

Сергей Волох

# Ubuntu Linux

С нуля

2-е издание



## Вы узнаете, как:

- устанавливать и настраивать Ubuntu Linux;
- работать в графическом режиме и командной строке;
- устанавливать драйверы оборудования;
- решать проблемы, возникающие при установке программ;
- подключаться к Интернету;
- находить и устранять неисправности работы сети;
- работать со службами и файловым сервером Samba;
- создавать сертификаты Let's Encrypt;
- обеспечивать бесперебойную работу системы.



Сергей Волох

# Ubuntu Linux

С нуля

2-е издание

Санкт-Петербург

«БХВ-Петербург»

2021

УДК 004.451  
ББК 32.973.26-018.2  
В68

**Волох С. В.**

В68 Ubuntu Linux с нуля. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2021. — 416 с.: ил.

ISBN 978-5-9775-6761-9

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.

Во втором издании учтены особенности и новшества последней версии ОС Ubuntu Linux, добавлены главы по созданию и установке сертификатов Let's Encrypt и настройке файлового сервера Samba.

Электронный архив на сайте издательства содержит дополнительные материалы и статьи по теме книги.

*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>Людмила Гауль</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Дизайн обложки	<i>Карины Соловьевой</i>

"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

ISBN 978-5-9775-6761-9

© ООО "БХВ", 2021  
© Оформление. ООО "БХВ-Петербург", 2021

# Оглавление

<b>Предисловие .....</b>	<b>11</b>
Для кого эта книга .....	12
Как устроена книга .....	12
Как читать эту книгу.....	13
Чем отличается новое издание от предыдущих .....	13
От автора .....	13
Благодарности.....	14
<b>Введение .....</b>	<b>15</b>
История появления ОС Linux .....	16
GNU и Linux.....	17
Появление различных дистрибутивов.....	18
Дистрибутивы Linux.....	19
Red Hat Enterprise Linux .....	19
Fedora Linux.....	20
Debian.....	20
Ubuntu .....	20
Linux Mint.....	21
Slackware.....	21
Логотип Linux .....	21
Область применения Linux .....	22
Идеология свободного программного обеспечения .....	23
Безопасен ли открытый исходный код?.....	24
Критика и очевидные достоинства ОС Linux.....	25
<b>ЧАСТЬ I. ЗНАКОМЬТЕСЬ: UBUNTU.....</b>	<b>27</b>
<b>Глава 1. Подготовка к установке ОС Ubuntu Linux.....</b>	<b>28</b>
ОС Ubuntu Linux .....	28
Версии дистрибутивов ОС Ubuntu .....	28
Версии LTS и не-LTS .....	29
Варианты установки ОС Ubuntu Linux .....	30
Порядок установки операционных систем .....	30
Как получить дистрибутив ОС Ubuntu?.....	31
Варианты записи образа диска с ОС Ubuntu .....	33
Создание загрузочного DVD-диска.....	33
Создание загрузочного флеш-накопителя .....	33
Использование программы UNetbootin .....	33
Использование программы Rufus .....	35
Файловые системы ОС Ubuntu Linux.....	38
Файловые системы ext2, ext3 и ext4.....	38

Файловая система Btrfs .....	39
Файловая система XFS .....	39
Файловые системы ReiserFS и Reiser4 .....	39
Файловая система JFS .....	40
Файловая система ZFS.....	40
Выводы .....	40
<b>Глава 2. Теоретический минимум о файловой системе в ОС Ubuntu Linux.....</b>	<b>41</b>
Разделы жесткого диска .....	41
Первичные, расширенные и логические разделы .....	42
Точки монтирования .....	43
Точки монтирования в ОС Ubuntu .....	44
Очередность создания разделов .....	45
Объем файла подкачки swarfile.....	45
Объем корневого раздела.....	46
Объем раздела /home .....	46
Подготовительные действия перед установкой .....	46
Выводы .....	47
<b>Глава 3. Создание разделов для установки ОС Ubuntu Linux .....</b>	<b>48</b>
Выбор типа носителя для загрузки операционной системы в меню BIOS.....	48
Особенности установки ОС на платах с UEFI.....	49
Варианты запуска и установки ОС Ubuntu .....	50
Запуск ОС Ubuntu с загрузочного USB-накопителя .....	50
Запуск ОС Ubuntu с DVD-привода.....	51
Для чего нужен Live-режим? .....	52
Программа GParted.....	52
Возможности программы GParted.....	54
Разметка жесткого диска в программе GParted.....	54
Выводы .....	61
<b>Глава 4. Установка ОС Ubuntu Linux .....</b>	<b>62</b>
Запуск операционной системы в Live-режиме.....	62
Установка операционной системы из Live-режима.....	63
Выводы .....	70
<b>Глава 5. Первый запуск и первоначальная настройка ОС Ubuntu .....</b>	<b>71</b>
Запуск ОС Ubuntu .....	71
Настройка подключения к Интернету в Ubuntu .....	74
Подключение к сети по Wi-Fi.....	74
Настройка подключений в окне настроек системы .....	75
Если не определилась сетевая карта .....	76
Создание подключения Ethernet .....	77
Подключение через DSL/PPPoE.....	78
Особый случай подключения через DSL/PPPoE.....	80
Подключение с помощью USB-модема .....	87
Возможные проблемы с созданием подключений.....	90
Выводы .....	91

<b>Глава 6. Установка обновлений и локализация операционной системы.....</b>	<b>92</b>
Установка обновлений .....	92
Установка пакетов локализации .....	95
Выводы .....	97
<b>Глава 7. Обзор интерфейса ОС Ubuntu.....</b>	<b>98</b>
Графическая среда рабочего стола.....	98
Рабочий стол в GNOME.....	98
Панель основного меню .....	99
Панель задач.....	101
Рабочее место.....	103
Меню приложений.....	105
Быстрый запуск приложений.....	107
Выводы .....	107
<b>Глава 8. Настройка операционной системы и решение проблем.....</b>	<b>108</b>
Параметры системы.....	108
Настройка раскладки клавиатуры .....	109
Смена системного времени.....	112
Настройка спящего режима .....	113
Пустой рабочий стол .....	114
Изменение порядка загрузки операционной системы .....	115
Выводы .....	117
<b>ЧАСТЬ II. UBUNTU ДЛЯ ПОЛЬЗОВАТЕЛЯ.....</b>	<b>119</b>
<b>Глава 9. Структура файловой системы .....</b>	<b>120</b>
Файловый менеджер Nautilus .....	120
Имена устройств .....	122
Структура файловой системы .....	124
Общее описание структуры каталогов.....	125
Корневая файловая система.....	127
Домашний каталог пользователя.....	128
Каталог /bin/ .....	128
Каталог /boot/ .....	129
Каталог /cdrom/ .....	129
Каталог /dev/ .....	129
Каталог /etc/ .....	129
Каталог /home/.....	130
Каталоги /lib/ и /lib64/.....	130
Каталог /lost+found/ .....	131
Каталог /media/.....	131
Каталог /mnt/ .....	131
Каталог /opt/ .....	131
Каталог /proc/ .....	131
Каталог /root/ .....	132
Каталог /run/ .....	132
Каталог /sbin/.....	132
Каталог /snap/ .....	132

