `ignore` — позволяет проигнорировать событие;
- ◆ `reboot` — позволяет выполнить перезагрузку операционной системы;
- ◆ `halt` — позволяет завершить работу операционной системы;
- ◆ `lock` — позволяет заблокировать компьютер, требуя ввода пароля для входа в систему.

По умолчанию параметры файла `/etc/systemd/logind.conf` закомментированы символом `#`. Для того чтобы задействовать тот или иной параметр, необходимо убрать символ комментария и сохранить файл.

ПРИМЕЧАНИЕ

Для редактирования файла `/etc/systemd/logind.conf` вам потребуются права `root`, для этого выполните следующую команду: `gksudo gedit /etc/systemd/logind.conf`.

Для вступления в силу внесенных вами изменений выполните команду:

```
systemctl restart systemd-logind
```

Ведение логов в systemd

Как и любая другая программа, система инициализации `systemd` позволяет вести логи (журналы) событий.

Лог (log) — текстовый файл с записями о событиях той или иной программы, расположенных в хронологическом порядке.

Размер логов и другие параметры ведения логов для системы инициализации `systemd` вы можете настроить в файле `/etc/systemd/journald.conf`. Синтаксис этого файла идентичен файлу `/etc/systemd/logind.conf`, поэтому не будем останавливаться на этом и поговорим на тему того, как читать журналы записи, оставленные системой инициализации `systemd`.

Для просмотра логов используется команда `journalctl` (рис. 25.3).

```
sergey@Test-PC: ~
-- Logs begin at Cp 2017-05-24 14:53:11 +06, end at Чт 2017-05-25 09:08:3
май 24 14:53:11 Test-PC systemd-journald[221]: Runtime journal (/run/log/
май 24 14:53:11 Test-PC kernel: Initializing cgroup subsys cpuset
май 24 14:53:11 Test-PC kernel: Initializing cgroup subsys cpu
май 24 14:53:11 Test-PC kernel: Initializing cgroup subsys cpuacct
май 24 14:53:11 Test-PC kernel: Linux version 4.4.0-75-generic (bulld@lg
май 24 14:53:11 Test-PC kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-4.
май 24 14:53:11 Test-PC kernel: KERNEL supported cpus:
май 24 14:53:11 Test-PC kernel: Intel GenuineIntel
май 24 14:53:11 Test-PC kernel: AMD AuthenticAMD
lines 1-10
```

Рис. 25.3. Просмотр логов с помощью команды `journalctl`

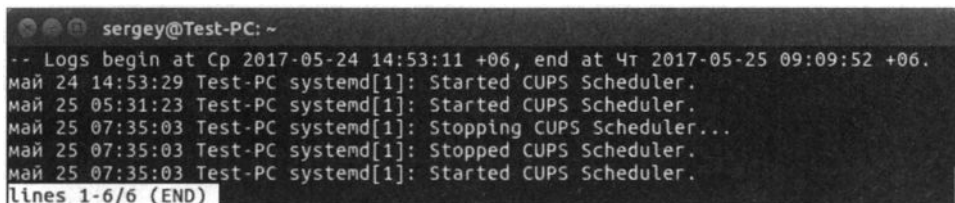
Как вы заметили, если ввести команду `journalctl` без аргументов, то она выдаст огромное количество записей и читать такой список несколько затруднительно. Для вывода записей для конкретного юнита, например `cups`, выполните следующую команду:

```
journalctl -u cups
```

Как видите, записей стало намного меньше, и теперь они касаются только требуемой службы (рис. 25.4).

ПРИМЕЧАНИЕ

Писать полное имя юнита вовсе не обязательно. Терминал корректно обработает название юнита без указания его принадлежности к службе. Таким образом, команды `journalctl -u cups` и `journalctl -u cups.service` выдадут одинаковый результат.



```
sergey@Test-PC: ~  
-- Logs begin at Cp 2017-05-24 14:53:11 +06, end at Чт 2017-05-25 09:09:52 +06.  
май 24 14:53:29 Test-PC systemd[1]: Started CUPS Scheduler.  
май 25 05:31:23 Test-PC systemd[1]: Started CUPS Scheduler.  
май 25 07:35:03 Test-PC systemd[1]: Stopping CUPS Scheduler...  
май 25 07:35:03 Test-PC systemd[1]: Stopped CUPS Scheduler.  
май 25 07:35:03 Test-PC systemd[1]: Started CUPS Scheduler.  
lines 1-6/6 (END)
```

Рис. 25.4. Вывод логов для юнита cups

Записи, выводимые командой `journalctl`, хранятся в файле `/var/log/syslog` в обычном текстовом виде. По мере наполнения файла записями, файл переименовывается в `/var/log/syslog.1` и запись продолжается в файл `/var/log/syslog`. После того как происходит заполнение файла `/var/log/syslog`, предыдущий файл `/var/log/syslog.1` переименовывается в `/var/log/syslog.2` и помещается в сжатый архив `/var/log/syslog.2.gz`. Таким образом, логи архивируются в хронологическом порядке и занимают гораздо меньший объем на жестком диске.

В большинстве случаев вам, возможно, не придется прибегать к чтению логов, однако они могут помочь вам выяснить причину того или иногда сбоя не только сервиса, но и любой другой программы.

ПРИМЕЧАНИЕ

Более подробную информацию о параметрах `journalctl` вы можете получить из справки по команде `man journalctl`.

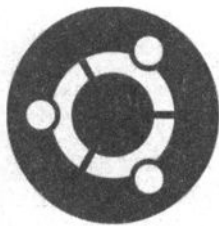
Выводы

Мы познакомились с системой инициализации `systemd`, а также узнали, что представляют собой службы в ОС Ubuntu Linux.

Мы научились управлять работой служб: останавливать, запускать и перезапускать их. Помимо этого, мы узнали, что службы могут стартовать вместе с операционной системой, и научились исключать их из автоматической загрузки.

И напоследок мы узнали, как и где искать и читать логи системы инициализации `systemd`.

ГЛАВА 26



Управление процессами в ОС Ubuntu

Процессы в ОС Ubuntu

Все мы привыкли к тому, что в ОС Windows есть Диспетчер задач, и если "зависает" какая-то программа, то он позволяет завершить зависший процесс. Однако это не всегда спасает, и единственным выходом остается перезагрузка операционной системы.

Операционная система Linux, будь то Ubuntu или другой дистрибутив, стабильнее, чем ОС Windows, и зависания программ встречаются крайне редко, но все же происходят. Так как операционная система Linux изначально была предназначена для профессионалов, то в ней не было графических средств для завершения зависших процессов. Все действия выполнялись исключительно в Консоли. Конечно, сейчас для ОС Ubuntu уже существуют программы-аналоги Диспетчера задач, но командная строка предоставляет более мощный инструмент по управлению процессами. К тому же, умение пользоваться командами является большим достоинством. Почему? Представьте ситуацию, когда зависла программа с графическим интерфейсом, а установленный вами аналог Диспетчера задач отказывается запускаться, и при этом вы не знаете никакого другого способа завершить процесс, кроме как перезагрузить компьютер. Неразумно, правда? Если бы вы знали команды завершения процессов, то вам не пришлось перезагружать операционную систему. Вы бы просто ввели в Консоли команду завершения процесса и вернулись в графический интерфейс.

Давайте приступим к изучению управления процессами в ОС Ubuntu Linux.

Системный монитор процессов *top*

Предположим, что, работая за компьютером, вы стали замечать торможение в работе программ, словно какой-то процесс отнимает много ресурсов у компьютера. Для того чтобы выяснить, какой процесс наиболее нагружает ресурсы компьютера, разработчики включили в состав операционной системы Ubuntu программу *top* (от англ. *table of processes* — таблица процессов).

Системный монитор *top* в режиме реального времени отображает процессы и сортирует их по нагрузке на процессор. Запускается программа одноименной командой (рис. 26.1).

```

sergey@Test-PC: ~
top - 12:34:45 up 21:41,  1 user,  load average: 0,22, 0,22, 0,18
Tasks: 236 total,   1 running, 235 sleeping,   0 stopped,   0 zombie
%Cpu(s):  6,8 us,  2,4 sy,   0,0 ni, 88,7 id,  2,0 wa,   0,0 hi,   0,0 si,   0,0 st
KiB Mem : 6050288 total,  356780 free, 2068440 used, 3625068 buff/cache
KiB Swap: 4607996 total, 4607996 free,   0 used, 3491148 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM     TIME+ COMMAND
 2025 sergey    20   0 1709628 159080 82968 S   6,0   2,6   7:25.25 compiz
   954 root       20   0  576044 121884  56108 S   3,3   2,0   6:14.08 Xorg
 6790 sergey    20   0 1461556 191160 108440 S   1,7   3,2   0:58.79 soffice.bin
 7733 root       20   0     0     0     0 S   1,0   0,0   0:00.03 kworker/0:2
 2290 sergey    20   0  974224  15420   9192 S   0,7   0,3   5:47.18 TVGuiSlave+
 2881 sergey    20   0  671812  50672  30800 S   0,7   0,8   0:08.32 gnome-term+
    7 root       20   0     0     0     0 S   0,3   0,0   1:42.49 rcu_sched
   547 root       20   0   21328  10200   2024 S   0,3   0,2   0:05.71 mount.ntfs
 1388 root       20   0 165024  11240   9904 S   0,3   0,2   4:42.91 teamviewerd
 1667 sergey    20   0   44080   4612   2888 S   0,3   0,1   0:07.81 dbus-daemon
 1698 sergey    20   0  348900   9416   5720 S   0,3   0,2   0:32.89 ibus-daemon
 1746 sergey    20   0   473656 32092  26448 S   0,3   0,5   0:06.45 ibus-ui-gt+
 1803 sergey    20   0  739396  64680  34040 S   0,3   1,1   0:41.20 hud-service
 1818 sergey    20   0  650056  56580  27420 S   0,3   0,9   0:21.11 unity-pane+
 1887 sergey     9 -11  437128  15248   9668 S   0,3   0,3   0:00.10 pulseaudio
 2291 sergey    20   0  129460  33520  19536 S   0,3   0,6   5:38.80 TVGuiDeleg+
 2361 sergey    20   0 1209092 235300 123004 S   0,3   3,9   8:57.15 chrome

```

Рис. 26.1. Отображение нагрузки на процессор программой *top*

Вывод программы разбит на 12 колонок:

- ◆ PID — идентификатор процесса;
- ◆ USER — пользователь, от чьего имени запущен процесс;
- ◆ PR — текущий приоритет процесса;
- ◆ NI — числовое значение "nice", влияющее на приоритет процесса;
- ◆ VIRT — объем виртуальной памяти, которая используется процессом;
- ◆ RES — объем физической памяти, используемой данным процессом;
- ◆ SHR — объем разделяемой памяти процесса;
- ◆ S — статус процесса (S — sleep, R — running, Z — zombie);
- ◆ %CPU — процент использования ресурсов процессора;
- ◆ %MEM — процент использования оперативной памяти данным процессом;
- ◆ TIME+ — общее время активности данного процесса;
- ◆ COMMAND — имя команды, которая запустила данный процесс.

Теперь вернемся на первую строку (*top*) вывода программы *top* и разберем ее на составляющие:

- ◆ 12:34:45 — текущее системное время;
- ◆ up 21:41 — общее время работы системы, которое отсчитывается с момента запуска операционной системы;

- ♦ 1 user — количество открытых пользовательских сессий. В данном случае, на компьютере работает 1 пользователь;
- ♦ load average: 0,22, 0,22, 0,18 — три средних значения загрузки системы: нагрузка в последнюю минуту, нагрузка за последние 5 минут и за последние 15 минут.

Вторая строка (Tasks) отображает статистику работающих процессов. Также разберем ее на составляющие:

- ♦ 236 total — общее количество запущенных процессов;
- ♦ 1 running — общее количество работающих в данный момент процессов;
- ♦ 235 sleeping — количество ожидающих процессов;
- ♦ 0 stopped — количество остановленных процессов;
- ♦ 0 zombie — количество процессов, которые ожидают команды от родительского процесса.

Третья строка (%Cpu(s)) отображает статистику использования процессора. Рассмотрим более подробно каждую составляющую вывода:

- ♦ 6,8 us — процент использования пользовательскими процессами ресурсов центрального процессора;
- ♦ 2,4 sy — процент использования системными процессами ресурсов центрального процессора;
- ♦ 0,0 ni — процент использования центрального процессора процессами, приоритет которых был повышен с использованием числового значения "nice";
- ♦ 88,7 id — процент времени, когда центральный процессор простаивает без нагрузки;
- ♦ 2,0 wa — процент использования ресурсов процессора процессами, которые ожидают выполнения операций;
- ♦ 0,0 hi — процент использования ресурсов центрального процессора обработчиками аппаратных прерываний (IRQ);
- ♦ 0,2 si — процент использования ресурсов центрального процессора обработчиками программных прерываний;
- ♦ 0,0 st — процент количества ресурсов центрального процессора, которые были заимствованы гипервизором для решения задач виртуальной машины.

Четвертая и пятая строки отображают статистику использования физической оперативной памяти (КиБ Mem) и раздела файла подкачки (КиБ Swap). Рассмотрим объединенный вывод этих строк:

- ♦ 6050288 total — общее количество памяти (физической в 4-й строке и виртуальной в 5-й строке);
- ♦ 356780 free — количество свободной памяти (отдельно физической и отдельно файла подкачки);

- ◆ 2068440 used — количество используемой памяти (отдельно физической и отдельно файла подкачки);
- ◆ 362568 buff/cache — количество используемой физической памяти в кэше буферов;
- ◆ 3491148 avail Mem — количество доступной виртуальной памяти.

Содержимое окна системного монитора `top` обновляется каждые 5 секунд, и по умолчанию первым в списке процессов отображается тот процесс, который больше всего использует ресурсы центрального процессора, т. е. вывод отсортирован по колонке `%CPU`. Однако вы вправе отсортировать отображаемый список процессов по одному из следующих принципов:

- ◆ сортировка по PID в порядке убывания выполняется нажатием клавиш `<Shift>+<N>`;
- ◆ сортировка по возрасту выполняется нажатием клавиш `<Shift>+<A>`;
- ◆ сортировка по времени выполнения выполняется нажатием клавиш `<Shift>+<T>`;
- ◆ сортировка по использованию памяти выполняется нажатием клавиш `<Shift>+<M>`;
- ◆ сортировка по использованию ресурсов процессора выполняется нажатием клавиш `<Shift>+<P>`.

Теперь, когда мы разобрались с каждой строкой вывода системного монитора `top`, мы без труда сможем определить то приложение, которое либо зависло, либо стало потреблять большое количество ресурсов.

Итак, вы определили приложение, которое начало потреблять значительное количество ресурсов, и хотите принудительно завершить его работу. Для этого в открытом окне системного монитора `top`, в английской раскладке клавиатуры, нажмите клавишу `<k>` и введите PID того процесса, чью работу вы хотите остановить (рис. 26.2).

После этого нажмите клавишу `<Enter>` и подтвердите остановку процесса, нажав клавишу `<Enter>` (рис. 26.3).

```

sergey@Test-PC: ~
top - 08:59:00 up 1 day, 18:06, 1 user, load average: 0,26, 0,34, 0,30
Tasks: 224 total, 2 running, 222 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1,6 us, 0,4 sy, 0,0 ni, 98,1 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
КиБ Mem : 6050288 total, 244980 free, 2339648 used, 3465660 buff/cache
КиБ Swap: 4607996 total, 4604152 free, 3844 used. 3120352 avail Mem

PID to signal/kill [default pid = 23902] 23902
  PID USER  PR  NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
 23902 sergey  20   0 1090912 215128 125684 S    1,5  3,6    0:26.06 chrome
    7 root    20   0      0      0      0 S    0,8  0,0    3:27.51 rcu_sched
   954 root    20   0  575088 121156  55380 S    0,8  2,0   13:56.93 Xorg
  1698 sergey  20   0  350212  10864   5720 S    0,8  0,2    0:56.13 ibus-daemon
  2025 sergey  20   0 1710460 160184  82964 S    0,8  2,6   21:48.13 compiz
  2290 sergey  20   0  974480  15640   9192 S    0,8  0,3   11:09.91 TVGuiSlave+
  2291 sergey  20   0 1294600 33520  19536 S    0,8  0,6   10:55.30 TVGuiDeleg+
  2361 sergey  20   0 1369684 264924 122040 S    0,8  4,4   14:06.63 chrome
  
```

Рис. 26.2. Завершение работы процесса с PID равным 23902

```

sergey@Test-PC: ~
top - 09:08:25 up 1 day, 18:15, 1 user, load average: 0,16, 0,45, 0,38
Tasks: 222 total, 5 running, 217 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1,0 us, 0,3 sy, 0,0 ni, 98,7 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
КиБ Mem : 6050288 total, 559244 free, 2100748 used, 3390296 buff/cache
КиБ Swap: 4607996 total, 4604152 free, 3844 used, 3434980 avail Mem
Send pid 23902 signal [15/sigterm]
  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND

```

Рис. 26.3. Программа `top` требует подтверждения завершения работы процесса

После того как вы подтвердили завершение процесса, требуемый процесс будет остановлен, и вы сможете продолжить работу в системе. Как видите, ничего сложного, и вовсе не нужно перезагружать компьютер.

ПРИМЕЧАНИЕ

Если не ввести PID процесса и нажать клавишу `<Enter>`, то по умолчанию будет завершен тот процесс, который стоит первым в списке.

Мы рассмотрели лишь небольшую часть функционала системного монитора `top`. Для получения более подробной информации о работе с этой программой выполните команду `man top`.

Получение списка процессов

Системный монитор `top` — это один из способов получения списка процессов. Помимо него существует множество программ, способных выполнять подобные действия, но предназначены они для несколько иных задач. Так, в ОС Ubuntu присутствует программа `ps`, которая, в отличие от системного монитора `top`, позволяет отображать статический снимок процессов.

Программа `ps` (от англ. *process status*) — это консольная программа в Linux-подобных дистрибутивах, позволяющая выводить отчет о запущенных в системе процессах.

Синтаксис программы `ps` следующий:

`ps [параметры]`

Рассмотрим наиболее часто используемые параметры программы `ps` (табл. 26.1).

Таблица 26.1. Параметры команды `ps`

Параметр	Описание
<code>-A</code>	Позволяет отобразить все работающие процессы
<code>-a</code>	Позволяет отобразить только те процессы, которые связаны с данным Терминалом, за исключением системных процессов сеанса
<code>-d</code>	Позволяет отобразить все процессы, кроме главных процессов сеанса
<code>-o</code>	Позволяет задать отображаемые в выводе столбцы

Таблица 26.1 (окончание)

Параметр	Описание
-T	Позволяет отобразить все процессы, которые были запущены в текущем Терминале
-r	Позволяет вывести информацию только о работающих в данный момент процессах
-u	Позволяет вывести в удобном для восприятия виде
-x	Позволяет отобразить все работающие процессы

В качестве примера отобразим пользовательские процессы в удобном для восприятия виде:

```
ps -aux
```

В результате выполнения команды вывод будет разбит на 11 колонок, некоторые из них вам уже будут знакомы:

- ◆ USER — имя пользователя, от чьего имени был запущен процесс;
- ◆ PID — идентификатор процесса;
- ◆ %CPU — процент использования ресурсов процессора;
- ◆ %MEM — процент использования оперативной памяти;
- ◆ VSZ — виртуальный размер запущенного процесса;
- ◆ RSS — занимаемый процессом объем памяти;
- ◆ TTY — номер Терминала, в котором был запущен процесс;
- ◆ STAT — состояние, в котором сейчас находится процесс;
- ◆ START — дата запуска процесса;
- ◆ TIME — процессорное время, которое потребовалось для запуска процесса;
- ◆ COMMAND — команда, которая запустила процесс.

Вывод оказался очень большим и переполненным лишними столбцами. Давайте укажем только нужные столбцы:

```
ps -Ao pid,user,command
```

В результате выполнения данной команды мы получим вывод, состоящий из трех столбцов: идентификатора процесса, имени пользователя, который запустил процесс, и столбца, в котором указана команда, запустившая процесс (рис. 26.4).

```

sergey@Test-PC: ~
sergey@Test-PC:~$ ps -Ao pid,user,command
  PID USER      COMMAND
    1 root      /sbin/init splash
    2 root      [kthreadd]
    3 root      [ksoftirqd/0]

```

Рис. 26.4. Форматированный вывод списка процессов

Даже в этом случае, при явном указании нужных столбцов, вывод оказался объемным. В таком количестве информации довольно легко пропустить искомый процесс, поэтому попробуем сократить вывод, выполнив следующую команду:

```
ps -A | grep office
```

Теперь в вывод попадут только те процессы, которые имеют в своем названии искомое слово *office* (рис. 26.5).

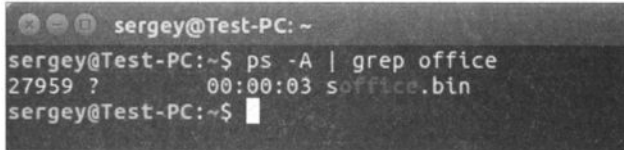


Рис. 26.5. Вывод информации о процессе по искомому слову

Получив идентификатор искомого процесса, нужно завершить процесс следующей командой:

```
kill 27959
```

О том, откуда взялась команда *kill*, мы сейчас и поговорим.

Завершение процесса командой *kill*

Программа *kill* (от англ. *kill* — убить) — это команда, предназначенная для передачи сигнала указанному процессу. В список сигналов, которые может посылать команда *kill*, входит команда прекращения выполнения процесса.

На данный момент командой *kill* поддерживается 64 сигнала. Перечислять все 64 сигнала мы не будем. Назовем только часто используемые сигналы.

- ◆ **SIGINT** — Interrupt (прерывание). Данный сигнал означает завершение работы процесса, но перед этим процесс может корректно завершить свою работу.
- ◆ **SIGQUIT** — сигнал сообщает процессу о том, что он должен завершить свою работу. Перед этим процесс генерирует дамп памяти.
- ◆ **SIGILL** — Illegal Instruction (неверная инструкция). Данный сигнал посылается процессу в том случае, когда центральный процессор встречается незнакомую для него команду.
- ◆ **SIGKILL** — сигнал прекращает работу процесса, при этом работа процесса прерывается без сохранения данных.
- ◆ **SIGTERM** — Software Termination (программное завершение). Данный сигнал завершает процесс, но завершение процесса обрабатывается самим процессом, позволяя ему завершить свою работу.
- ◆ **SIGSTOP** — сигнал останавливает работу процесса.

Ознакомиться с полным списком сигналов вы можете, выполнив следующую команду:

```
kill -l
```

Вывод, полученный в результате выполнения этой команды, будет разбит на несколько колонок и пронумерован значениями, указывающими номер сигнала (рис. 26.6).

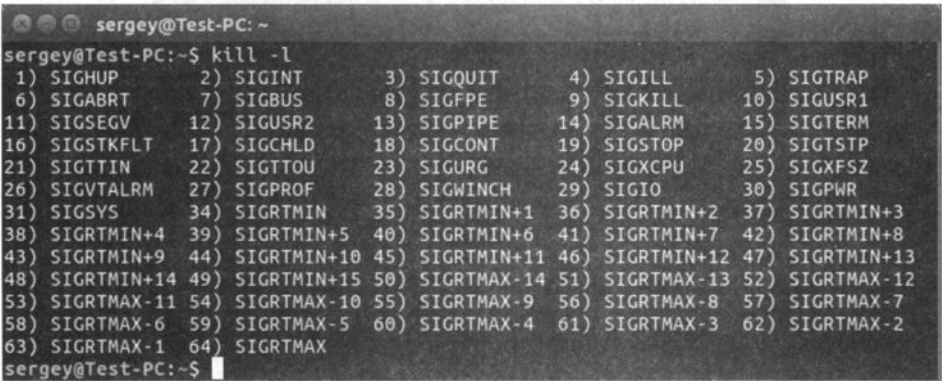


Рис. 26.6. Список сигналов команды kill

Как видим, сигнал SIGSTOP имеет номер 19, а сигнал SIGKILL номер 9. Это означает, что вместо написания названий сигналов можно использовать цифровые эквиваленты, соответствующие номеру сигнала.

Давайте рассмотрим синтаксис команды kill:

```
kill [опции] <pid>
```

В параметре <pid> указывается PID процесса, которому требуется послать сигнал. Параметр [опции] может принимать значения, перечисленные в табл. 26.2.

Таблица 26.2. Опции команды kill

Сокращенный вариант	Полный вариант	Описание
-<сигнал> -s <сигнал>	--signal <сигнал>	Позволяет задать имя или номер посылаемого сигнала
-l	--list [сигнал]	Позволяет вывести имена сигналов (имя сигнала), соответствующие номеру или имени, переданному в параметре [сигнал]
-L	--table	Позволяет вывести информацию о сигналах в виде таблицы

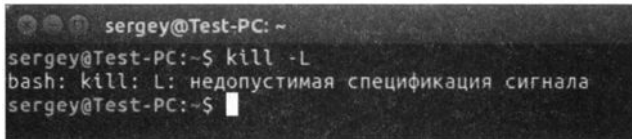
По умолчанию, если команде kill не передан номер или имя сигнала, присваивается команда SIGTERM, т. е. номер 15. Это означает, что в случае выполнения команды

`kill 27959` процессу будет послан сигнал `SIGTERM`, означающий завершение работы процесса им же самим.

Попробуем выполнить следующую команду:

```
kill -L
```

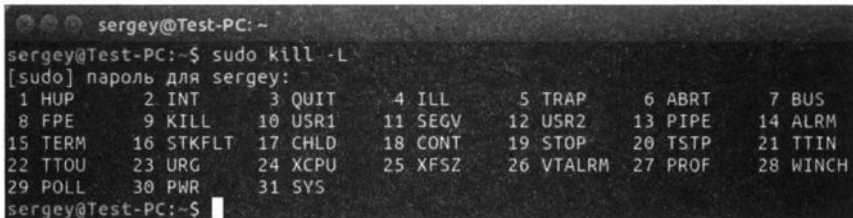
В ответ на это Терминал выдаст ошибку (рис. 26.7).



```
sergey@Test-PC: ~
sergey@Test-PC:~$ kill -L
bash: kill: L: недопустимая спецификация сигнала
sergey@Test-PC:~$
```

Рис. 26.7. Недопустимая спецификация сигнала

Ошибка возникает в виду того, что для выполнения данной команды необходимы `root`-права. Поэтому при выполнении этой команды с соответствующими правами вы получите список сигналов в виде таблицы (рис. 26.8).



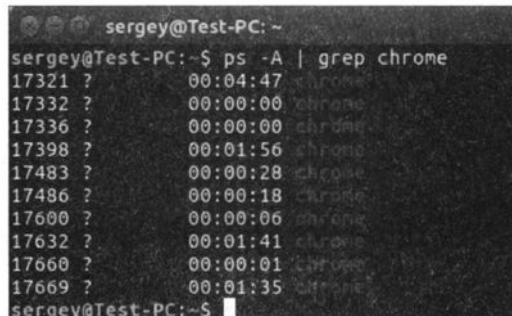
```
sergey@Test-PC: ~
sergey@Test-PC:~$ sudo kill -L
[sudo] пароль для sergey:
 1 HUP      2 INT      3 QUIT     4 ILL      5 TRAP     6 ABRT     7 BUS
 8 FPE      9 KILL     10 USR1    11 SEGV    12 USR2    13 PIPE    14 ALRM
15 TERM    16 STKFLT  17 CHLD    18 CONT    19 STOP    20 TSTP    21 TTIN
22 TTOU    23 URG     24 XCPU    25 XF SZ    26 VTALRM  27 PROF    28 WINCH
29 POLL    30 PWR     31 SYS
sergey@Test-PC:~$
```

Рис. 26.8. Выполнение команды с `root`-правами

В большинстве случаев использование дополнительных опций не требуется. Для остановки процесса достаточно указать его PID, и процесс будет остановлен.

Завершение процесса командой *killall*

Команда `killall` успешно справляется с одиночным процессом. А как быть в случае, если процессов несколько и требуется завершить все процессы, имеющие одинаковое имя, как это бывает с процессом `chrome` (рис. 26.9)?



```
sergey@Test-PC: ~
sergey@Test-PC:~$ ps -A | grep chrome
17321 ?        00:04:47 chrome
17332 ?        00:00:00 chrome
17336 ?        00:00:00 chrome
17398 ?        00:01:56 chrome
17483 ?        00:00:28 chrome
17486 ?        00:00:18 chrome
17600 ?        00:00:06 chrome
17632 ?        00:01:41 chrome
17660 ?        00:00:01 chrome
17669 ?        00:01:35 chrome
sergey@Test-PC:~$
```

Рис. 26.9. Несколько процессов `chrome` с разными идентификаторами

Завершать каждый процесс по отдельности не очень целесообразно, поэтому в состав ОС Ubuntu была включена программа `killall`.

Программа `killall` — консольная утилита в составе операционных систем на базе ОС Linux, позволяющая отправить сигнал всем процессам с указанным именем. При этом знать PID каждого процесса не обязательно.

Синтаксис команды `killall` следующий:

```
killall [опции] [сигнал] имя_процесса
```

Рассмотрим список опций, которые может принимать команда `killall` (табл. 26.3).

Таблица 26.3. Опции команды `killall`

Сокращенный вариант	Полный вариант	Описание
-e	--exact	Позволяет учитывать полные совпадения длинных имен
-I	--ignore-case	Позволяет игнорировать регистр написания имени процесса
-g	--process-group	Позволяет завершить работу группы процессов
-y	--younger-than	Позволяет завершить работу процессов, новее заданного <i><времени></i>
-o	--older-than	Позволяет завершить работу процессов, запуск которых произошел раньше указанного <i><времени></i>
-i	--interactive	Позволяет запрашивать подтверждение перед завершением процессов
-l	--list	Позволяет вывести список имен допустимых сигналов
-q	--quiet	Позволяет не выводить подробный отчет
-r	--regexp <i><имя></i>	Позволяет рассматривать заданное <i><имя></i> как регулярное выражение
-s	--signal <i><сигнал></i>	Позволяет послать процессу указанный сигнал, а не его имя
-u	--user <i><пользователь></i>	Позволяет завершить процессы указанного пользователя
-v	--verbose	Позволяет вывести ответ в случае успешной обработки процессом отправленного сигнала
-w	--wait	Позволяет дождаться завершения работы процесса
-Z	--context <i><имя></i>	Позволяет завершать работу процессов, имеющих контекст с указанным именем

Для завершения всех процессов с именем `chrome` выполните следующую команду:

```
killall chrome
```

В результате все процессы с указанным именем будут корректно завершены.

По умолчанию, если не указан сигнал, команда `killall` отправляет процессу сигнал `SIGTERM`. Для того чтобы отправить конкретный сигнал, выполните следующую команду:

```
killall -s 9 chrome
```

Рассмотренные способы завершения процессов сработают в том случае, если графический интерфейс продолжает отвечать на ваши команды. Если графический интерфейс перестал реагировать на вводимые команды, вам поможет Консоль, с которой мы познакомились несколько глав назад.

Давайте рассмотрим работу с Консолью для управления процессами в ОС Ubuntu Linux.

Управление процессами в Консоли

В том случае, когда "зависает" графический интерфейс, нам на помощь приходит Консоль. Ее можно запустить прямо из графического режима, нажав одно из сочетаний клавиш:

- ◆ `<Ctrl>+<Alt>+<F1>` — для запуска первой виртуальной консоли;
- ◆ `<Ctrl>+<Alt>+<F2>` — для запуска второй виртуальной консоли;
- ◆ `<Ctrl>+<Alt>+<F3>` — для запуска третьей виртуальной консоли;
- ◆ `<Ctrl>+<Alt>+<F4>` — для запуска четвертой виртуальной консоли;
- ◆ `<Ctrl>+<Alt>+<F5>` — для запуска пятой виртуальной консоли;
- ◆ `<Ctrl>+<Alt>+<F6>` — для запуска шестой виртуальной консоли;
- ◆ `<Ctrl>+<Alt>+<F7>` — для запуска седьмой виртуальной консоли и возврата в графический режим.

После запуска одной из виртуальных консолей от вас требуется ввести свои логин и пароль, которые соответствуют вашей учетной записи. Теперь вы можете выполнять все те же команды, что и при работе в графическом режиме.

Так, для перезапуска графического окружения выполните следующую команду:

```
sudo systemctl restart lightdm
```

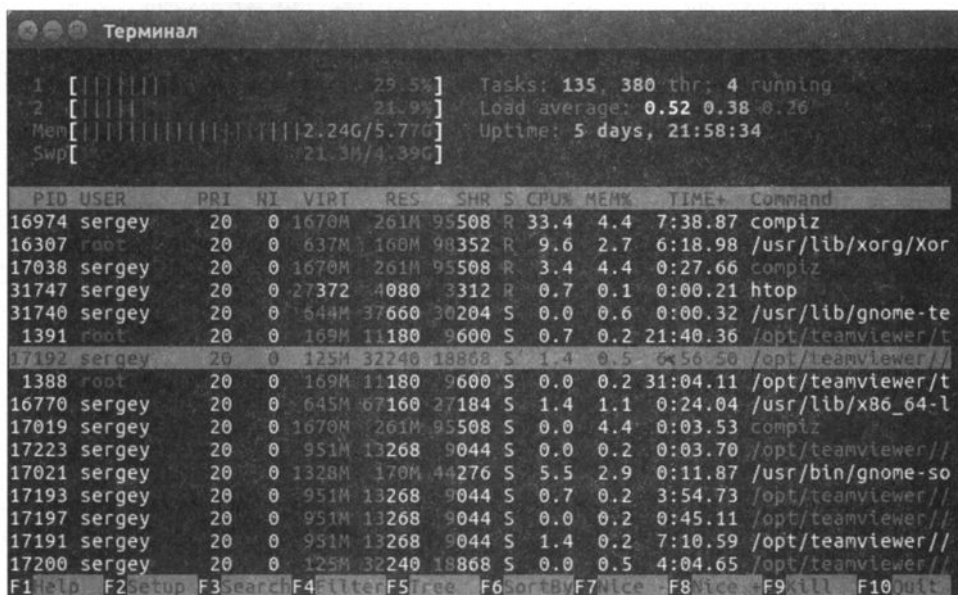
После этого дисплейный менеджер LightDM (The Light Display Manager) перезапустится, и вы сможете продолжить работу в графическом режиме, минуя перезагрузку операционной системы.