Каталог /srv/.....	133
Каталог /sys/ .....	133
Каталог /tmp/ .....	133
Каталог /usr/.....	133
Каталог /var/ .....	133
Выводы .....	134
<b>Глава 10. Типы файлов в ОС Ubuntu.....</b>	<b>135</b>
О файлах и каталогах .....	135
Содержимое и расширение файлов.....	136
Текстовые файлы .....	138
Исполняемые файлы.....	139
Прочие файлы .....	141
Изменение ассоциации файлов.....	141
Выводы .....	142
<b>Глава 11. Администратор и суперпользователь root .....</b>	<b>143</b>
Привилегированный пользователь root .....	143
Обычная учетная запись и запись администратора .....	144
Привилегии администратора .....	146
Административные задачи в текстовом режиме.....	148
Выводы .....	148
<b>Глава 12. Знакомство с Консолью и Терминалом .....</b>	<b>149</b>
Что такое Терминал и Консоль .....	149
Запуск Консоли.....	150
Запуск Терминала .....	151
Синтаксис команд в Терминале.....	153
Команда <i>echo</i> .....	153
Просмотр версии ядра.....	154
Установка текущего каталога .....	154
Просмотр текущего каталога.....	156
Просмотр содержимого каталога .....	156
Просмотр справки по командам .....	157
Подсказки Терминала.....	159
История введенных ранее команд.....	160
Очистка окна Терминала.....	160
Остановка выполнения команды .....	161
Выход из Терминала.....	161
Выводы .....	162
<b>Глава 13. Полезные команды Терминала .....</b>	<b>163</b>
О пользе работы в Терминале .....	163
Команда <i>whoami</i> , или "Кто я?" .....	163
Команда <i>who</i> .....	164
Создание файлов.....	165
Создание каталогов .....	168
Удаление пустых каталогов.....	169
Удаление файлов и непустых каталогов.....	170
Опасность выполнения команды <i>rm -rf/</i> .....	171

Копирование файлов и каталогов.....	172
Перемещение файлов и каталогов.....	176
Вывод содержимого файла .....	178
Вывод содержимого файла в обратной последовательности .....	181
Команды <i>more</i> и <i>less</i> для просмотра содержимого файлов.....	182
Команда поиска файлов .....	186
Отображение каталога программы .....	188
Поиск текстового шаблона в файлах .....	188
Управление остановкой системы .....	196
Выводы .....	198

## **Глава 14. Работа с привилегиями root..... 199**

О привилегиях root .....	199
Команда <i>sudo</i> вместо прав root.....	200
Что использовать: <i>sudo</i> или <i>su</i> ? .....	200
Команда <i>sudo</i> .....	201
Запуск графических программ от имени администратора.....	205
Настройка <i>sudo</i> и прав доступа на выполнение команд.....	207
Время действия введенного пароля .....	211
Выводы .....	211

## **Глава 15. Установка драйверов в ОС Ubuntu ..... 212**

Драйверы в ОС Ubuntu.....	212
Установка сетевой карты Realtek .....	213
Общий метод установки драйвера беспроводной сети .....	217
Установка драйвера видеокарты из репозитория .....	221
Общий метод установки видеодрайвера AMD/ATI Radeon.....	222
Установка аудиодрайвера .....	225
Установка ALSA из исходных текстов .....	226
Выводы .....	230

## **Глава 16. Установка принтера в ОС Ubuntu ..... 231**

Об установке принтеров, сканеров и МФУ .....	231
Где взять необходимые драйверы? .....	231
Устройства Brother .....	232
Устройства Canon .....	232
Устройства Epson.....	232
Устройства HP .....	232
Устройства Samsung .....	232
Устройства Xerox.....	233
Установка службы CUPS .....	233
Установка принтера Epson L800 средствами операционной системы.....	234
Настройка общего доступа .....	238
Установка МФУ Epson Stylus CX5900 с помощью драйверов .....	240
Выводы .....	245

## **Глава 17. Установка и удаление программ в ОС Ubuntu ..... 246**

Особенности установки и удаления программ.....	246
Зависимости пакетов в ОС Ubuntu Linux.....	247
Установка приложений с помощью Менеджера приложений Ubuntu .....	248

Установка приложений из deb-пакетов.....	251
Установка deb-пакетов с помощью Терминала.....	253
Установка приложений из Терминала .....	256
Установка приложений из исходных текстов.....	260
Репозитории в ОС Ubuntu .....	264
Snap-пакеты в Ubuntu .....	268
Удаление программ .....	269
Выводы .....	270
<b>Глава 18. Установка дополнительного программного обеспечения .....</b>	<b>271</b>
Дополнительное программное обеспечение в Ubuntu.....	271
Дополнительные медиамодули.....	271
Мультимедийные проигрыватели .....	273
Web-браузеры .....	274
BitTorrent-клиент .....	275
Файловый менеджер.....	275
Редактирование изображений.....	276
Запись CD/DVD-дисков .....	276
Выводы .....	277
<b>Глава 19. Запуск Windows-приложений в ОС Linux.....</b>	<b>278</b>
Wine для запуска приложений ОС Windows .....	278
Установка Wine.....	278
Установка и запуск Windows-приложений.....	281
Выводы .....	284
<b>Глава 20. Работа с архивами в ОС Ubuntu.....</b>	<b>285</b>
Об архивах в ОС Ubuntu.....	285
Менеджер архивов.....	285
Несжатые и сжатые архивы .....	288
Архиватор tar.....	289
Архиватор gzip.....	293
Архиватор bzip2.....	295
Прочие архиваторы .....	295
Неизвестный тип файла.....	297
Выводы .....	297
<b>ЧАСТЬ III. UBUNTU ДЛЯ ОПЫТНОГО ПОЛЬЗОВАТЕЛЯ .....</b>	<b>299</b>
<b>Глава 21. Группы пользователей и права доступа.....</b>	<b>300</b>
Пользовательские группы .....	300
Права доступа в Linux .....	300
Буквенное представление прав доступа к файлу .....	302
Буквенное представление прав доступа к каталогам.....	304
Цифровое представление прав доступа .....	307
Символьные биты SUID, SGID и sticky .....	308
Права доступа по умолчанию .....	310
Смена прав .....	312
Смена владельца файла .....	315