К этому методу перезагрузки графического окружения стоит прибегать в случае, когда вы наблюдаете сбой в его работе, либо невозможности продолжить работу в операционной системе.

Прочие инструменты по управлению процессами

На данный момент времени для управления процессами существует великое множество программ, которые доступны как в Менеджере приложений Ubuntu, так и на сторонних репозиториях.

Вместо консольной программы `top` многие предпочитают более удобную в плане управления программу `htop`. Она доступна в Менеджере приложений Ubuntu и имеет псевдографический интерфейс (рис. 26.10).



Терминал

1 [|||||] 29.5% Tasks: 135, 380 thr: 4 running
 2 [|||||] 21.9% Load average: 0.52 0.38 0.26
 Mem [|||||] 12.24G/5.77G Uptime: 5 days, 21:58:34
 Swp [|||||] 21.3M/4.39G

PID	USER	PRI	NI	VRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
16974	sergey	20	0	1670M	261M	95508	R	33.4	4.4	7:38.87	complz
16307	root	20	0	637M	168M	98352	R	9.6	2.7	6:18.98	/usr/lib/xorg/Xor
17038	sergey	20	0	1670M	261M	95508	R	3.4	4.4	0:27.66	complz
31747	sergey	20	0	27372	4080	3312	R	0.7	0.1	0:00.21	htop
31740	sergey	20	0	644M	37660	30204	S	0.0	0.6	0:00.32	/usr/lib/gnome-te
1391	root	20	0	169M	11180	9600	S	0.7	0.2	21:40.36	/opt/teamviewer/t
17192	sergey	20	0	125M	32240	18868	S	1.4	0.5	6:56.50	/opt/teamviewer//
1388	root	20	0	169M	11180	9600	S	0.0	0.2	31:04.11	/opt/teamviewer/t
16770	sergey	20	0	645M	67160	27184	S	1.4	1.1	0:24.04	/usr/lib/x86_64-l
17019	sergey	20	0	1670M	261M	95508	S	0.0	4.4	0:03.53	complz
17223	sergey	20	0	951M	13268	9044	S	0.0	0.2	0:03.70	/opt/teamviewer//
17021	sergey	20	0	1328M	170M	44276	S	5.5	2.9	0:11.87	/usr/bin/gnome-so
17193	sergey	20	0	951M	13268	9044	S	0.7	0.2	3:54.73	/opt/teamviewer//
17197	sergey	20	0	951M	13268	9044	S	0.0	0.2	0:45.11	/opt/teamviewer//
17191	sergey	20	0	951M	13268	9044	S	1.4	0.2	7:10.59	/opt/teamviewer//
17200	sergey	20	0	125M	32240	18868	S	0.0	0.5	4:04.65	/opt/teamviewer//

F1:help F2:Setup F3:Search F4:Filter F5:Free F6:SortBy F7:Nice F8:Nice F9:kill F10:quit

Рис. 26.10. Программа `htop`

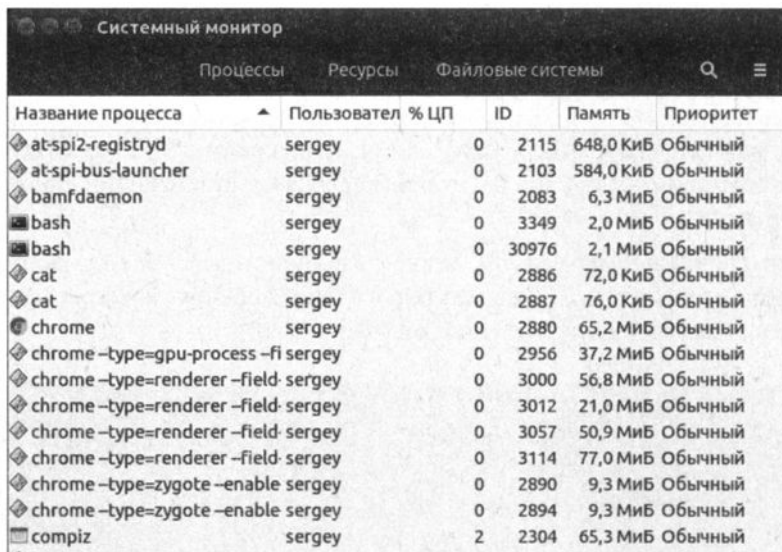
Программа `htop` позволяет выделять нужный процесс и завершать его работу, не вводя его идентификатора. Для завершения процесса вам достаточно выделить нужный процесс мышью или клавишами-стрелками и нажать клавишу `<F9>`, а затем подтвердить завершение процесса клавишей `<Enter>`.

В стандартной установке ОС Ubuntu присутствует графическая утилита под названием Системный монитор (рис. 26.11).

Запустить Системный монитор можно из главного меню, введя в строке поиска фразу `системный монитор` и выбрав соответствующий ярлык для запуска. Также вы можете вызвать Системный монитор из Терминала следующей командой:

```
gnome-system-monitor
```

Разобраться в этой графической утилите вам не составит труда, т. к. Системный монитор напоминает Диспетчер задач, который вы привыкли видеть в ОС Windows, поэтому отдельно рассматривать работу в нем мы не будем.



The screenshot shows the 'Системный монитор' (System Monitor) window. It has tabs for 'Процессы' (Processes), 'Ресурсы' (Resources), and 'Файловые системы' (File Systems). The 'Процессы' tab is active, displaying a table of running processes. The table has columns for 'Название процесса' (Process Name), 'Пользователь' (User), '% ЦП' (CPU %), 'ID', 'Память' (Memory), and 'Приоритет' (Priority). The processes listed include system services like 'at-spi2-registryd', user applications like 'bash', 'cat', and 'chrome', and system components like 'compiz'.

Название процесса	Пользователь	% ЦП	ID	Память	Приоритет
at-spi2-registryd	sergey	0	2115	648,0 КиБ	Обычный
at-spi-bus-launcher	sergey	0	2103	584,0 КиБ	Обычный
bamfdemon	sergey	0	2083	6,3 МиБ	Обычный
bash	sergey	0	3349	2,0 МиБ	Обычный
bash	sergey	0	30976	2,1 МиБ	Обычный
cat	sergey	0	2886	72,0 КиБ	Обычный
cat	sergey	0	2887	76,0 КиБ	Обычный
chrome	sergey	0	2880	65,2 МиБ	Обычный
chrome -type=gpu-process -fi	sergey	0	2956	37,2 МиБ	Обычный
chrome -type=renderer -field	sergey	0	3000	56,8 МиБ	Обычный
chrome -type=renderer -field	sergey	0	3012	21,0 МиБ	Обычный
chrome -type=renderer -field	sergey	0	3057	50,9 МиБ	Обычный
chrome -type=renderer -field	sergey	0	3114	77,0 МиБ	Обычный
chrome -type=zygote -enable	sergey	0	2890	9,3 МиБ	Обычный
chrome -type=zygote -enable	sergey	0	2894	9,3 МиБ	Обычный
compiz	sergey	2	2304	65,3 МиБ	Обычный

Рис. 26.11. Системный монитор

По умолчанию Системный монитор отображает процессы текущего пользователя. Для того чтобы просмотреть процессы, требующие привилегий root, выполните соответствующую команду:

```
sudo gnome-system-monitor
```

Выводы

Мы изучили очень важную тему, посвященную управлению процессами в ОС Ubuntu Linux. Полученные знания и навыки помогут вам грамотно и правильно управлять процессами в рассматриваемой операционной системе.

Мы познакомились с различными инструментами управления процессами, умение пользоваться каждым из которых понадобится в дальнейшей работе с ОС Ubuntu.

Приобретенные знания в ходе изучения данной главы помогут вам без труда освоить другие инструменты по управлению процессами в этой операционной системе.

ГЛАВА 27



Проверка работоспособности сети в ОС Ubuntu

Сеть в ОС Ubuntu

Про настройку соединения с Интернетом мы говорили в начале знакомства с ОС Ubuntu Linux, однако до сих пор не рассматривали ситуацию, когда необходимо проверить его работоспособность.

Всем нам знакома ситуация, когда, включив компьютер, мы обнаруживаем, что доступ в Интернет отсутствует. Многие недолго думая, звонят в службу техподдержки своего провайдера, и оператор долго и упорно пытается объяснить, что и как нужно нажимать, чтобы "Интернет заработал". В большинстве случаев провайдер ни при чем, и доступа в Интернет нет по причине каких-либо сбоев в программном обеспечении вашего компьютера или в результате неправильных действий со стороны пользователя. Например, часто пользователь в силу своей неопытности отключает сетевой адаптер, и сеть перестает работать. Разумеется, провайдер в этом не виноват, и проблему нужно искать в настройках сетевого оборудования. Благо в ОС Windows выяснить проблему неработающей сети довольно легко.

В ОС Ubuntu Linux проверка работоспособности сети осуществляется штатными средствами, но в основном в режиме командной строки, т. е. используя Терминал. Не стоит пугаться командной строки, т. к. проверка работоспособности сети не представляет сложности. Сама диагностика выполняется поэтапно для удобства и простоты определения причин неработоспособности соединения с Интернетом. Возможно, уже на первом или втором этапе проверки вы найдете причину неисправности.

Определение сетевого оборудования

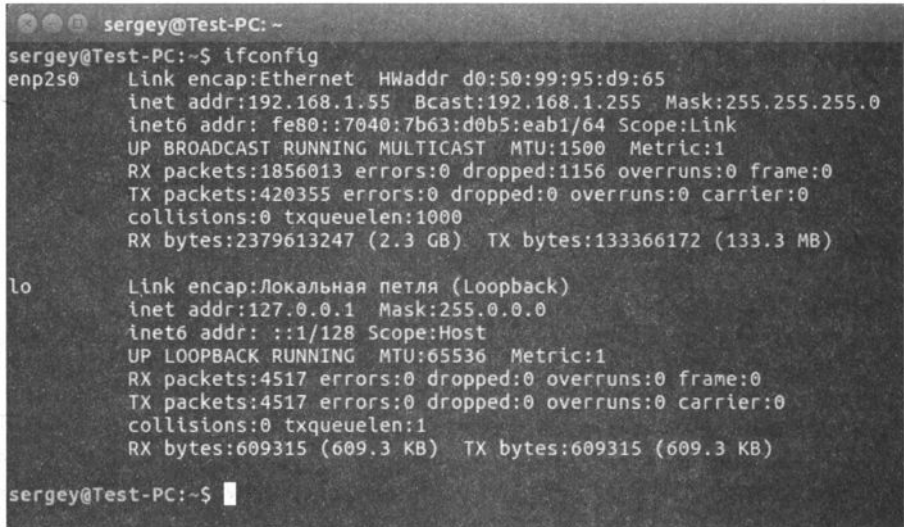
Первым делом необходимо убедиться в том, что сетевая карта вашего компьютера успешно определяется и работает в ОС Ubuntu. Для этого выполните следующую команду:

```
ifconfig
```


ПРИМЕЧАНИЕ

Проверку подключения сетевого кабеля к сетевой карте компьютера или Wi-Fi-роутеру мы опустим, т. к. это само собой разумеется.

В том случае если сетевая карта успешно определена операционной системой и включена, то вы увидите примерно следующую картину (рис. 27.1).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ ifconfig  
enp2s0      Link encap:Ethernet  HWaddr d0:50:99:95:d9:65  
            inet addr:192.168.1.55  Bcast:192.168.1.255  Mask:255.255.255.0  
            inet6 addr: fe80::7040:7b63:d0b5:eab1/64 Scope:Link  
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
            RX packets:1856013 errors:0 dropped:1156 overruns:0 frame:0  
            TX packets:420355 errors:0 dropped:0 overruns:0 carrier:0  
            collisions:0 txqueuelen:1000  
            RX bytes:2379613247 (2.3 GB)  TX bytes:133366172 (133.3 MB)  
  
lo          Link encap:Локальная петля (Loopback)  
            inet addr:127.0.0.1  Mask:255.0.0.0  
            inet6 addr: ::1/128 Scope:Host  
            UP LOOPBACK RUNNING  MTU:65536  Metric:1  
            RX packets:4517 errors:0 dropped:0 overruns:0 frame:0  
            TX packets:4517 errors:0 dropped:0 overruns:0 carrier:0  
            collisions:0 txqueuelen:1  
            RX bytes:609315 (609.3 KB)  TX bytes:609315 (609.3 KB)  
  
sergey@Test-PC:~$
```

Рис. 27.1. Обнаружена сетевая карта

Команда `ifconfig` показала два сетевых интерфейса: `enp2s0` и `lo`. Как раз первый и является сетевой картой. Второй сетевой интерфейс является локальной петлей, т. е. по своей сути играет роль виртуального сетевого интерфейса, предназначенного для различных тестов передачи данных в сети.

Наличие второй строки "`inet addr...`" свидетельствует о том, что соединение с сетью установлено и работоспособно.

Если, выполнив команду `ifconfig`, вы увидели только один виртуальный сетевой интерфейс `lo`, то это значит, что физический сетевой интерфейс был отключен.

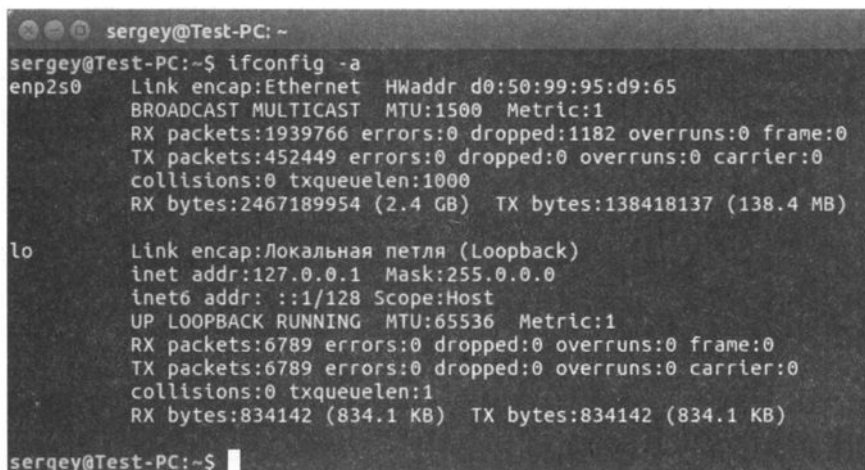
ПРИМЕЧАНИЕ

Данное действие справедливо при условии, что при установке операционной системы сетевая карта была успешно определена или вы вручную устанавливали драйверы.

Для того чтобы включить физический сетевой интерфейс, который ранее был рабочим, вам необходимо сначала узнать его имя:

```
ifconfig -a
```

В результате выполнения данной команды вы увидите заветный сетевой интерфейс с именем `enp2s0` (рис. 27.2).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ ifconfig -a  
enp2s0    Link encap:Ethernet  HWaddr d0:50:99:95:d9:65  
          BROADCAST MULTICAST  MTU:1500  Metric:1  
          RX packets:1939766 errors:0 dropped:1182 overruns:0 frame:0  
          TX packets:452449 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:2467189954 (2.4 GB)  TX bytes:138418137 (138.4 MB)  
  
lo        Link encap:Локальная петля (Loopback)  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING  MTU:65536  Metric:1  
          RX packets:6789 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:6789 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1  
          RX bytes:834142 (834.1 KB)  TX bytes:834142 (834.1 KB)  
  
sergey@Test-PC:~$
```

Рис. 27.2. Обнаружен отключенный сетевой интерфейс

Обратите внимание, что у нас отсутствует вторая строка "inet addr...", как было показано на рис. 27.2. Отсутствие этой строки говорит о том, что данный сетевой интерфейс не задействован. Давайте исправим это и задействуем его:

```
sudo ifconfig enp2s0 up
```

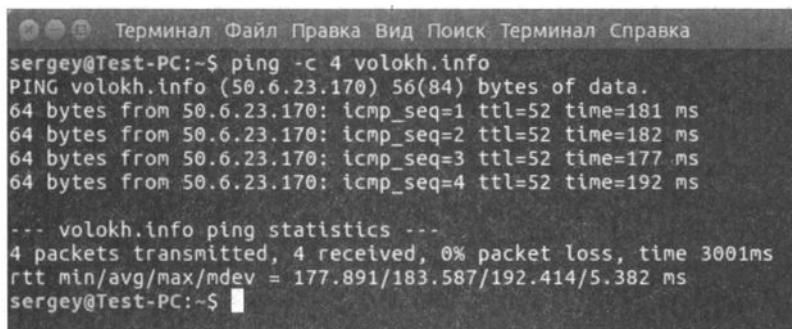
Теперь наш сетевой интерфейс `enp2s0` включен и работоспособность сети должна восстановиться.