Смена атрибутов файла .....	315
Просмотр атрибутов .....	317
Выводы .....	318
<b>Глава 22. Жесткие и символические ссылки .....</b>	<b>319</b>
Ссылки в ОС Linux .....	319
Типы ссылок .....	320
Создание ссылок .....	321
Работа с символическими ссылками .....	323
Работа с жесткими ссылками .....	325
Выводы .....	326
<b>Глава 23. Монтирование файловых систем .....</b>	<b>327</b>
Точка монтирования .....	327
Монтирование устройств в ОС Ubuntu .....	327
Способы постоянных именовании .....	328
Именование посредством UUID .....	330
Именование посредством имен устройств .....	331
Именование посредством меток устройств .....	332
Именование посредством Hardware ID .....	333
Универсальный метод получения информации о блочных устройствах .....	334
Автоматическое монтирование .....	335
Ручное монтирование и размонтирование устройства .....	338
Выводы .....	339
<b>Глава 24. Уровни выполнения и цели в ОС Ubuntu .....</b>	<b>340</b>
Инициализация системы .....	340
Уровни выполнения .....	340
Современная система уровня запуска .....	343
Выводы .....	346
<b>Глава 25. Службы в ОС Ubuntu .....</b>	<b>347</b>
Службы, сервисы и демоны .....	347
Службы в ОС Ubuntu .....	347
Управление службами в ОС Ubuntu .....	348
Автозагрузка служб и отключение служб .....	352
Управление питанием .....	353
Управление питанием ACPI .....	354
Ведение журналов в systemd .....	355
Выводы .....	357
<b>Глава 26. Управление процессами в ОС Ubuntu .....</b>	<b>358</b>
Процессы в ОС Ubuntu .....	358
Системный монитор процессов <i>top</i> .....	358
Получение списка процессов .....	362
Завершение процесса командой <i>kill</i> .....	364
Завершение процесса командой <i>killall</i> .....	366
Управление процессами в Консоли .....	368
Прочие инструменты по управлению процессами .....	369
Выводы .....	370

<b>Глава 27. Проверка работоспособности сети в ОС Ubuntu.....</b>	<b>371</b>
Сеть в ОС Ubuntu.....	371
Определение сетевого оборудования.....	371
Проверка соединения с Интернетом.....	373
Трассировка маршрута.....	375
Выводы.....	377
<b>Глава 28. Подключение к удаленному рабочему столу.....</b>	<b>378</b>
Удаленный рабочий стол.....	378
Подключение к рабочему столу Windows.....	378
Подключение с помощью AnyDesk.....	381
Выводы.....	384
<b>Глава 29. Восстановление загрузчика GRUB.....</b>	<b>385</b>
Сбой в работе загрузчика.....	385
Восстановление загрузчика GRUB.....	385
Простой способ восстановления загрузчика GRUB.....	388
Восстановление GRUB в консоли загрузчика.....	389
Выводы.....	392
<b>Глава 30. Установка и использование ОС Ubuntu в VirtualBox.....</b>	<b>393</b>
ОС Ubuntu в виртуальной машине.....	393
Виртуальная машина Oracle VM VirtualBox.....	393
Возможные проблемы при использовании виртуальной машины.....	394
Установка дополнений.....	395
Подключение общей папки.....	396
Выводы.....	398
<b>Глава 31. Создание и установка сертификатов Let's Encrypt.....</b>	<b>399</b>
Сертификаты Let's Encrypt.....	399
Какие бывают сертификаты.....	399
Способы создания сертификата Let's Encrypt.....	400
Создание сертификата Let's Encrypt.....	401
Установка сертификата в панели управления хостингом.....	403
Проверка правильности установки сертификата.....	404
Продление сертификата.....	405
Выводы.....	405
<b>Глава 32. Настройка файлового сервера Samba.....</b>	<b>406</b>
Файловый сервер Samba.....	406
Подготовка перед установкой Samba.....	406
Установка файлового сервера Samba.....	407
Пояснение директив конфигурационного файла smb.conf.....	409
Выводы.....	411
<b>Заключение.....</b>	<b>412</b>
<b>Приложение. Описание электронного архива.....</b>	<b>413</b>
<b>Предметный указатель.....</b>	<b>414</b>

# 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 этой 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 эту 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 здесь.

Книга будет полезна и тем, кто уже хоть немного знаком с Ubuntu Linux либо с другой операционной системой на базе Linux. Пользователю, знакомому с рассматриваемой ОС, 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 имеется 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, 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.

## Чем отличается новое издание от предыдущих

Материал книги второго издания полностью переработан с учетом особенностей новой версии Ubuntu Linux.

Добавлены главы о создании и об установке сертификатов Let's Encrypt, а также добавлена глава о создании и настройке файлового сервера Samba.

## 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, которые зачастую оказываются сборником разномастных статей и берут первую попавшуюся, а потом просто забрасывают изучение новой операционной системы, т. к. материал таких книг попросту логически несвязный и непоследовательный.

В свое время, пытаясь найти хорошую и действительно полезную книгу по ОС Linux, я с сожалением обнаружил, что большинство из них либо устарели, либо просто были сборниками статей, по которым изучение ОС Linux становилось настолько некомфортным, что просто хотелось забросить эту идею. Спустя некоторое время я решил написать качественное руководство, которое не только без труда поможет вам освоить азы операционной системы Ubuntu Linux, но и обеспечит ба-

зис знаний, который послужит мощным фундаментом для более глубокого изучения этой операционной системы, а самое главное, не станет той книгой, которая оттолкнет вас от изучения ОС Linux.

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](http://volokh.info).

## Благодарности

В начале книг авторы любят кого-нибудь благодарить. Некоторые даже благодарят своих домашних питомцев (да, есть и такие). 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 издание книги совсем неизвестного автора, а не последовали совету "попробовать поработать с другими более известными авторами". Сотрудники издательства проделали огромную работу для того, 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 ядре ОС Debian GNU, которая, в свою очередь, базируется на ядре Linux.

## **ПРИМЕЧАНИЕ**

Очень часто к названию дистрибутива добавляют 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 ОС, установить ее и пользоваться ею абсолютно бесплатно.

На данный момент Ubuntu Linux — самый популярный дистрибутив не только среди рядовых пользователей, но и среди IT-специалистов. Такой интерес к 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 поддерживает большинство 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 (которые выпускаются раз в два года).

Эти и многие другие факторы являются несомненными достоинствами данной операционной системы и весомыми аргументами при выборе ее в качестве базовой ОС для вашего компьютера. Ubuntu Linux своей простотой и мощностью заслуживает внимания и достойна изучения. Однако стоит узнать историю появления самого ядра Linux, т. к. ядро Linux — основа операционной системы Ubuntu Linux.

## История появления ОС Linux

Linux (Линукс) — семейство свободно распространяемых операционных систем, основанных на ядре ОС UNIX.

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

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

ОС Minix (Миникс, от слова mini-UNIX) имела ряд ограничений на ее использование; например, применение Minix разрешалось только в образовательных целях. Это означало, что любое функционирование данной операционной системы в коммерческих целях было противозаконным. Именно поэтому Линус Торвалдс начал разработку ядра Linux на базе ОС Minix, которая, в свою очередь, является UNIX-подобной операционной системой. Немного запутанно, но такова история появления Linux.

Само название Linux происходит от объединения имени его основателя Линуса и названия ОС UNIX. Однако сам Линус Торвалдс изначально дал имя ядру Freax — гибридный английский 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, Торвалдс хотел сделать его доступным для коммерческого использования, однако в начале 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 Торвалдса на сайте **comp.os.minix**

From: torvalds@sklaava.helsinki.fi (Linus Benedict Torvalds)

To: Newsgroups: comp.os.minix

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 возжусь 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](mailto:torvalds@kruuna.helsinki.fi))

P.S. Она свободна от кода `minix` и включает мультизадачную файловую систему. Она НЕпереносима (используется переключение задач 386 и пр.) и, возможно, никогда не будет поддерживать ничего, кроме АТ-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-овую 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, а 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 Торвалдс и представляет миру свое ядро. А ведь это именно то, чего так не хватало Ричарду Столлману.

Факт, что Линус Торвалдс 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 Торвалдс свою разработку миру разработчиков.

К слову сказать, правильней 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 определенным набором утилит. Так начали появляться первые дистрибутивы.

*Дистрибутив* (англ. *distribute* — распространять) — это форма распространения конкретного 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 определенным 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. Это старейший дистрибутив 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. В1).

*Таблица В1. 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, в них разработчики воплощали 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 другого, 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) он 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 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 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 и 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 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, данный 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 — службу), и без перезагрузки операционная система примет новую конфигурацию. 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 несколько, и можно долго их перечислять.

Наверняка у многих из вас дома имеется *маршрутизатор*. Никогда не задумывались о том, что за ОС управляет работой этого устройства? Нет? Тогда для вас станет новостью, что многие маршрутизаторы управляются ПО на базе 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 часто забывают упомянуть о факте работы устройства на базе Linux. Если бы при загрузке или выключении устройства появлялся логотип пингвина Tux, то пользователи бы задумались: "Не Linux ли это?"

## Идеология свободного программного обеспечения

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 соблюдаются 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 права 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.

Для того чтобы подчеркнуть отличие лицензии GNU GPL от других лицензий (copyright), которые ограничивают свободу, ввели новый термин — **copyleft** (копи-лефт).

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 ведутся споры на тему безопасности открытого исходного кода.

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 кодом 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 (кросс-платформенный 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 дебаты 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 (аналогом 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, бóльшая 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 только 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 недолюбливать 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 будут *свободность* и *открытость* ОС Linux. Именно два этих качества позволили ОС Linux стать такой, какой мы ее видим сейчас.

Вторым достоинством стала *безопасность* операционной системы. 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 тот, что предлагается 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 столь горячую тему для обсуждения, а лучше перейдем непосредственно к изучению ОС Linux.



# ЧАСТЬ I

## Знакомьтесь: Ubuntu

<b>Глава 1.</b>	Подготовка к установке ОС Ubuntu Linux
<b>Глава 2.</b>	Теоретический минимум о файловой системе в ОС Ubuntu Linux
<b>Глава 3.</b>	Создание разделов для установки ОС Ubuntu Linux
<b>Глава 4.</b>	Установка ОС Ubuntu Linux
<b>Глава 5.</b>	Первый запуск и первоначальная настройка ОС Ubuntu
<b>Глава 6.</b>	Установка обновлений и локализация операционной системы
<b>Глава 7.</b>	Обзор интерфейса ОС Ubuntu
<b>Глава 8.</b>	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 южноафриканских языков банту и означает человечность и взаимопомощь друг другу.

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

Требования для ОС Ubuntu минимальны:

- ◆ процессор с тактовой частотой 2 ГГц;
- ◆ 4 Гбайт оперативной памяти;
- ◆ VGA-совместимая графическая карта, поддерживающая разрешение 1024×768 точек;
- ◆ 25 Гбайт свободного дискового пространства.

Эти требования касаются только самой операционной системы, но не стоит забывать, что помимо нее на вашем компьютере будут работать программы различной сложности, которые задействуют часть ресурсов компьютера, поэтому чем мощнее у вас компьютер, тем лучше.

### Версии дистрибутивов ОС Ubuntu

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

- ◆ Ubuntu Desktop;
- ◆ Ubuntu Server.

**ПРИМЕЧАНИЕ**

Ubuntu Linux 16.04 стала последней 32-разрядной версией. И уже версия Ubuntu 18.04 оказалась последним выпуском дистрибутива, для которого формировались 32-разрядные пакеты.

В книге мы будем использовать 64-битную Desktop-версию операционной системы Ubuntu Linux.

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

Помимо промежуточных (не финальных версий), 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 именно такую 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 (выпускаемые каждые полгода), 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 номер версии 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 и т. д.

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 19.04. Использовать ее в качестве настольной операционной системы не рекомендуется, т. к. эта версия ОС нестабильна и предназначена в основном для тестирования новшеств, т. е. она не является LTS-версией.

## 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 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 10. 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.

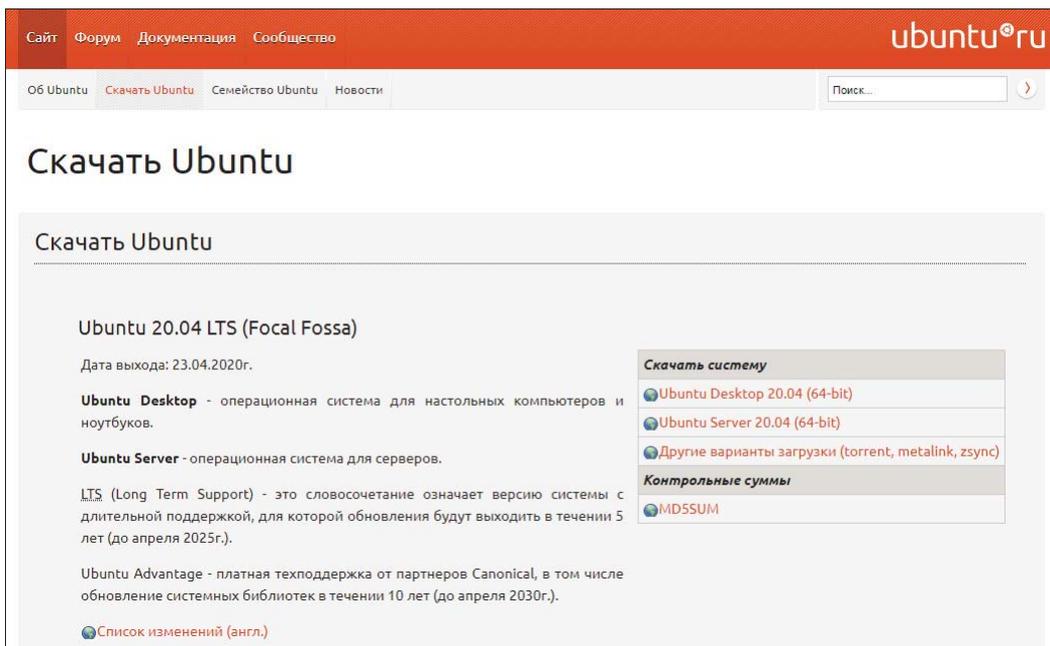
Разработчики ОС Ubuntu учли эту особенность Windows и сделали все таким образом, что установщик Ubuntu Linux более дружелюбно относится к другим опе-

рациональным 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. Если вы сомневаетесь в своих действиях или боитесь ошибиться, то лучше сохраните свои данные на другой независимый носитель.