Проверка соединения с Интернетом

В том случае когда все сетевые интерфейсы включены и настроены, а соединение с удаленным узлом все так же отсутствует, то рекомендуется прибегнуть к команде `ping`, которая покажет, доступен удаленный узел или нет:

```
ping -c 4 volokh.info
```

В случае доступности удаленного узла (сайта) результат будет примерно таким, как представлено на рис. 27.3.



```
Терминал Файл Правка Вид Поиск Терминал Справка  
sergey@Test-PC:~$ ping -c 4 volokh.info  
PING volokh.info (50.6.23.170) 56(84) bytes of data:  
64 bytes from 50.6.23.170: icmp_seq=1 ttl=52 time=181 ms  
64 bytes from 50.6.23.170: icmp_seq=2 ttl=52 time=182 ms  
64 bytes from 50.6.23.170: icmp_seq=3 ttl=52 time=177 ms  
64 bytes from 50.6.23.170: icmp_seq=4 ttl=52 time=192 ms  
  
--- volokh.info ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3001ms  
rtt min/avg/max/mdev = 177.891/183.587/192.414/5.382 ms  
sergey@Test-PC:~$
```

Рис. 27.3. Проверка доступности удаленного узла

Сейчас мы не будем подробно рассматривать синтаксис программы `ping`, а поясним лишь некоторые моменты.

Программа `ping` (от англ. *Packet InterNet Grouper*) предназначена для диагностики работы сети. В частности, она позволяет произвести проверку целостности передачи данных по сети. Программа `ping` отправляет запросы (ICMP Echo-Request) на указанный узел и фиксирует поступающие ответы (ICMP Echo-Reply).

Так, приведенная выше команда `ping` отправляет 4 пакета (параметр `-c 4`) на удаленный узел `volokh.info`. Размер каждого полученного пакета равен 64 байта. В ответ на выполненную команду мы получили следующий ответ от удаленного узла:

```
64 bytes from 50.6.23.170: icmp_seq=1 ttl=52 time=181 ms
```

Разобьем этот ответ на составляющие:

- ◆ 64 bytes from 50.6.23.170 — получено 64 байта от узла с IP-адресом 50.6.23.170;
- ◆ `icmp_seq=1` — номер отправленного пакета;
- ◆ `ttl=52` — время жизни полученного пакета;
- ◆ `time=181` — время, которое потребовалось для получения этого пакета;
- ◆ `ms` — единица измерения времени получения пакета.

После информации по каждому пакету указаны суммарные сведения по данному запросу:

```
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
```

Данная строка сообщает, что было отправлено 4 пакета, из которых были получены все 4, ноль пакетов было потеряно, и на все это понадобилось 3001 миллисекунда.

В случае недоступности удаленного узла ответ был бы примерно следующим (листинг 27.1).

Листинг 27.1. Недоступность узла 192.168.1.5

```
PING 192.168.1.5 (192.168.1.5) 56(84) bytes of data.  
From 192.168.1.55 icmp_seq=1 Destination Host Unreachable  
From 192.168.1.55 icmp_seq=2 Destination Host Unreachable  
From 192.168.1.55 icmp_seq=3 Destination Host Unreachable  
From 192.168.1.55 icmp_seq=4 Destination Host Unreachable
```

```
--- 192.168.1.5 ping statistics ---
```

```
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3014ms  
pipe 3
```

В данном случае удаленный узел с IP-адресом 192.168.1.5 недоступен, о чем свидетельствует строка "Destination Host Unreachable", а также процент потерянных пакетов в суммарной информации по данному запросу.

Таким образом, если вам не удастся загрузить нужный вам сайт, то стоит проверить его ответ, прибегнув к команде `ping`.

Однако программа `ping` не поможет вам с точностью до 100% определить причину неполадок в сети в силу разных причин, например, блокировки удаленным сервером ICMP-запросов.

Программа `ping` позволяет выяснить тот узел, на котором возникли неполадки в работе сети. Так, если вы запросили удаленный узел, а он не ответил, то рекомендуется проверить ICMP-ответ вашего провайдера. Либо если вы подключены к Интернету через маршрутизатор, то рекомендуется послать запрос на его IP-адрес и уже на основании полученного ответа искать причину неисправности.

ПРИМЕЧАНИЕ

Полный список возможностей программы `ping` вы сможете получить по команде справки `man ping`.

Трассировка маршрута

Нередко требуемый удаленный узел бывает недоступен, но при этом маршрутизатор и адрес провайдера успешно отвечают на ICMP-запросы. Казалось бы, в данном случае невозможно выяснить причину, по которой теряются пакеты где-то на половине пути следования к требуемому удаленному узлу, но даже из этой ситуации есть выход — программа `traceroute`.

Программа `traceroute` — это служебная программа, не входящая в состав ОС Ubuntu Linux и предназначенная для определения маршрутов следования переданных данных в сети.

Для установки программы `traceroute` выполните следующую команду:

```
sudo apt install traceroute
```

В качестве примера рассмотрим трассировку маршрута до узла **yandex.ru**:

```
traceroute yandex.ru
```

После выполнения данной команды мы получим результат, подобный представленному в листинге 27.2.

Листинг 27.2. Трассировка узла yandex.ru

```
traceroute to yandex.ru (77.88.55.88), 30 hops max, 60 byte packets
1 192.168.1.230 (192.168.1.230) 0.603 ms 0.675 ms 0.750 ms
2 10.254.253.253 (10.254.253.253) 1.901 ms 1.899 ms 1.884 ms
3 mx480.omkc.ru (217.25.208.193) 1.920 ms 1.929 ms 1.937 ms
4 rt1.omkc.ru (217.25.208.157) 2.096 ms 2.136 ms 2.143 ms
5 yandex-yekt.rb-ix.ru (185.1.0.57) 13.840 ms 13.731 ms 13.705 ms
6 ugr-b-cl-xe-7-2-0.yndx.net (213.180.213.18) 38.761 ms ugr-b-cl-xe-7-2-
1.yndx.net (213.180.213.20) 40.408 ms ugr-b-cl-xe-7-2-0.yndx.net
(213.180.213.18) 38.019 ms
```

```

7 m9-pl-bel14.yndx.net (87.250.239.82) 38.822 ms 38.341 ms 38.806 ms
8 iva-b-c2-ae13.yndx.net (87.250.239.104) 37.995 ms iva-b-c2-ae4.yndx.net
(87.250.239.117) 38.920 ms iva-b-c2-ae13.yndx.net (87.250.239.104) 39.491 ms
9 yandex.ru (77.88.55.88) 39.063 ms 39.135 ms 38.397 ms

```

В полученном результате трассировки до указанного узла мы можем увидеть 9 строк, которые называются *прыжками*.

ПРИМЕЧАНИЕ

Максимальное количество прыжков не должно превышать 30. Это максимальное значение, превысив которое, пакет может затеряться и не дойти до нужного узла.

В первой строке пакет обращается к маршрутизатору, который имеет адрес 192.168.1.230. Этот маршрутизатор подключен к Интернету. После получения запроса маршрутизатор отправляет пакет на IP-адрес 10.254.253.253, принадлежащий провайдеру. Затем пакет проходит несколько цепочек: от 3 и заканчивая 9. Лишь только на 9-й цепочке пакет доходит до указанного узла.

Таким способом мы можем проследить путь следования пакета, а в случае разрыва соединения с Интернетом можно увидеть, на какой цепочке пакет затерялся и не дошел до узла назначения.

Рассмотрим следующий пример:

```
tracert google.com
```

Обратите внимание на полученный результат (листинг 27.3).

Листинг 27.3. Трассировка узла google.com

```


tracert to google.com (188.43.69.98), 30 hops max, 60 byte packets
 1 192.168.1.230 (192.168.1.230) 0.265 ms 0.302 ms 0.378 ms
 2 10.254.253.253 (10.254.253.253) 1.779 ms 1.774 ms 1.765 ms
 3 mx480.omkc.ru (217.25.208.193) 1.845 ms 2.154 ms 2.149 ms
 4 rt1.omkc.ru (217.25.208.157) 2.166 ms 2.181 ms 2.192 ms
 5 omk02.transtelecom.net (188.43.2.66) 3.096 ms 3.097 ms 3.122 ms
 6 * * *
...
30 * * *

```

ПРИМЕЧАНИЕ

Полный список параметров команды `tracert` вы найдете в справочной информации, выполнив команду `man tracert`.

Как видите, пакет потерялся после 5-го прыжка. Символы астериска (звездочки) на 6–30 прыжках сообщают об отсутствии ответа на запрос. В данном случае в этом нет ничего страшного, т. к. указанный узел доступен и отвечает на ICMP-запросы (рис. 27.4).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ ping -c 4 google.com  
PING google.com (188.43.69.93) 56(84) bytes of data:  
64 bytes from cache.google.com (188.43.69.93): icmp_seq=1 ttl=58 time=8.95 ms  
64 bytes from cache.google.com (188.43.69.93): icmp_seq=2 ttl=58 time=8.91 ms  
64 bytes from cache.google.com (188.43.69.93): icmp_seq=3 ttl=58 time=8.90 ms  
64 bytes from cache.google.com (188.43.69.93): icmp_seq=4 ttl=58 time=8.85 ms  
  
--- google.com ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3005ms  
rtt min/avg/max/mdev = 8.857/8.908/8.955/0.100 ms  
sergey@Test-PC:~$
```

Рис. 27.4. Удаленный узел доступен и отвечает на ICMP-запросы

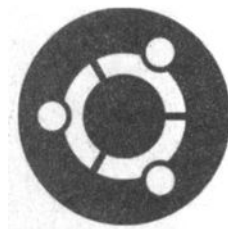
Причина поведения, когда один из узлов не отвечает, кроется в том, что он отклоняет запрос в соответствии с заданными ему настройками. Обычно это делается в целях безопасности, закрыв неиспользуемые порты и отключив ненужные службы.

Получается, что программа `traceroute` не всегда способна показать истинную картину, поэтому приходится использовать как трассировку маршрута командой `traceroute`, так и запрос доступности узла с помощью команды `ping`. Однако трассировка маршрута бывает полезной в том случае, когда, к примеру, пакет, дойдя до вашего маршрутизатора, вдруг теряется, то можно предполагать, что произошел разрыв сети на линии провайдера. В этом случае вам остается звонить в службу техподдержки и ждать, пока специалисты устранят проблему.

Выводы

Мы познакомились с инструментами, которые могут быть полезными при диагностике неполадок в сети. Мы научились проверять работоспособность сетевой карты, а также доступность удаленного узла с помощью команды `ping`. Мы освоили трассировку маршрута с помощью программы `traceroute`. Полученные нами знания будут полезны при определении источника проблемы в многоуровневой топологии сети.

ГЛАВА 28



Подключение к удаленному рабочему столу

Удаленный рабочий стол

О том, что такое *удаленный рабочий стол*, многие знают не понаслышке. Особенно это знакомо тем пользователям, которые часто обращаются в службу техподдержки или к своему другу за помощью в решении проблем, возникающих в процессе работы за компьютером. Выглядит это так: вы звоните в службу техподдержки, а специалист просит вас установить программу удаленного доступа и принять входящее подключение к рабочему столу. При этом вы можете наблюдать за тем, как сотрудник службы техподдержки удаленно настраивает ваш компьютер, точнее, его программное обеспечение. Отсюда и появилось понятие *удаленного рабочего стола*.

Программ для подключения к удаленному рабочему столу существует великое множество, и любая по-своему хороша и имеет свои недостатки. Рассматривать каждую программу в отдельности мы не будем, т. к. принцип их настройки и работы очень похож. Сейчас мы обсудим тему подключения к рабочему столу Windows из ОС Ubuntu Linux.

Подключение к рабочему столу Windows

Для подключения к удаленному рабочему столу Windows из ОС Ubuntu часто используется программа Remmina.

Remmina — это программа для ОС Ubuntu, предназначенная для подключения к удаленному рабочему столу и поддерживающая протоколы RDP, VNC, NX, XDMCP и SSH.

Программа доступна для установки как в Менеджере приложений Ubuntu, так с помощью следующей команды:

```
sudo apt install remmina
```

Установите Remmina любым удобным для вас способом и запустите приложение (рис. 28.1).

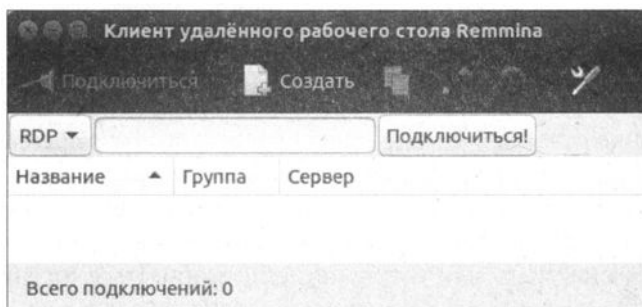


Рис. 28.1. Клиент удаленного рабочего стола Remmina

Для создания подключения к удаленному рабочему столу Windows нажмите кнопку **Создать**. Откроется окно создания нового профиля (рис. 28.2).

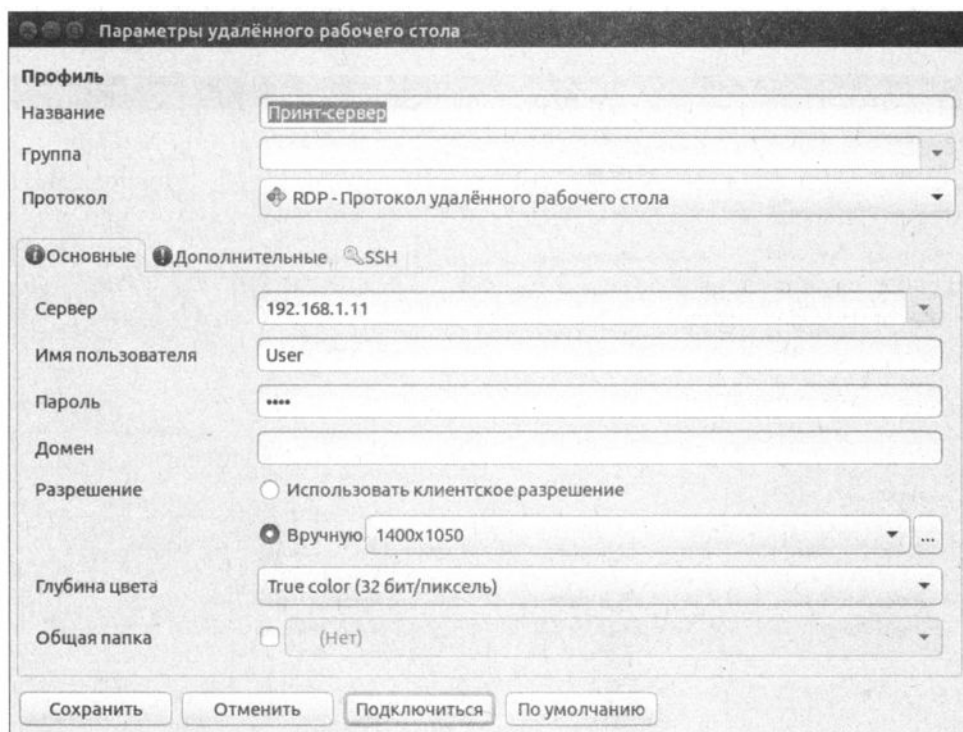


Рис. 28.2. Создание нового профиля для подключения

В поле **Название** введите имя вашего подключения. Название может быть любым и нужно только для удобства. Поле ввода **Группа** можно оставить пустым, т. к. в данный момент подключений у нас не так много, чтобы их группировать.

Поскольку мы подключаемся к удаленной машине с ОС Windows, то в раскрывающемся списке нужно выбрать протокол **RDP - Протокол удалённого рабочего стола**. Далее в поле ввода **Сервер** необходимо ввести IP-адрес удаленной машины. В нашем случае удаленный компьютер находится в локальной сети, поэтому мы

вводим соответствующий IP-адрес. Затем нужно ввести имя учетной записи в поле ввода **Имя пользователя**, а в поле **Пароль** — ее пароль.

Далее требуется настроить разрешение экрана. По умолчанию переключатель установлен в положение режима использования клиентского разрешения. Если оставить выбранный режим, то будет использовано то разрешение экрана, которое установлено на удаленном компьютере, а оно не всегда совпадает с нашим. Так, если на удаленной машине установлено низкое разрешение, а у нас имеется монитор с большим разрешением, то нам будет неудобно работать с низким разрешением. Именно поэтому рекомендуется установить переключатель в положение **Вручную** и в выпадающем списке выбрать то разрешение, которое задано на вашем компьютере. После этого для параметра **Глубина цвета** выберите режим **True color (32 бит/пиксель)**.

На вкладке **Дополнительные** рекомендуется настроить качество "картинки" и звука (рис. 28.3).

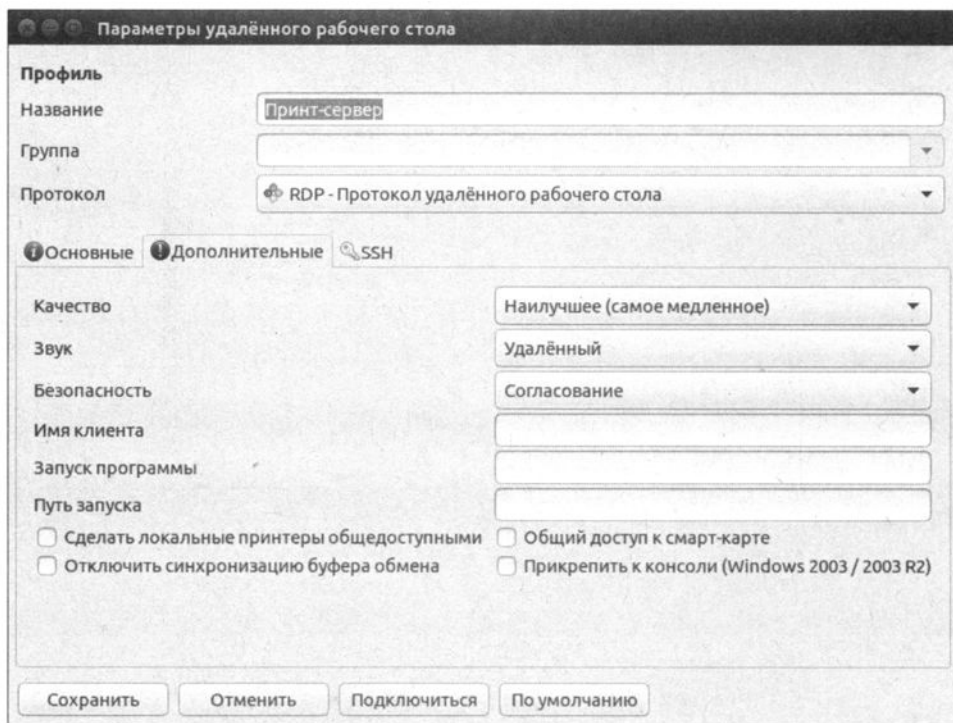


Рис. 28.3. Настройка параметров звука и отображения

В выпадающем списке выбора **Качество** задайте качество вывода изображения. Если удаленный компьютер находится в локальной сети, то имеет смысл выбрать пункт **Наилучшее (самое медленное)**. В списке выбора **Звук** выберите пункт **Удалённый**, тогда вы будете слышать то же, что и удаленный пользователь.

После задания минимальных настроек нажмите кнопку **Сохранить**.

ПРИМЕЧАНИЕ

В случае если вы подключаетесь к рабочему столу, где требуется работать с важными данными, то рекомендуется настроить параметры шифрования передаваемых данных на вкладке **SSH**. Иначе весь передаваемый трафик по сети будет доступен для прослушивания специализированными программами.

Перед тем как подключиться к удаленному рабочему столу ОС Windows, нужно разрешить входящее подключение. Делается это на вкладке **Удаленный доступ** окна **Свойства системы** (рис. 28.4).

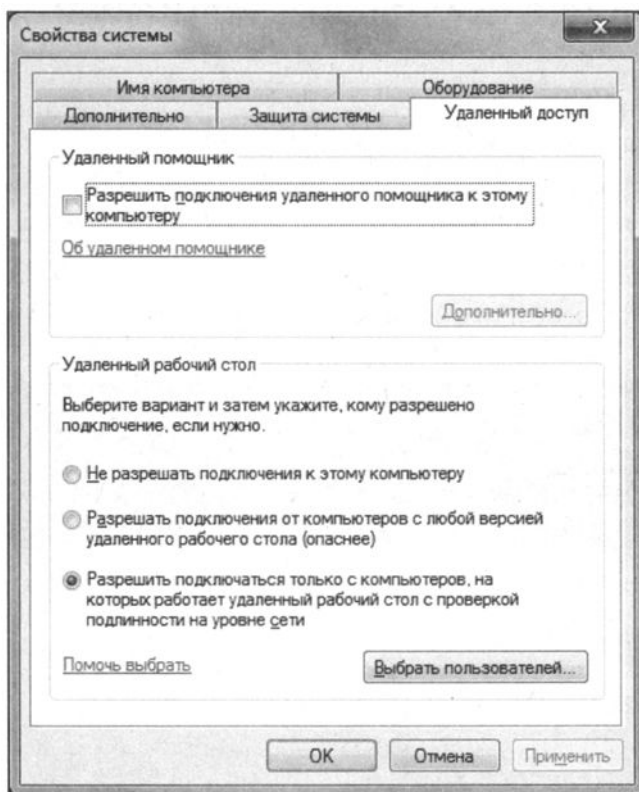


Рис. 28.4. Окно настроек удаленного доступа в ОС Windows 7

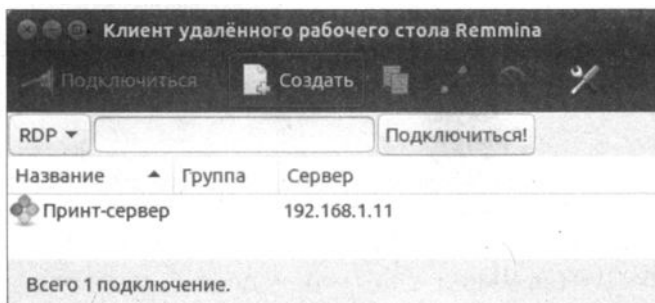


Рис. 28.5. Окно программы Remmina с созданным подключением

Убедитесь в том, что подключение к удаленному рабочему столу разрешено. Только после этого вы сможете подключиться к удаленному компьютеру.

Итак, после того как все предварительные настройки сделаны, вы можете подключиться к удаленному компьютеру в главном окне программы Remmina (рис. 28.5).

Для быстрого подключения просто два раза щелкните по созданному подключению, и удаленный рабочий стол будет открыт.

Подключение с помощью TeamViewer

TeamViewer — это программа для подключения к удаленному рабочему столу, поддерживаемая множеством операционных систем, а также имеющая огромный функционал по управлению удаленным компьютером.

Программа доступна для установки в Менеджере приложений Ubuntu. Кроме того, вы можете установить TeamViewer с помощью следующей команды:

```
sudo apt install teamviewer
```

После установки откроется главное окно программы TeamViewer (рис. 28.6).

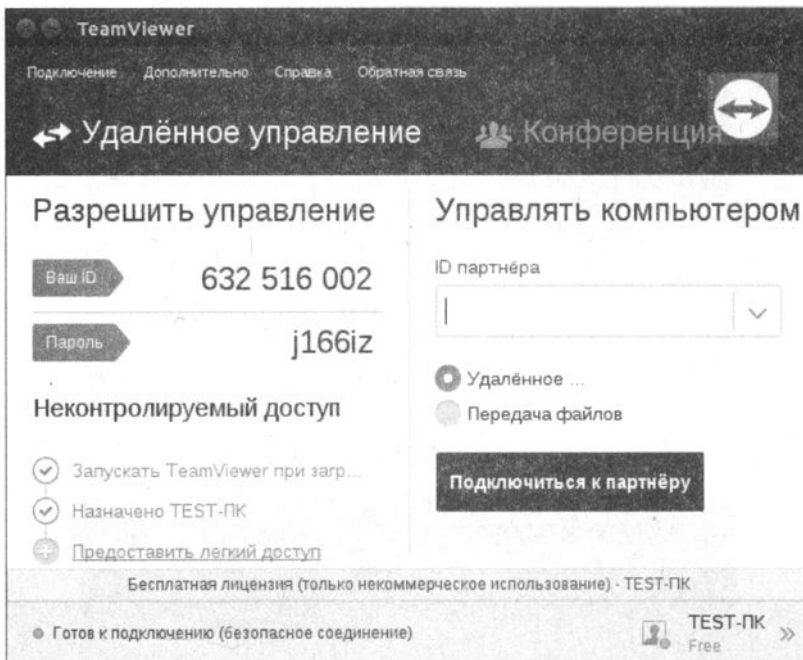


Рис. 28.6. Главное окно программы TeamViewer

Особенность работы TeamViewer в том, что установка данной программы должна быть произведена на обоих компьютерах: на вашем, с которого будет производиться удаленный доступ, и на удаленном, к которому вы планируете подключаться.

Подключение к удаленному рабочему столу очень простое: вы просто вводите ID удаленного компьютера, который указан в левой части главного окна программы, нажимаете кнопку **Подключиться к партнёру** и указываете пароль, который также выводится в левой части главного окна удаленного компьютера.

ПРИМЕЧАНИЕ

В левой части главного окна программы указан ваш ID и пароль для подключения. Это значит, что, введя эти данные, к вашему компьютеру можно будет получить удаленный доступ. Для того чтобы получить удаленный доступ к другому компьютеру, вам нужно знать ID и пароль того компьютера, на котором установлена программа TeamViewer.

Программа TeamViewer бесплатна только для некоммерческого использования. Это означает, что вы вправе запускать ее только для личных целей. Для использования TeamViewer в коммерческих целях вам придется приобрести лицензию.

Если вы часто подключаетесь к одному и тому же компьютеру, то в настройках программы можете создать постоянный пароль для подключения (рис. 28.7).

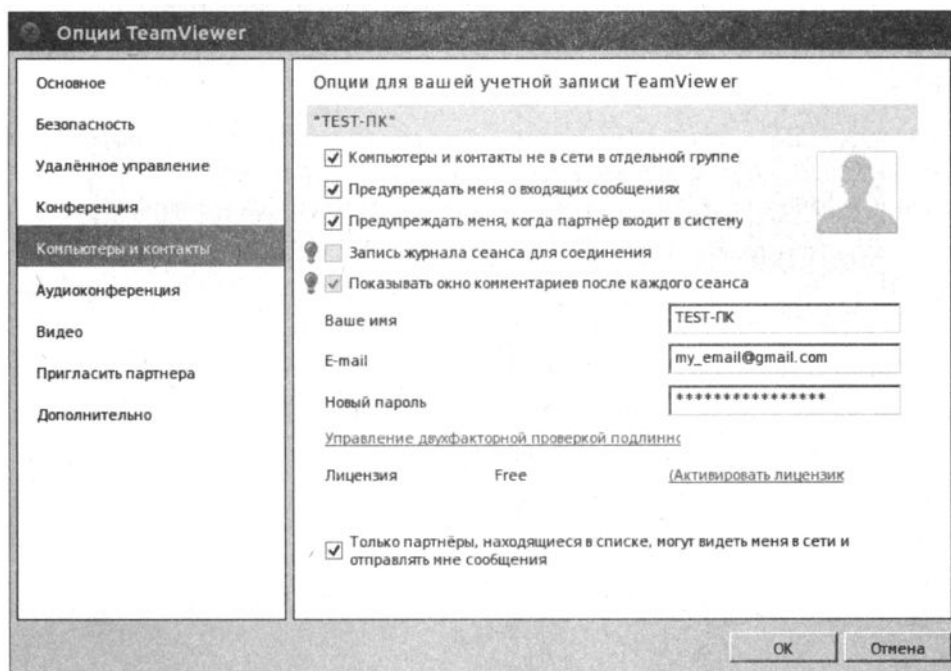


Рис. 28.7. Настройка постоянного пароля для удаленного доступа

После этого вам не потребуется постоянно спрашивать ID и пароль для доступа к удаленному компьютеру, а нужно будет только ввести указанные данные в списке разрешенных подключений окна программы TeamViewer (рис. 28.8).

Программа TeamViewer удобна тем, что позволяет подключаться не только к ОС Windows, но также к ОС Linux. Кроме того, вы с такой же легкостью можете управлять удаленным рабочим столом с помощью вашего смартфона.

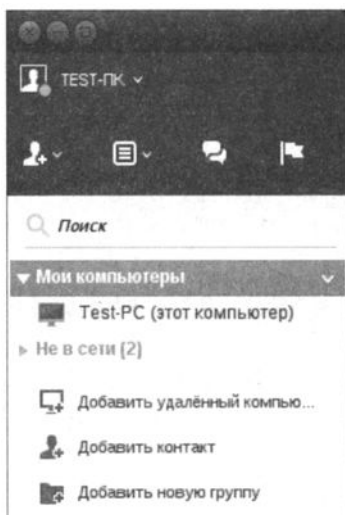


Рис. 28.8. Список разрешенных подключений

Выводы

Мы изучили весьма полезную тему подключения к удаленному рабочему столу. В данной главе мы рассмотрели два наиболее простых варианта подключения к удаленным рабочим столам: с помощью программ Remmina и TeamViewer.

Знания, полученные в этой главе, помогут нам в решении возникших проблем у неопытных пользователей, а также в том случае, когда нужно получить удаленный доступ к своему же компьютеру.

ГЛАВА 29



Восстановление загрузчика GRUB

Сбой в работе загрузчика

ОС Ubuntu является стабильно работающей системой по сравнению с ОС Windows, но тем не менее от сбоев никто не застрахован.

В начале знакомства с ОС Ubuntu мы говорили о том, что если вы устанавливаете Ubuntu Linux на один жесткий диск с ОС Windows, то нужно соблюдать порядок установки: первой устанавливается ОС Windows, а только затем ОС Ubuntu. Делается это по той причине, что ОС Windows монопольно ведет себя по отношению к другим загрузчикам и заменяет "незнакомый" загрузчик своим, делая тем самым невозможным запуск ОС Ubuntu.

Справедливости ради нужно отметить, что не всегда ОС Windows является причиной сбоев в работе загрузчика GRUB, и предугадать каждый сбой просто невозможно. Поэтому важно научиться восстанавливать работу загрузчика, благо это возможно и не является непосильной задачей. Перейдем к делу.

Восстановление загрузчика GRUB

Стоит отметить, что существует несколько способов восстановления загрузчика GRUB, которые схожи. Мы рассмотрим способ восстановления загрузчика с помощью временной смены текущего корневого каталога.

Итак, если ОС Ubuntu Linux перестала запускаться, приготовьте загрузочный диск или USB с ОС Ubuntu и загрузитесь в Live-режим (рис. 29.1).

Теперь, когда вы запустили операционную систему в Live-режиме, откройте Терминал и выполните следующую команду:

```
sudo fdisk -l
```

Данная команда выведет список разделов, которые присутствуют на жестком диске вашего компьютера (рис. 29.2).

Вспоминаем, на каком разделе у нас находится корневая файловая система. Вспомнили? Да, под корневую файловую систему мы отводили раздел /dev/sda6. Теперь,

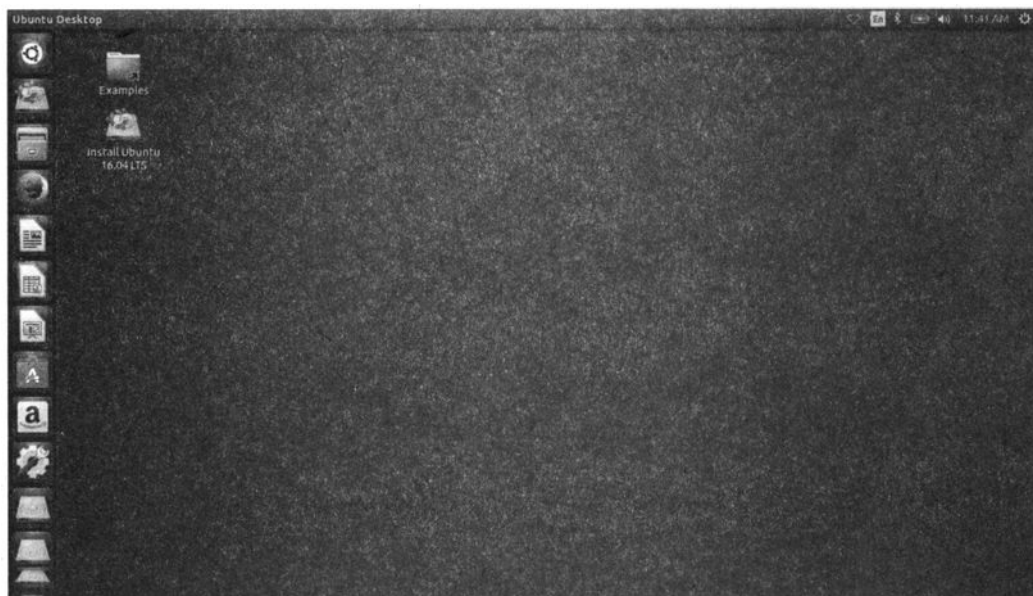


Рис. 29.1. ОС Ubuntu загрузилась в Live-режиме