## Как получить дистрибутив ОС Ubuntu?

Сейчас получить дистрибутив с ОС довольно легко. Ранее, при очень медленном интернет-соединении, существовала возможность заказывать доставку установочного диска почтой. Те времена прошли, и загрузка установочного образа из Сети занимает всего лишь пару минут. Для загрузки установочного образа перейдите на русскоязычный сайт <http://ubuntu.ru/> и вверху страницы нажмите ссылку **Скачать Ubuntu**. Вашему вниманию будут представлены дистрибутивы двух версий: Ubuntu Desktop и Ubuntu Server. Ранее мы уже определились с нужной версией ОС и остановили свой выбор на Ubuntu Desktop. Ее можно скачать по прямой ссылке или через любой torrent-клиент (рис. 1.1).



Сайт Форум Документация Сообщество ubuntu®ru

Об Ubuntu **Скачать Ubuntu** Семейство Ubuntu Новости Поиск...

## Скачать Ubuntu

### Скачать Ubuntu

**Ubuntu 20.04 LTS (Focal Fossa)**

Дата выхода: 23.04.2020г.

**Ubuntu Desktop** - операционная система для настольных компьютеров и ноутбуков.

**Ubuntu Server** - операционная система для серверов.

**LTS (Long Term Support)** - это словосочетание означает версию системы с длительной поддержкой, для которой обновления будут выходить в течении 5 лет (до апреля 2025г.).

Ubuntu Advantage - платная техподдержка от партнеров Canonical, в том числе обновление системных библиотек в течении 10 лет (до апреля 2030г.).

[Список изменений \(англ.\)](#)

Скачать систему
<a href="#">Ubuntu Desktop 20.04 (64-bit)</a>
<a href="#">Ubuntu Server 20.04 (64-bit)</a>
<a href="#">Другие варианты загрузки (torrent, metalink, zsync)</a>
Контрольные суммы
<a href="#">MD5SUM</a>

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 разра-

ботчики 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 в таких случаях бывают полезны.

Для прямой загрузки ISO-образа операционной системы щелкните по ссылке **Ubuntu Desktop 20.04 (64 bit)**, и файл образа дистрибутива будет загружен посредством вашего браузера.

**ПРИМЕЧАНИЕ**

Предпочтительнее загружать дистрибутив с помощью torrent-клиента. Для этого нужно щелкнуть по ссылке **Другие варианты загрузки (torrent, metalink, zsync)** и уже затем выбрать нужную версию для загрузки. Затем следует выбрать вариант **ubuntu-20.04-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, поэтому и загружать установочный образ мы будем из-под ОС Windows.

Нам остается только дождаться окончания загрузки дистрибутива Ubuntu (рис. 1.2). Когда ISO-образ дистрибутива будет загружен, запишем его на DVD-диск и приступим к установке.

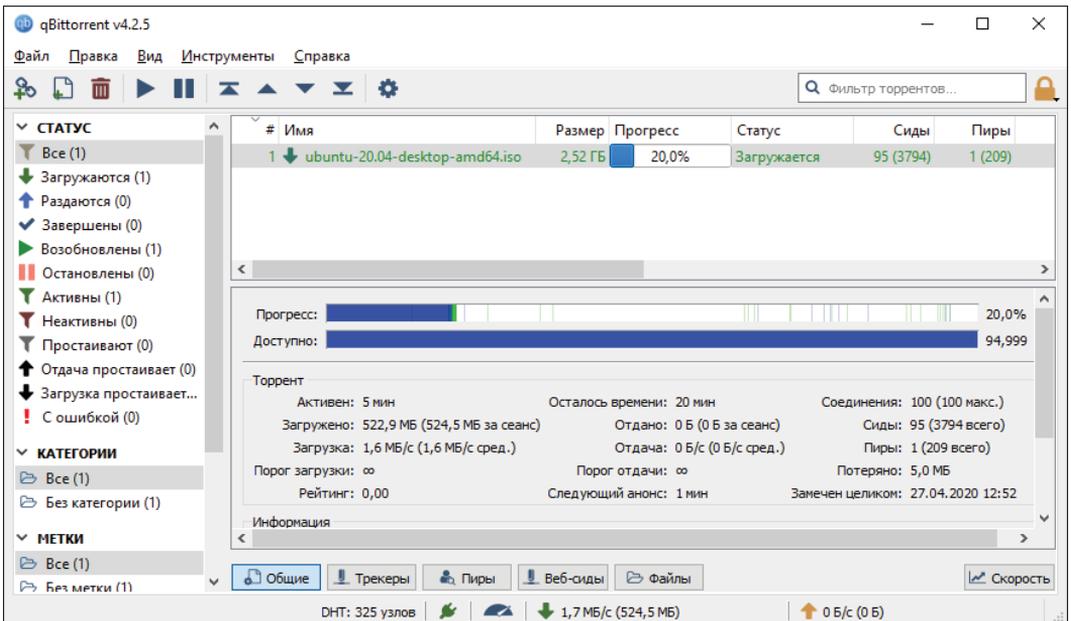


Рис. 1.2. Загрузка ОС Ubuntu Linux с помощью torrent-клиента

## Варианты записи образа диска с ОС Ubuntu

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

### Создание загрузочного DVD-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.

### Создание загрузочного флеш-накопителя

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

### Использование программы UNetbootin

Для создания загрузочной флешки существует множество программ. Первой мы рассмотрим программу UNetbootin по той причине, что она изначально предназначена для создания загрузочной флешки для различных Linux-дистрибутивов, в том числе и в самой ОС Ubuntu.

Загрузить программу можно с сайта <https://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 4, а лучше 8 Гбайт. Флешку нужно предварительно отформатировать под файловую систему FAT32, если она была у вас с другой файловой системой.

**ПРИМЕЧАНИЕ**

Если не выполнить форматирование, то находящиеся на флешке файлы не удаляются. Таким образом, если ранее на флешку был записан другой образ дистрибутива, то в дальнейшем возникнет ошибка установки операционной системы.

После этого запустите программу UNetbootin и установите переключатель **Образ диска** (рис. 1.3).

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

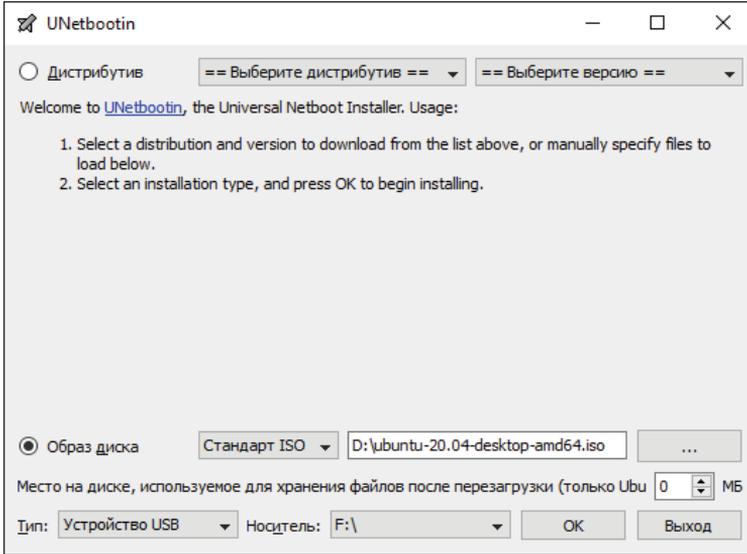


Рис. 1.3. Программа UNetbootin

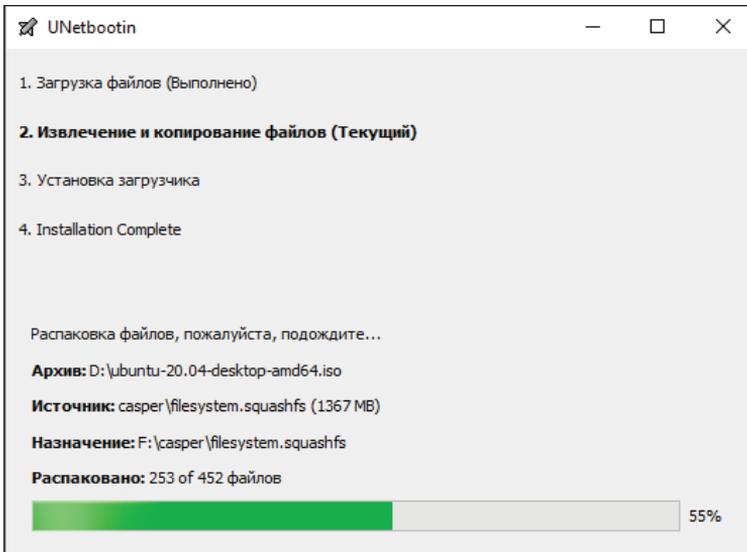


Рис. 1.4. Начат процесс копирования файлов на USB-накопитель

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).

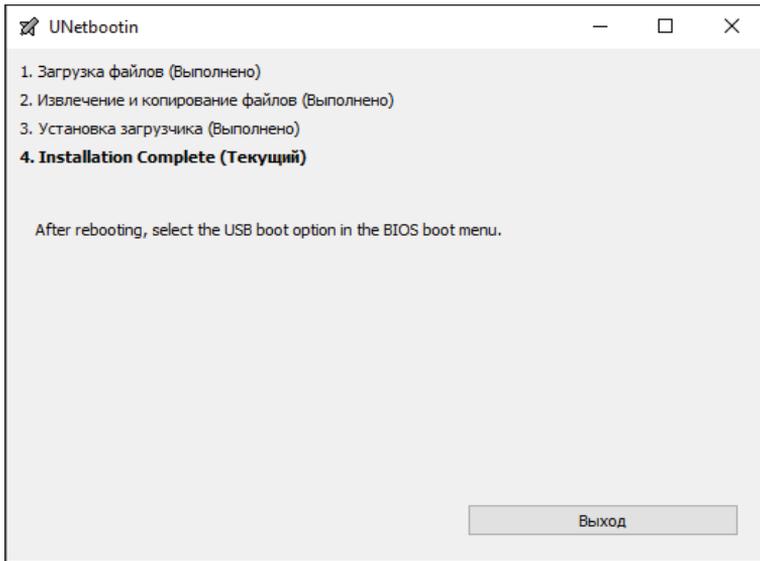


Рис. 1.5. Программа UNetbootin сообщила о завершении процесса создания загрузочной флешки

## Использование программы Rufus

Загрузочную флешку можно создать с помощью другой программы — Rufus, скачать которую можно с сайта <https://rufus.ie>. Программа Rufus очень быстро завоевала сердца пользователей благодаря простому интерфейсу и своим функциональными возможностями. Единственный недостаток этой программы — она работает только под ОС Windows. Сейчас это для нас не важно, т. к. мы будем создавать загрузочную флешку именно в ОС Windows 10.

Запустите программу Rufus и выберите установочный образ ОС Ubuntu Linux, нажав кнопку **Выбрать** (рис. 1.6).

В раскрывающемся списке раздела можно выбрать схему создания раздела: **MBR** либо **GPT**. Если у вас современная материнская плата с поддержкой UEFI вместо устаревшего BIOS, вы можете смело выбирать **GPT**. В нашем случае мы остановимся на **MBR**.

В поле ввода **Новая метка тома** вы можете задать понятную вам метку для вашей флешки или просто оставить значение по умолчанию и нажать кнопку **СТАРТ** для начала создания загрузочной флешки. Будет выведено окно, как на рис. 1.7.

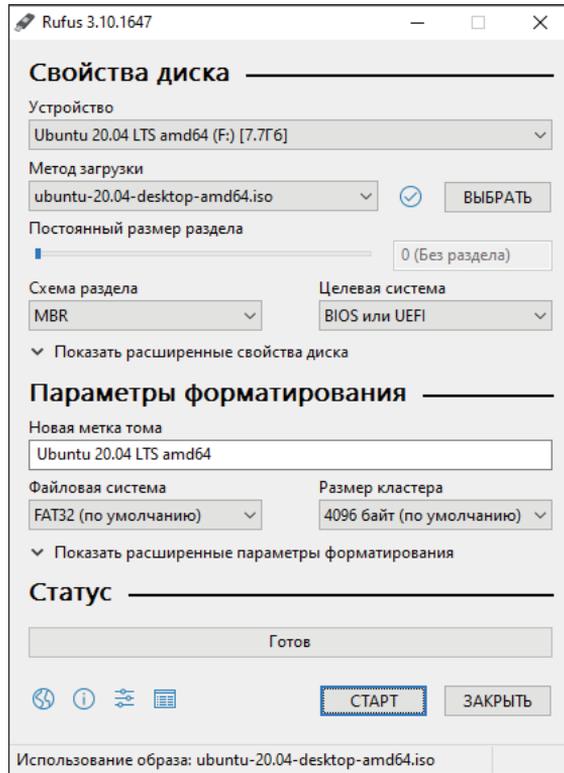


Рис. 1.6. Выбор загрузочного образа в программе Rufus

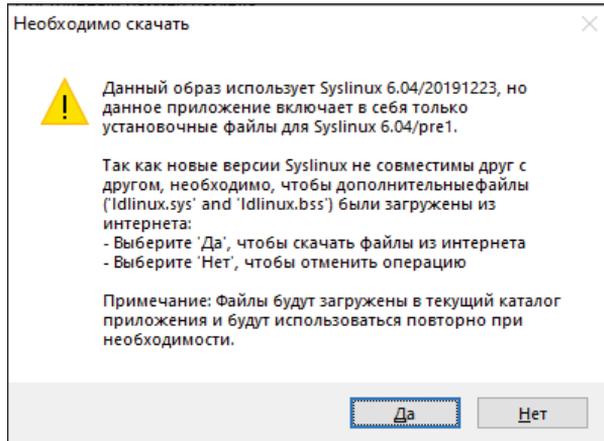


Рис. 1.7. Загрузка недостающих для создания образа файлов

Нажмите кнопку **Да**, чтобы программа Rufus смогла скачать недостающие файлы, которые необходимы для создания загрузочной флешки. После этого программа попросит указать, в каком режиме будет создан данный загрузочный образ (рис. 1.8).