```
ubuntu@ubuntu: ~
I/O size (minimum/optimal): 512 bytes / 512 bytes
Тип метки диска: dos
Идентификатор диска: 0x2bd2c32a

Устр-во   Загрузочный   Start  Конец  Секторы   Size Id Тип
/dev/sda1 *      2048    206847    204800    100M  7 HPFS/NTFS/exFAT
/dev/sda2      206848  163842047  163635200  78G   7 HPFS/NTFS/exFAT
/dev/sda3     163842048  471042047  307200000  146,5G 7 HPFS/NTFS/exFAT
/dev/sda4     471044094  625141759  154097666  73,5G  5 Расширенный
/dev/sda5     471044096  480260095    9216000    4,4G  82 Linux swap / Sola
/dev/sda6     480262144  582662534  102400391  48,8G  83 Linux
/dev/sda7     582664192  625141759  42477568   20,3G  83 Linux

Диск /dev/sdb: 931,5 GiB, 1000204886016 байтов, 1953525168 секторов
Единицы измерения: секторов из 1 * 512 = 512 байтов
Размер сектора (логический/физический): 512 байт / 4096 байт
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Тип метки диска: dos
Идентификатор диска: 0x4f260adc

Устр-во   Загрузочный   Start  Конец  Секторы   Size Id Тип
/dev/sdb1      2048  1953521663  1953519616  931,5G  7 HPFS/NTFS/exFAT
ubuntu@ubuntu:~$
```

Рис. 29.2. Список разделов жесткого диска

когда мы выяснили номер раздела корневой файловой системы, нужно его смонтировать:

```
sudo mount /dev/sda6 /mnt
```

После монтирования корневого каталога нам осталось смонтировать еще несколько каталогов:

```
sudo mount --bind /dev /mnt/dev
```

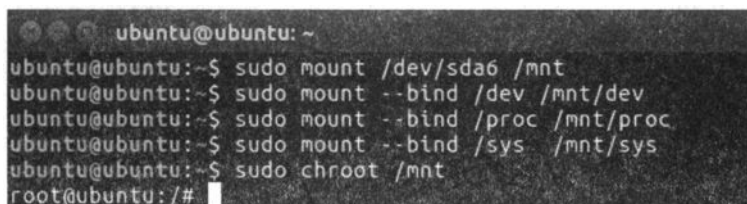
```
sudo mount --bind /proc /mnt/proc
```

```
sudo mount --bind /sys /mnt/sys
```

Важные каталоги смонтированы, осталось сделать текущим созданное окружение временной корневой файловой системы:

```
sudo chroot /mnt
```

Корневая файловая система успешно установлена (рис. 29.3).



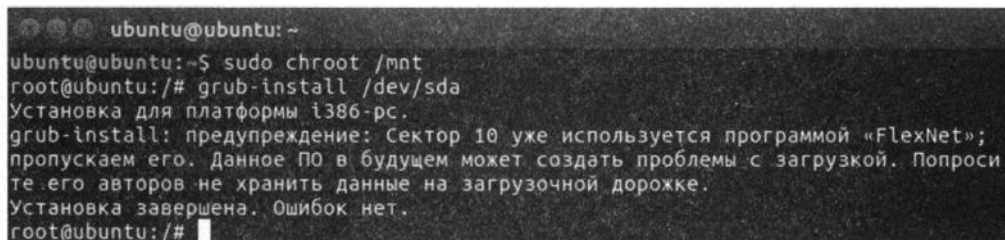
```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ sudo mount /dev/sda6 /mnt  
ubuntu@ubuntu:~$ sudo mount --bind /dev /mnt/dev  
ubuntu@ubuntu:~$ sudo mount --bind /proc /mnt/proc  
ubuntu@ubuntu:~$ sudo mount --bind /sys /mnt/sys  
ubuntu@ubuntu:~$ sudo chroot /mnt  
root@ubuntu:/#
```

Рис. 29.3. Корневая файловая система успешно установлена

Теперь нам остается произвести установку загрузчика на жесткий диск:

```
grub-install /dev/sda
```

Загрузчик успешно установлен (рис. 29.4).



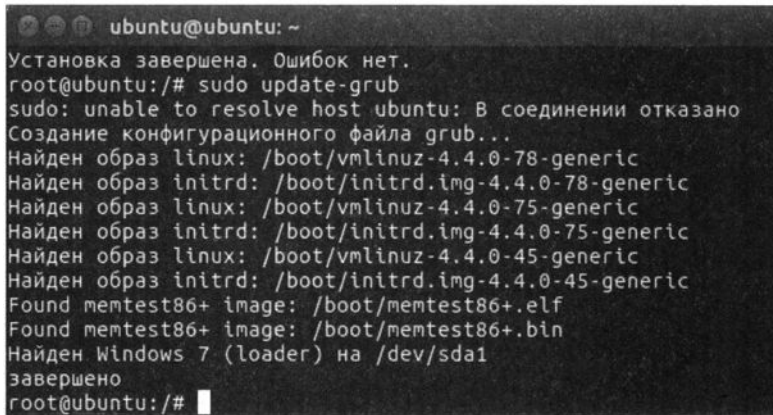
```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ sudo chroot /mnt  
root@ubuntu:/# grub-install /dev/sda  
Установка для платформы i386-pc.  
grub-install: предупреждение: Сектор 10 уже используется программой «FlexNet»;  
пропускаем его. Данное ПО в будущем может создать проблемы с загрузкой. Попро-  
сите его авторов не хранить данные на загрузочной дорожке.  
Установка завершена. Ошибок нет.  
root@ubuntu:/#
```

Рис. 29.4. Установка загрузчика успешно завершена

Сообщение "Ошибок нет" свидетельствует об успешной операции установки загрузчика. После проделанных изменений необходимо обновить меню загрузчика, выполнив следующую команду:

```
sudo update-grub
```

В ходе выполнения данной команды будут найдены образ ядра Linux и загрузчик ОС Windows 7 (рис. 29.5).



```

ubuntu@ubuntu: ~
Установка завершена. Ошибок нет.
root@ubuntu:/# sudo update-grub
sudo: unable to resolve host ubuntu: В соединении отказано
Создание конфигурационного файла grub...
Найден образ linux: /boot/vmlinuz-4.4.0-78-generic
Найден образ initrd: /boot/initrd.img-4.4.0-78-generic
Найден образ linux: /boot/vmlinuz-4.4.0-75-generic
Найден образ initrd: /boot/initrd.img-4.4.0-75-generic
Найден образ linux: /boot/vmlinuz-4.4.0-45-generic
Найден образ initrd: /boot/initrd.img-4.4.0-45-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
Найден Windows 7 (loader) на /dev/sda1
завершено
root@ubuntu:/#

```

Рис. 29.5. Найден образ ядра ОС Linux и ОС Windows 7

Теперь, после того как обновлен загрузчик, мы можем выйти из созданного временного окружения:

```
exit
```

Осталось размонтировать созданные временные каталоги:

```

sudo umount /mnt/dev
sudo umount /mnt/proc
sudo umount /mnt/sys

```

Не забываем размонтировать раздел с корневой файловой системой:

```
sudo umount /mnt
```

Для корректного завершения операции по восстановлению загрузчика перезагрузим компьютер следующей командой:

```
sudo reboot
```

Операционная система предложит извлечь установочный диск и нажать клавишу <Enter>. После этого вы сможете продолжить пользоваться своей операционной системой.

Простой способ восстановления загрузчика GRUB

Данный способ восстановления загрузчика GRUB применяется в том случае, когда вы имеете возможность загрузиться в ОС Ubuntu Linux, а ОС Windows отказывается загрузиться.

Итак, для восстановления загрузчика GRUB загружаемся в Live-режиме и открываем Терминал. Первым делом выясняем, на каком разделе находится корневая файловая система ОС Ubuntu:

```
sudo fdisk -l
```

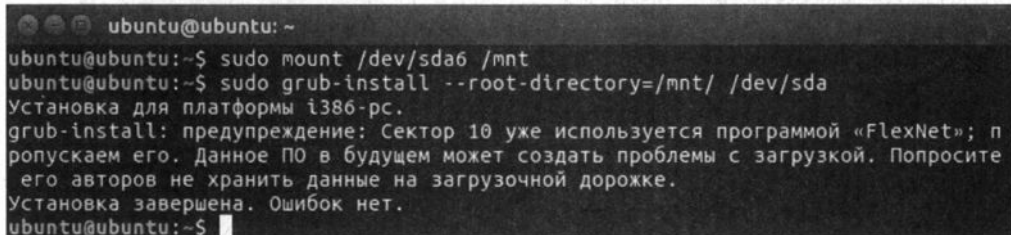
После этого монтируем раздел с корневой файловой системой:

```
sudo mount /dev/sda6 /mnt
```

Корневая файловая система смонтирована, устанавливаем загрузчик:

```
sudo grub-install --root-directory=/mnt/ /dev/sda
```

Сообщение об отсутствии ошибок говорит о том, что загрузчик успешно установлен (рис. 29.6).



```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ sudo mount /dev/sda6 /mnt  
ubuntu@ubuntu:~$ sudo grub-install --root-directory=/mnt/ /dev/sda  
Установка для платформы i386-pc.  
grub-install: предупреждение: Сектор 10 уже используется программой «FlexNet»; п  
ропускаем его. Данное ПО в будущем может создать проблемы с загрузкой. Попросите  
его авторов не хранить данные на загрузочной дорожке.  
Установка завершена. Ошибок нет.  
ubuntu@ubuntu:~$
```

Рис. 29.6. Успешная установка загрузчика GRUB

Размонтируем раздел с корневой файловой системой:

```
sudo umount /dev/sda6
```

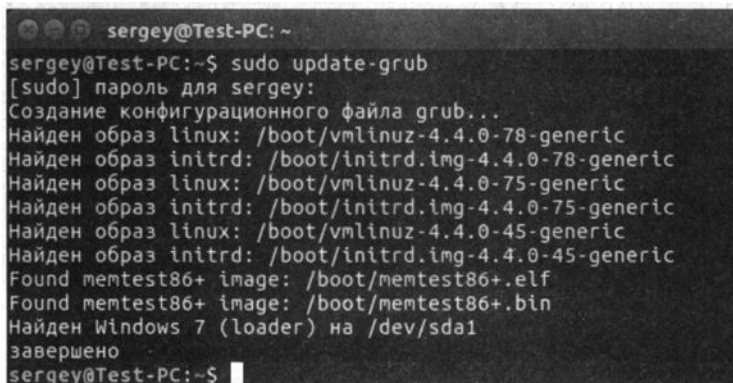
Выполняем перезагрузку уже знакомым нам способом:

```
sudo reboot
```

После того как ОС Ubuntu загрузилась в обычном режиме, запускаем Терминал и выполняем обновление меню загрузчика:

```
sudo update-grub
```

Меню загрузчика успешно обновлено (рис. 29.7), и вы можете продолжить пользо-ваться как ОС Ubuntu, так и ОС Windows 7.



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ sudo update-grub  
[sudo] пароль для sergey:  
Создание конфигурационного файла grub...  
Найден образ linux: /boot/vmlinuz-4.4.0-78-generic  
Найден образ initrd: /boot/initrd.img-4.4.0-78-generic  
Найден образ linux: /boot/vmlinuz-4.4.0-75-generic  
Найден образ initrd: /boot/initrd.img-4.4.0-75-generic  
Найден образ linux: /boot/vmlinuz-4.4.0-45-generic  
Найден образ initrd: /boot/initrd.img-4.4.0-45-generic  
Found memtest86+ image: /boot/memtest86+.elf  
Found memtest86+ image: /boot/memtest86+.bin  
Найден Windows 7 (loader) на /dev/sda1  
завершено  
sergey@Test-PC:~$
```

Рис. 29.7. Успешное обновление меню загрузчика GRUB

Восстановление GRUB в консоли загрузчика

Чуть выше мы рассмотрели два способа, когда восстанавливали загрузчик GRUB, загрузившись в Live-режиме. Существует еще способ, предусматривающий восстановление загрузчика в консоли самого загрузчика GRUB. Данный вариант полезен в том случае, когда не удастся загрузиться ни в ОС Ubuntu, ни в ОС Windows. Давайте приступим к рассмотрению этого способа восстановления загрузчика.

Итак, когда появится меню загрузчика GRUB, нажмите клавишу <с>. Так вы запустите консоль загрузчика (рис. 29.8).

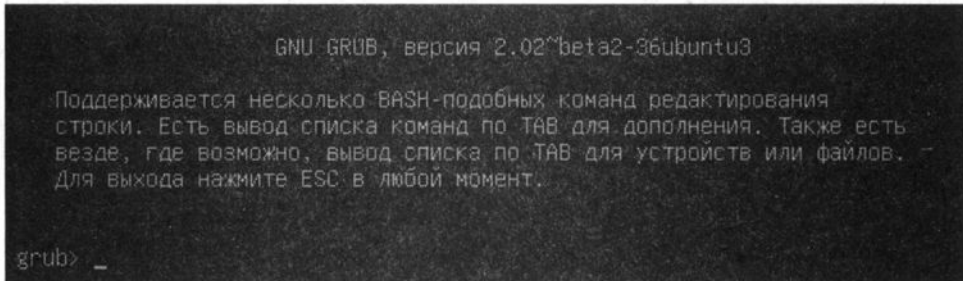


Рис. 29.8. Режим командной строки в загрузчике GRUB

Теперь нам нужно получить список разделов (рис. 29.9), выполнив следующую команду:

```
ls
```

ПРИМЕЧАНИЕ

Обратите внимание, что имена разделов отличаются от тех, которые дает им операционная система.

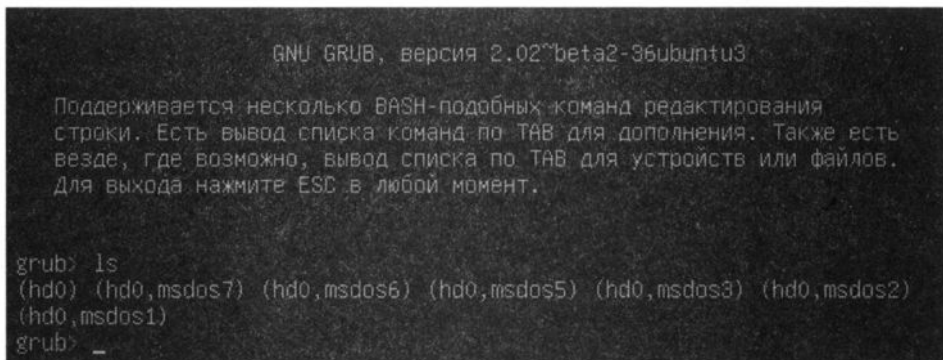


Рис. 29.9. Получены имена разделов жесткого диска

Имена жестких дисков начинаются с букв `hd`, после которых следует порядковый номер. Так, `hd0` означает первый жесткий диск, а `hd1` — второй жесткий диск. Разделы жесткого диска начинаются с указания номера жесткого диска, после которо-

го следует номер раздела, начинающийся с приставки `msdos`. После приставки `msdos` идет порядковый номер раздела, который начинается с единицы. Так, запись `(hd0, msdos5)` означает, что раздел `msdos5` находится на первом жестком диске `hd0`.

ПРИМЕЧАНИЕ

Обратите внимание, что раздел `(hd0,msdos4)` отсутствует в списке разделов, т. к. является лишь расширенным разделом, в котором перечислены все расширенные разделы.