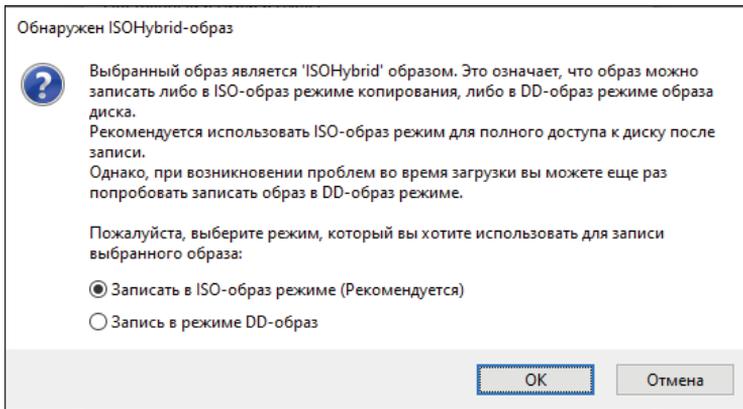


Рис. 1.8. Выбор режима создания загрузочного образа

Оставьте предложенное значение по умолчанию и нажмите кнопку **ОК**. На предупреждение программы о том, что данная операция уничтожит все файлы, находящиеся на флешке, ответьте согласием, нажав кнопку **ОК**. Начнется процесс копирования файлов загрузочного образа на флешку. По завершении процесса надпись статуса "Готов" будет свидетельствовать о том, что процесс создания загрузочной флешки завершен успешно (рис. 1.9).

Нажмите кнопку **ЗАКРЫТЬ**, чтобы выйти из программы.

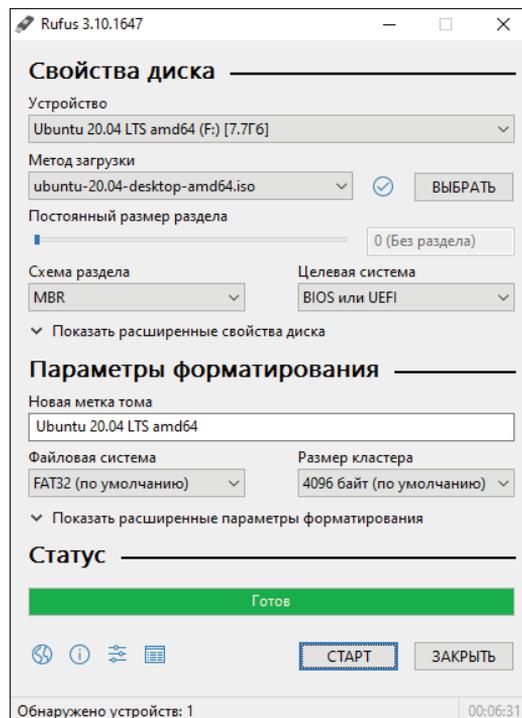


Рис. 1.9. Создание загрузочной флешки завершенно

## 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, ext3, 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. Эта 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. 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. Однако ожидать того, что

в ближайшее время ReiserFS внедрят в поддержку ядра, не придется по причине того, что эта файловая система не имеет активной поддержки со стороны разработчиков.

## 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.

## Файловая система ZFS

ZFS (Zettabyte File System — расширенная файловая система) — экспериментальная файловая система, добавленная в поддержку ОС Ubuntu, начиная с версии 19.10. Изначально она предназначалась для ОС Solaris, выпущенной компанией Sun Microsystems.

Данная ФС позволяет работать с большими по размеру файлами — в 16 Эбайт. Кроме того, эта файловая система выступает в роли менеджера физических и логических дисков, т. е. ZFS позволяет охватывать (объединять) множество дисков.

Очень полезной особенностью ZFS является то, что при перезаписи данных информация записывается в совершенно другой блок, и только потом уже (в случае успешного процесса перезаписи) обновляются метаданные файловых систем. Эта возможность позволяет не потерять данные, например при внезапном отключении электричества. Если процесс перезаписи завершился с ошибкой, то метаданные файловой системы будут по-прежнему указывать на старый блок с информацией.

Это неполный перечень особенностей ZFS. Однако даже при всех своих впечатляющих возможностях ZFS пока еще только пытается "прижиться" в ОС Linux, и не факт, что это когда-то произойдет. Именно поэтому использовать ZFS на рабочих серверах все еще не рекомендуется.

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 версию данной операционной системы и скачали ее образ. Затем научились создавать загрузочный образ в программах UNetbootin и Rufus для последующей установки ОС Ubuntu 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, обычно один раздел жесткого диска выделяется для хранения файлов самой ОС, а другой — для хранения личной информации.

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

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

Преимущества использования разделов множество. Перечислим только основные:

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

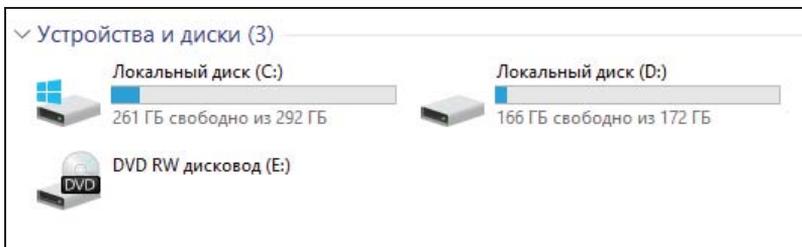


Рис. 2.1. Разделы жесткого диска в ОС Windows

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.

По мере развития технологического прогресса производители компьютерного оборудования стали отказываться от использования MBR, и этот метод разбиения информации жесткого диска постепенно сошел на нет.

Производители компьютерного оборудования поняли, что требуется что-то новое, нежели MBR. Так появился расширяемый микропрограммный интерфейс EFI (Extensible Firmware Interface), который предназначен заменить собой морально устаревшую систему BIOS.