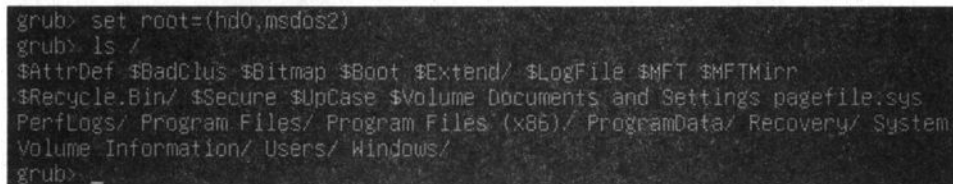
Для того чтобы понять, какой именно раздел является загрузочным, необходимо просмотреть список содержимого предполагаемого корневого раздела. Допустим, вы считаете, что раздел `(hd0, msdos2)` является корневым каталогом, и устанавливаете его в качестве корневой файловой системы:

```
set root=(hd0,msdos2)
```

Теперь просмотрим список содержимого выбранного раздела предполагаемой корневой файловой системы:

```
ls /
```

Как видите, список содержимого ничем не напоминает корневую файловую систему ОС Ubuntu Linux, следовательно, данный раздел не может являться разделом с корневой файловой системой (рис. 29.10).



```
grub> set root=(hd0,msdos2)
grub> ls /
$AttrDef $BadClus $Bitmap $Boot $Extend/ $LogFile $MFT $MFTMirr
$Recycle.Bin/ $Secure $UpCase $Volume Documents and Settings pagefile.sys
PerfLogs/ Program Files/ Program Files (x86)/ ProgramData/ Recovery/ System
Volume Information/ Users/ Windows/
grub> _
```

Рис. 29.10. Просмотр содержимого предполагаемого корневого каталога

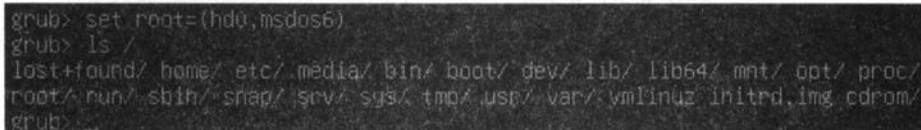
Теперь попробуем задать раздел `(hd0, msdos6)` в качестве корневой файловой системы:

```
set root=(hd0,msdos6)
```

Проверим список содержимого файловой системы:

```
ls /
```

Судя по списку содержимого данного раздела, мы можем с уверенностью сказать, что это и есть искомый раздел с корневой файловой системой (рис. 29.11).



```
grub> set root=(hd0,msdos6)
grub> ls /
lost+found/ home/ etc/ media/ bin/ boot/ dev/ lib/ lib64/ mnt/ opt/ proc/
root/ run/ /sbin/ snap/ srv/ sys/ tmp/ usr/ var/ vmlinuz initrd.img cdrom/
grub> _
```

Рис. 29.11. Найден раздел с корневой файловой системой ОС Ubuntu

Загрузим ядро ОС Ubuntu Linux следующей командой:

```
linux /vmlinuz root=/dev/sda6
```

После этого определим `initrd` (от англ. *Initial RAM Disk* — диск в оперативной памяти для начальной инициализации):

```
initrd /initrd.img
```

Теперь нам остается только загрузить операционную систему, выполнив следующую команду загрузки ОС Ubuntu Linux:

```
boot
```

Начнется загрузка ОС Ubuntu, и ваша система будет опять в работоспособном состоянии.

Выводы

Мы рассмотрели ситуацию, когда из-за сбоев в работе загрузчика GRUB отказывалась запускаться операционная система. Мы научились восстанавливать работу данного загрузчика в Live-режиме операционной системы Ubuntu Linux, а также обновлять меню загрузчика в штатном режиме работы этой ОС.

Мы научились восстанавливать работоспособность загрузчика в командной строке загрузчика.

Полученные знания в ходе изучения данной главы помогут нам оперативно устранять неисправности, возникающие при работе с операционной системой Ubuntu Linux.

ГЛАВА 30



Увеличение разрешения ОС Ubuntu в VirtualBox

ОС Ubuntu в виртуальной машине

Очень часто при знакомстве с ОС Ubuntu и во время экспериментов с ней установку операционной системы производят в виртуальной машине Oracle VM VirtualBox.

Начинающие пользователи боятся устанавливать незнакомую операционную систему на основную рабочую станцию. Набившие шишки, уже более опытные пользователи могут поделиться своим печальным опытом потери важных данных из-за неправильной разметки жесткого диска. Немаловажным аргументом в пользу выбора установки новой операционной системы в виртуальной машине является то, что не все пользователи могут позволить себе еще один компьютер ради изучения новой операционной системы.

По этой и по многим другим причинам очень часто выручает так называемая виртуальная машина от компании Oracle под названием VM VirtualBox. Установив виртуальную машину, вы сможете экспериментировать в системе и не бояться потери данных.

Виртуальная машина Oracle VM VirtualBox

Виртуальная машина — программная система, которая позволяет эмулировать аппаратное обеспечение компьютера и тем самым устанавливать в данной среде любую совместимую с ней операционную систему.

На данный момент существует множество виртуальных машин: VirtualBox, Virtual Iron, VMware Workstation, Windows Virtual PC и т. д. Однако наиболее распространенной является VM VirtualBox.

Установка ОС Ubuntu в виртуальной машине ничем не отличается от установки операционной системы на обычный компьютер, поэтому отдельно мы не будем затрагивать эту тему, а предположим, что у вас уже установлена ОС в вашей виртуальной машине Oracle VM VirtualBox.

Низкое разрешение экрана в виртуальной машине

Загрузив ОС Ubuntu и попытавшись настроить разрешение экрана, мы можем увидеть следующую картину (рис. 30.1).

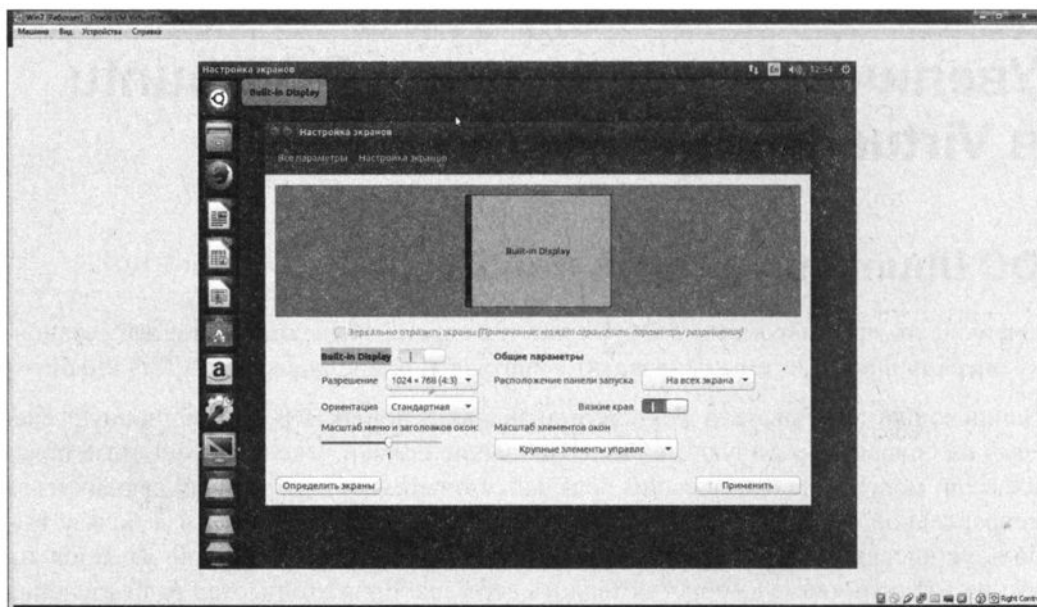


Рис. 30.1. Низкое разрешение экрана

Максимально доступное разрешение экрана составляет всего лишь 1024×768 пикселей. Это означает, что вы не сможете увеличить разрешение выше этого значения, а часть экрана остается незадействованной.

Разумеется, работать в системе с низким разрешением — не лучший вариант. И многие пользователи, сталкиваясь с таким неудобством, бросают изучение мира Linux, оставаясь на дружелюбной к пользователям ОС Windows. Однако, как вы уже знаете, выход есть из любой ситуации, и даже из этой. Поэтому не стоит бросать изучение ОС Ubuntu Linux, а имеет смысл решить проблему с низким разрешением.

Установка дополнений

Многим из нас сейчас пришло в голову попытаться установить видеодрайвер, но это неверное решение, и даже не стоит тратить на это время. Проблема кроется в другом — в самой настройке виртуальной машины Oracle VM VirtualBox.

Сперва нам нужно подключить образ диска дополнений для нашей виртуальной машины. Для этого в панели меню виртуальной машины Oracle VM VirtualBox вы-

берите пункт меню **Устройства**, а затем пункт **Подключить образ диска Дополнений гостевой ОС....** Откроется окно автоматического запуска приложения (рис. 30.2).

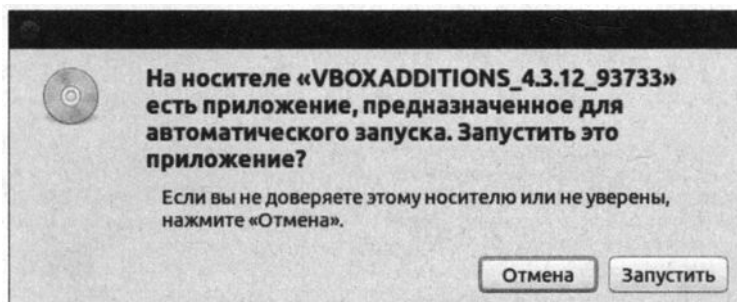


Рис. 30.2. Окно автоматического запуска приложения

Нажмите кнопку **Запустить**, на запрос аутентификации введите свой пароль учетной записи и нажмите кнопку **Аутентифицировать** (рис. 30.3).

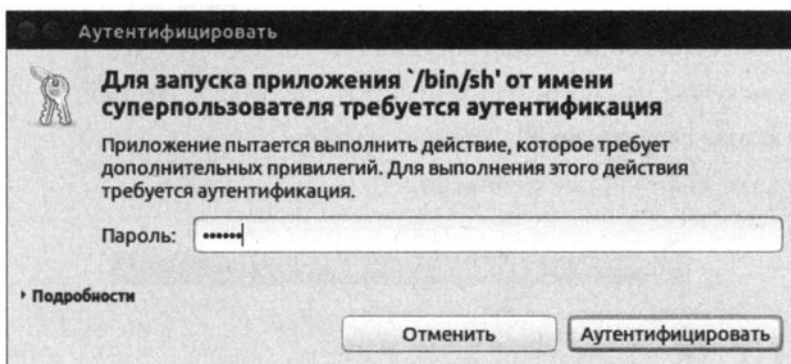


Рис. 30.3. Запрос аутентификации

В ответ на это откроется окно Терминала и начнется установка дополнений гостевой операционной системы (рис. 30.4).

По окончании процесса установки нажмите клавишу <Enter>, чтобы закрыть окно Терминала.

ПРИМЕЧАНИЕ

Об успешном окончании процесса установки свидетельствует сообщение в Терминале "Press Return to close this window..."

Устанавливаемые дополнения являются обычным файлом VBoxGuestAdditions.iso, который расположен в каталоге C:\Program Files\Oracle\VirtualBox\.

После установки дополнения гостевой операционной системы откройте Терминал и выполните следующую команду:

```
sudo xdiagnose
```

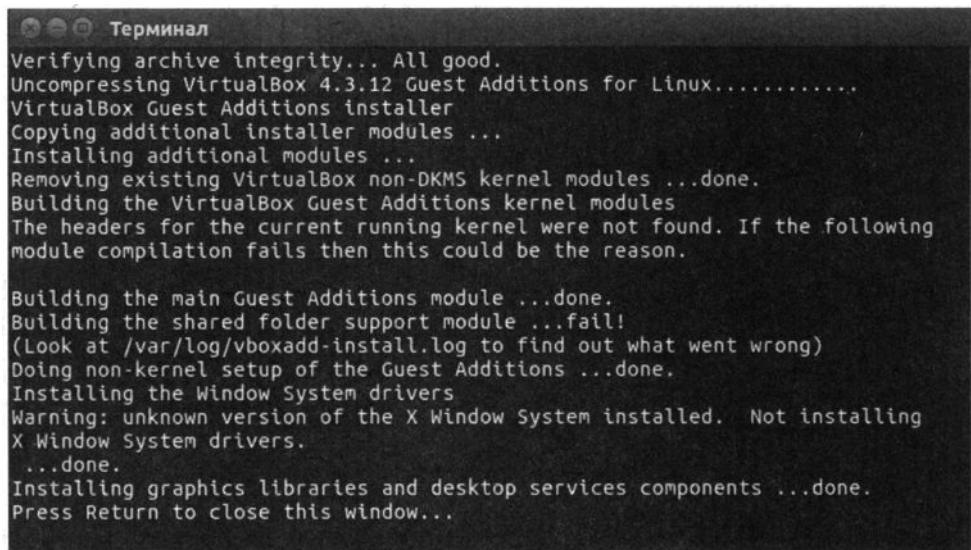



Рис. 30.4. Установка дополнений гостевой операционной системы

Откроется окно **Настройки X-диагностики** (рис. 30.5).

В открывшемся окне нам потребуется отметить следующие флажки:

- ◆ **Специальные сообщения об отладке графики;**
- ◆ **Отображать загрузочные сообщения.**

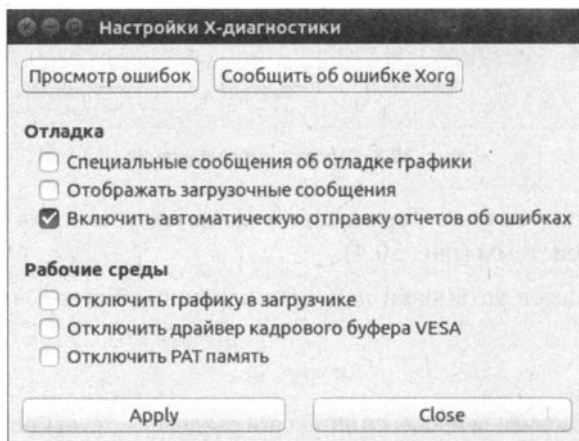


Рис. 30.5. Окно настройки X-диагностики

Нажмите кнопку **Apply** и дождитесь, пока в Терминале появится сообщение об успешном обновлении загрузчика (рис. 30.6).

Теперь можете закрыть окно **Настройки X-диагностики**, нажав кнопку **Close**, а затем закрыть окно Терминала.

Это еще не все. Осталось задействовать установленное дополнение. Откройте окно **Параметры системы** и в разделе **Система** выберите **Программы и обновления**. В одноименном открывшемся окне перейдите на вкладку **Дополнительные драйверы**, установите переключатель **Используется x86 virtualization solution - X11 guest utilities** из **virtualbox-guest-X11** (проприетарное) и нажмите кнопку **Применить изменения** (рис. 30.7).

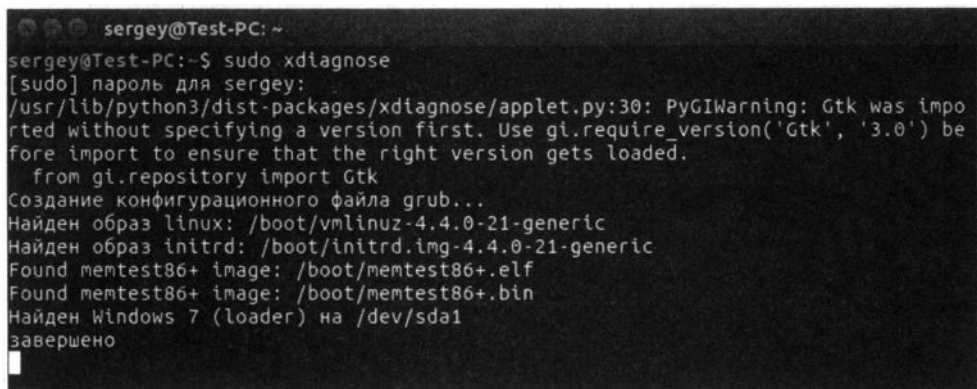


Рис. 30.6. Загрузчик GRUB успешно обновлен

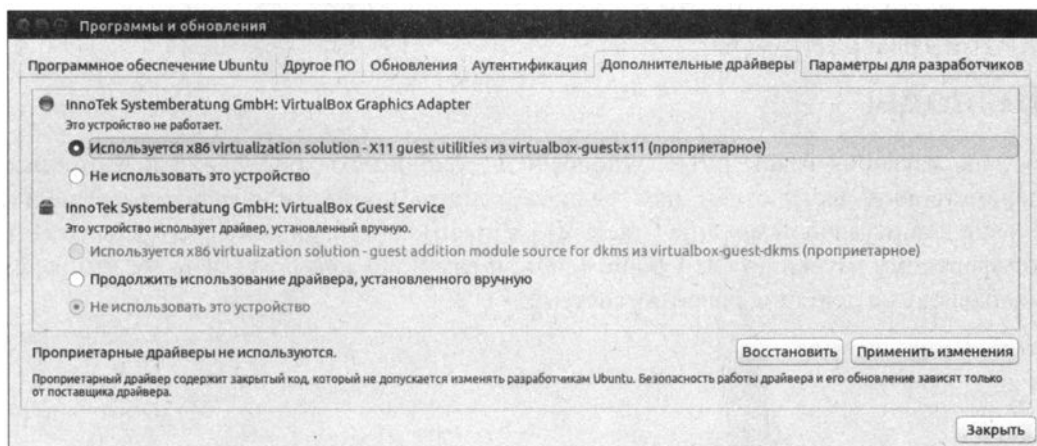


Рис. 30.7. Применение изменений виртуального графического адаптера

После ввода пароля в окне аутентификации пользователя и успешного сохранения изменений нажмите кнопку **Заккрыть** окна **Программы и обновления** и перезагрузите операционную систему.

Теперь вы сможете сменить разрешение на более подходящее для вашего монитора (рис. 30.8).

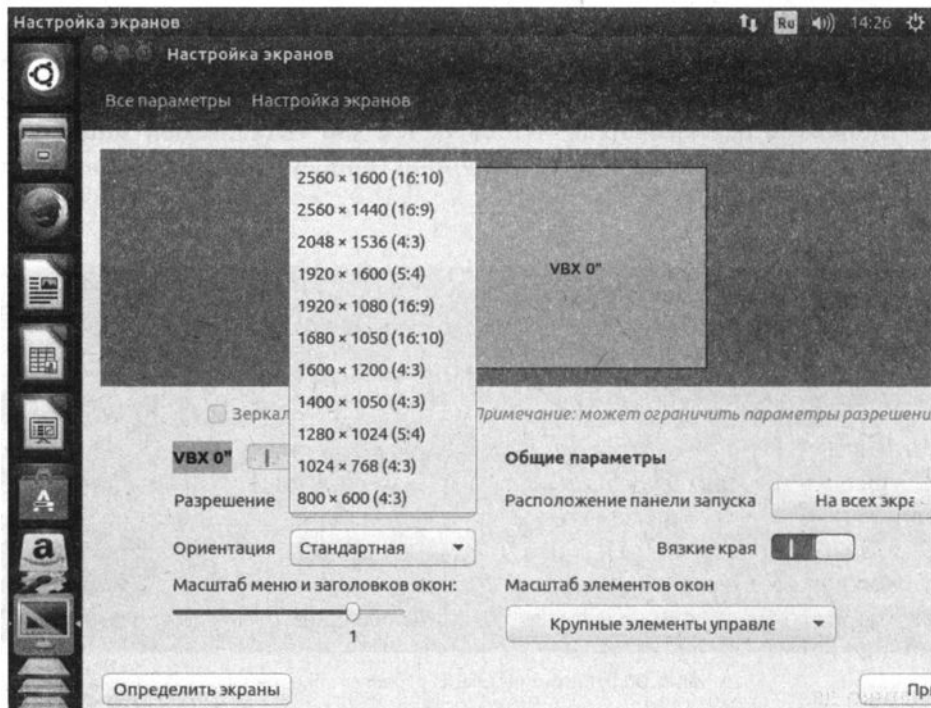


Рис. 30.8. Доступное разрешение виртуальной машины

Выводы

В ходе изучения главы путем установки необходимого дополнения и настройки параметров X-диагностики нам удалось решить проблему низкого разрешения экрана в виртуальной машине Oracle VM VirtualBox. Теперь ничто не препятствует комфортному изучению ОС Ubuntu Linux, и вам не нужно бояться за то, что ваши неправильные действия навредят системе.

Заключение

Вот и завершается наше знакомство с операционной системой Ubuntu Linux. Вы прошли путь от начинающего пользователя к опытному пользователю ОС Ubuntu Linux. Разумеется, описать все нюансы и тонкости операционной системы Ubuntu просто невозможно в силу того, что книга получилась бы огромной. К тому же, не познав азы, не стоит погружаться в изучение расширенных возможностей этой операционной системы.

К созданию данной книги было приложено немало труда и усилий. Автор надеется, что она помогла вам понять особенности операционной системы Ubuntu Linux, научиться свободно работать в ней и приобрести базовые знания и умения, которые помогут вам стать отличным специалистом в области Linux-подобных операционных систем.

Человеку свойственно ошибаться, и если вы нашли ошибку или опечатку в данной книге, пожалуйста, сообщите об этом автору через форму обратной связи на сайте volokh.info, либо отправив письмо на e-mail: volokhsv@yandex.ru. Автор всегда открыт для диалога, и если у вас есть какие-то замечания, пожелания или вы просто хотите поговорить на компьютерные темы, вам всегда будут рады на сайте автора.

ПРИЛОЖЕНИЕ

Описание электронного архива

По ссылке **<ftp://ftp.bhv.ru/9785977539531.zip>** можно скачать электронный архив с материалами, не вошедшими в печатную версию книги. Эта ссылка доступна также со страницы книги на сайте **www.bhv.ru**.

Содержимое электронного каталога представлено в табл. П1.

Таблица П1. Описание электронного каталога

Папки	Описание
\ Приложения	Содержит не вошедшие в книгу главы: <ul style="list-style-type: none">• "Немного истории";• "Замечания к выпуску ОС Ubuntu 17.04"
\volokh.info	Избранные статьи и заметки из блога автора volokh.info

Предметный указатель

A

ACPI 347
Advanced Linux Sound Architecture (ALSA)
219
◊ AlsaMixer 223
APN 84
Apt 251
Audacious, аудиоплеер 267

B

bash 145
BIOS 45, 333
Brasero, запись дисков 271
Btrfs 37

C

Chromium, браузер 269
Common UNIX Printing System (CUPS) 228
◊ настройка общего доступа 233
◊ установка CUPS 228
Coordinated Universal Time 105
Copyleft 23

D

Debian 20
Double Commander 270
Dpkg 248
DSL 75
◊ соединение 76
▫ настройка 82
▫ проблема с подключением 85

E

EFI 46
Ethernet 73
ext 36
ext3 36
ext4 37, 62
Extended Boot Record (EBR) 40
Extended File System 36

F

Fedora 20
Filesystem Hierarchy Standard (FHS) 120
Fourth extended file system 37
Freax 16

G

Gedit, редактор 109
General Public License 18, 23
GIMP, графический редактор 271
GNU 17–19, 23
GNU/Linux 18
GParted 50
◊ метки устройств 325
GParted Partition Editor 50
GPT 40
GRand Unified Bootloader (GRUB) 68, 109,
333
◊ восстановление загрузчика 380
GUID Partition Table 40

H

Head-Up Display (HUD) 101

I

ICMP Echo-Reply 366
ICMP Echo-Request 366
init, программа 333
Initial RAM Disk 384

J

JFS 38

L

LightDM 360
Linux 16
Linux Mint 21
Live-режим 47, 58, 69
◊ восстановление загрузчика 382
◊ восстановление загрузчика GRUB 377
Long Term Support (LTS) 30

M

Master boot record (MBR) 40, 46
Minix 16
Minix File System 36

N

Nautilus 115, 200
NetworkManager 70, 71, 73, 74, 82, 85

O

Open Sound System (OSS) 220
Oracle VM VirtualBox 385
◊ установка дополнений 386

P

Personal Packages Archive 260
Pinta, графический редактор 271
Power-On Self-Test (POST) 67
PPA 260
PPPoE-соединение 77
pppoeconf 77

Q

qBittorrent 269

R

Red Hat Enterprise Linux 20
Reiser4 38
ReiserFS 37

S

SGID 302
Slackware 19, 21
Sticky 302
sudo, утилита 143, 195
SUID 301
Systemctl 340
Systemd 336

U

Ubuntu 20
Ubuntu Desktop 30
Ubuntu Server 30
Ubuntu Tweak 98
Unified Extensible Firmware Interface (UEFI) 46
Unity 94
UNIX 17
USB-модем 82
◊ подключение USB-модема 82
User Identifier (UID) 194
UTC 105
UUID 322

V

Visudo, утилита 202
VLC, видеоплеер 268

W

Wi-Fi 71
Wine 272

X

XFS 37

А

Адаптер Wi-Fi 71
Архив 279

- ◇ несжатый 281
- ◇ пакетов персональный 108
- ◇ сжатый 281, 284

Архиватор:

- ◇ bzip2 288
- ◇ gzip 286
- ◇ tar 282

Ассоциация файлов 136
Аудиодрайвер 219

В

Видеодрайвер AMD Catalyst 217
Виртуальная машина 385

Г

Главное меню:

- ◇ Dash 100
- ◇ линзы 101

Группа пользователей 293

Д

Дерево каталогов 119
Дистрибутив 18
Драйвер дополнительный 207

З

Зависимость пакетов 242
Загрузчик GRUB 68
Запись:

- ◇ главная загрузочная 40
- ◇ учетная 141
 - root 140
 - администратора 139
 - обычная 139

И

Имя пользователя, получение 157
Индексный дескриптор inode 312

К

Карта сетевая:

- ◇ Broadcom 212
- ◇ Realtek 209

Каталог:

- ◇ /bin/ 124
- ◇ /boot/ 124
- ◇ /cdrom/ 124
- ◇ /dev/ 124
- ◇ /etc/ 125
- ◇ /home/ 125
- ◇ /lib/ 126
- ◇ /lib64/ 126
- ◇ /lost+found/ 126
- ◇ /media/ 124, 126
- ◇ /mnt/ 127
- ◇ /opt/ 127
- ◇ /proc/ 127
- ◇ /root/ 127
- ◇ /run/ 128
- ◇ /sbin/ 128
- ◇ /snap/ 128
- ◇ /srv/ 128
- ◇ /sys/ 128
- ◇ /tmp/ 129
- ◇ /usr/ 129
- ◇ /var/ 129
- ◇ копирование 166
- ◇ переименование 171
- ◇ перемещение 171
- ◇ поиск 181
- ◇ пользователя, домашний 123
- ◇ создание 162
- ◇ удаление 163, 164

Команда:

- ◇ cat 173
- ◇ cd 149
- ◇ chattr 308
- ◇ chown 308
- ◇ clear 155
- ◇ cp 166
- ◇ echo 148
- ◇ exit 156
- ◇ file 290
- ◇ find 181
- ◇ gksu 201
- ◇ grep 183
- ◇ history 155
- ◇ journalctl 348

Команда (*прод.*):

- ◊ kill 356, 358
- ◊ less 177
- ◊ ln 314
- ◊ ls 151
- ◊ lsattr 310
- ◊ man 152
- ◊ mkdir 162
- ◊ more 177
- ◊ mv 171
- ◊ pwd 150
- ◊ rm 164
- ◊ rmdir 163
- ◊ shutdown 191
- ◊ sudo 195
- ◊ tac 175
- ◊ touch 160
- ◊ umask 304
- ◊ uname 149
- ◊ which 183
- ◊ who 158
- ◊ whoami 157
- Консоль 144
- Корень файловой системы 119

Л

Линус Торвальдс 16
Лог 348

М

- Маска пользовательская 304
Менеджер архивов 280
- ◊ работа 280
- Менеджер обновлений 88
- ◊ обновление приложений 88
 - ◊ сведения об обновлениях 88
- Менеджер приложений Ubuntu 243
- ◊ удаление программ 262
 - ◊ установка:
 - BitTorrent-клиента 270
 - Google Chrome 246
 - qBittorrent 244
 - браузера 269
 - графического редактора 271
 - мультимедийных проигрывателей 267
 - программы для работы с CD/DVD-дисками 271
 - файлового менеджера 270

Меню главное 100
Метка устройства 322
Монтирование автоматическое 327

Н

Настройка X-диагностики 388

О

Операционная система, остановка 191
ОС Ubuntu Linux 29

П

- Панель запуска 97, 247
- ◊ закрепление на панели запуска 248
- Панель меню 95
- ◊ рабочий стол Ubuntu 95
- Параметры системы 103
- ◊ ввод текста 104
- Патрик Фолькердинг 19
Подключение типа Ethernet 73
Пользователь:
 - ◊ root 194
 - ◊ сведения 158
- Программа:
 - ◊ kill 356
 - ◊ ping 366
 - ◊ ps 354
 - ◊ traceroute 367

Р

- Рабочее место 98
Раздел:
 - ◊ жесткого диска 39
 - ◊ первичный 40
 - ◊ расширенный 40
- Режим спящий 107
Репозиторий 108, 258
- ◊ добавление 260
 - ◊ управление 259
- Ричард Столлман 17

С

Своп-файл *См. Файл подкачки*
Сектор загрузочный 40

Система:

- ◇ инициализации systemd 340
- ◇ файловая 36
 - корневая 123
- Системный монитор 361, 351
- Служба 339
- Ссылка 313
- ◇ жесткая 314
- ◇ символическая 313
- ◇ создание 314
- Суперпользователь root 138

Т

- Таблица разделов 40
- Текстовый редактор 202
- Терминал 72, 144
- ◇ восстановление загрузчика 380
- ◇ добавление репозитория 261
- ◇ запуск:
 - Windows-приложений 277
 - X-диагностики 387
- ◇ команды Терминала 148
- ◇ просмотр прав доступа к файлу 295
- ◇ удаление Bluefish 263
- ◇ установка:
 - Avidemux 254
 - Wine 272
 - программы Atom 262
- Тип учетной записи 142
- Точка монтирования 40, 62, 320

У

- Удаленный рабочий стол 370
- ◇ программа Remmina 370
- ◇ программа TeamViewer 374
- Уровень выполнения 333

Ф

Файл:

- ◇ /etc/systemd/journald.conf 348
- ◇ logind.conf 347
- ◇ sudoers 195, 202
- ◇ ассоциация 135
- ◇ вывод содержимого:
 - на экран 173
 - в обратной последовательности 175
- ◇ исполняемый 134
- ◇ копирование 166
- ◇ переименование 171
- ◇ перемещение 171
- ◇ подкачки, создание 53
- ◇ поиск 181
 - критерии 183
 - шаблона 183
- ◇ просмотр содержимого 177
- ◇ слияние с другим файлом 175
- ◇ создание 161
- ◇ текстовый 132
- ◇ удаление 164
- ◇ установка времени изменения 160

Ц

- Цели 336

Э

- Экранирование 150
- Экстент 311

Ю

- Юнит 337, 340

Я

- Язык системы 91
- ◇ установка и удаление языков 91

Внутреннее устройство Linux

Отдел оптовых поставок:

e-mail: opt@bhv.spb.su



- Пользовательское окружение и интерфейс командной строки CLI
- Файлы, каталоги и файловые системы
- Дискреционное, мандатное разграничение доступа и привилегии
- Процессы и нити
- Виртуальная память и отображаемые файлы
- Каналы, сокеты и разделяемая память
- Сетевая подсистема и служба SSH
- Оконная система X Window и графический интерфейс GUI
- Программирование на языке командного интерпретатора

Книга, которую вы держите в руках, адресована студентам, начинающим пользователям, программистам и системным администраторам операционной системы Linux. Она представляет собой введение во внутреннее устройство Linux — от ядра до сетевых служб и от утилит командной строки до графического интерфейса.

Все части операционной системы рассматриваются в контексте типичных задач, решаемых на практике, и поясняются при помощи соответствующего инструментария пользователя, администратора и разработчика.

Все положения наглядно проиллюстрированы примерами, разработанными и проверенными автором с целью привить читателю навыки самостоятельного исследования постоянно эволюционирующей операционной системы Linux.

Кетов Дмитрий Владимирович, ведущий преподаватель в области операционных систем и сетевых технологий Санкт-Петербургского политехнического университета (СПбПУ) с многолетним стажем, начальник отдела разработки программного обеспечения Санкт-Петербургского филиала АО “Концерн радиостроения <<Вега>>”. Профессионально занимается теорией построения и практикой разработки операционных систем и системного программного обеспечения.

Ubuntu Linux с нуля

**Ubuntu Linux — это
проще, чем кажется**

Эта книга — проводник в мир операционной системы Linux! В ней рассмотрен самый популярный дистрибутив Ubuntu Linux. С позиции пользователя описан весь цикл работы, начиная от установки и заканчивая восстановлением работы системы после сбоев. Рассмотрены вопросы подключения принтеров, сканеров и других периферийных устройств. Особое внимание уделено работе в командной строке, при этом все команды подробно раскрыты. Материал книги ориентирован на текущую версию, но будет применим и к будущим версиям Ubuntu Linux. Более того, полученных в книге знаний достаточно для перехода на любой другой дистрибутив Linux без дополнительного его изучения.



Волох Сергей Васильевич, программист, специалист в области информационных технологий, преподавал информатику в школе. Благодаря опыту работы в педагогической сфере, умеет простым и понятным языком рассказывать о сложных вещах. Ведет личный сайт <https://volokh.info>, посвященный информационным технологиям, где активно помогает пользователям в решении возникших проблем.



Электронный архив с материалами, не вошедшими в печатную версию книги, можно скачать по ссылке <ftp://ftp.bhv.ru/9785977539531.zip>, а также со страницы книги на сайте www.bhv.ru.

ISBN 978-5-9775-3953-1



БХВ-ПЕТЕРБУРГ

191036, Санкт-Петербург,
Гончарная ул., 20

Тел.: (812) 717-10-50,
339-54-17, 339-54-28

E-mail: mail@bhv.ru

Internet: www.bhv.ru