EFI был всего лишь первой версией, точнее, первым пробным проектом, который потребовал небольшой доработки, т. к. идеи, заложенные в нем, и его предназначение показали свою актуальность замены BIOS. Таким образом, доработка EFI получила название UEFI (Unified EFI).

UEFI стал заменой привычной BIOS, в котором вместо MBR используется GPT (GUID Partition Table). При использовании таблицы разделов GUID каждому разделу диска присваивается уникальный глобальный идентификатор — GUID. Этот идентификатор является настолько уникальным, что однажды сгенерированный GUID никогда больше не может быть повторен. Можно создавать сколько угодно разделов и быть уверенным в том, что GUID каждый раз будет уникальным. Это же, в свою очередь, снимает все ограничения на количество разделов.

Так, в ОС Windows можно создать до 128 разделов. Почему именно 128, а не больше? Дело в том, что при разметке жесткого диска резервируется именно 128 записей данных о разделах. Согласитесь, зачем еще больше разделов? Ведь на оглавление таблицы разделов резервируется определенная часть жесткого диска, которая в большинстве случаев будет попросту расходоваться зря.

Говорить о принципах логического разбиения можно много, но для начала понимания самой сути разбиения пространства жесткого диска этого будет достаточно. К тому же при установке или дальнейшей работе с ОС Ubuntu вам так или иначе придется с этим столкнуться.

## 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 будут / 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 у нас 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, становится понятным, 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 два раздела (точки монтирования) и задать размер файла подкачки (swapfile):

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

### **ПРИМЕЧАНИЕ**

Начиная с версии ОС Ubuntu 18.04, вместо точки монтирования /swap при установке операционной системы создается файл swapfile. Как таковой раздел /swap больше не существует, а является лишь файлом в корневой файловой системе.

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 10, 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 разделения 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 миниму-

му 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 swapfile.

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, можно создать каталог /home. В любом случае это можно сделать и после установки ОС Ubuntu.

## Очередность создания разделов

Изначально, когда компьютеры были не такими быстрыми и мощными, как сейчас, каждый программист бился за быстродействие системы и экономил процессорное время. Со временем эта необходимость отпала, и программы стали создавать без учета прежней экономии памяти и процессорного времени.

Примерно с тех пор при установке операционной системы и 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, данное правило очередности соблюсти не удастся, т. к. ОС Windows уже заняла требуемое место на жестком диске. Однако расстраиваться совсем не обязательно — на современном компьютере мы практически не заметим разницу в скорости обращения к жесткому диску.

## Объем файла подкачки swapfile

В Интернете можно встретить различные доводы в пользу выбора того или иного размера файла подкачки swapfile. 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. swapfile, 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 ни 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.

## 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 значительного объема 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, этот раздел создавать необязательно. 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 дополнительное 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 в разделе **Средства администрирования Windows** найдите и запустите программу Оптимизация дисков.

В окне программы Оптимизация дисков выделите нужный диск, например D:, и нажмите кнопку **Оптимизировать**. Будет произведен анализ с последующей дефрагментацией 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.

Также мы познакомились с GPT, которая пришла на смену MBR и сделала более гибким процесс создания и управления разделами жесткого диска.

Полученные 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, скорее всего, 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 <Del>, 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 <Del>). Теперь при загрузке компьютера во время появления надписи "Press Del to enter Setup" вы должны успеть нажать клавишу <Del>, после чего вы попадете в основное меню BIOS.

### **ПРИМЕЧАНИЕ**

Дать какие-то четкие инструкции по настройке загрузки в BIOS не удастся, т. к. эти 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 типа носителя, на котором у вас находится установочный образ с ОС Ubuntu).

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.

Для некоторых моделей 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". В этом случае настраивать приоритет загрузки в меню BIOS не нужно. От вас потребуется только один раз нажать клавишу <F12> и выбрать тип носителя, с которого вы хотите загрузить компьютер, а дальше установка будет уже происходить с выбранного носителя.

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 посредством клавиши <Enter>. Для того чтобы вернуться в предыдущее меню, нужно нажать клавишу <Esc>.

## **Особенности установки ОС на платах с UEFI**

Как нам уже известно, на новых компьютерах базовую систему ввода-вывода BIOS сменил 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 и является его прямым потомком, сильно доработанным и улучшенным.

Для совместимости с устаревшим программным обеспечением во многих версиях UEFI есть поддержка эмуляции BIOS, которая называется Legacy либо UEFI Disabled. Этот режим создан специально для того, чтобы вы смогли работать с устаревшей ОС, которая разрабатывалась еще до создания UEFI.

Платы, имеющие поддержку UEFI, позволяют работать с новой таблицей разделов GPT, которая обеспечивает создание уже не четырех (в случае с MBR), а множества разделов (до 128 разделов) на диске. При этом они все будут основными, т. е. первичными. Замечательно, не так ли?

Выбор приоритета загрузки в UEFI практически идентичен выбору приоритетов в меню BIOS, за исключением того, что UEFI эта процедура наиболее интуитивно понятна. Также UEFI поддерживает загрузку с нужного носителя, не прибегая к настройкам самого UEFI. Для этого при запуске компьютера достаточно нажать нужную клавишу (обычно это <F2> либо <F8>), и дальше установка будет происходить с выбранного вами носителя.

## 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.

Справедливости ради стоит отметить, что загрузка с USB-накопителя в скором будущем вытеснит загрузку с оптического диска CD либо DVD по той причине, что загрузка с USB-накопителя в разы быстрее и к тому же установочные дистрибутивы разрослись до таких размеров, что не всегда умещаются на оптическом диске.

### Запуск ОС Ubuntu с загрузочного USB-накопителя

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

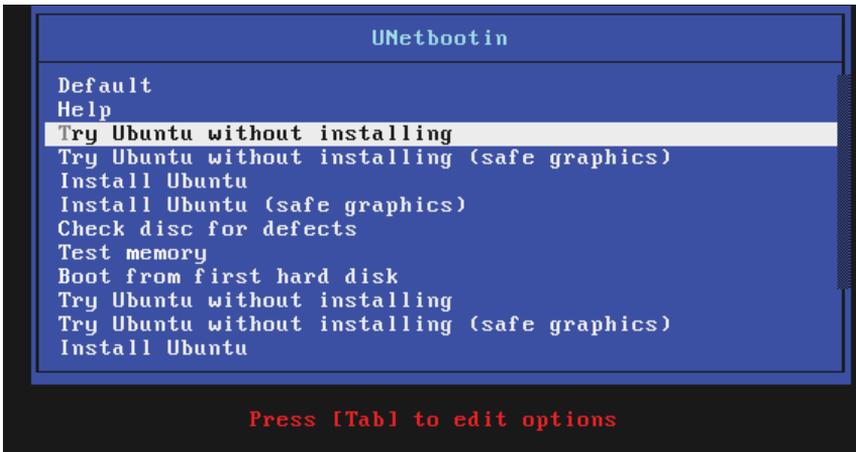


Рис. 3.1. Окно выбора меню загрузки в UNetbootin

#### ПРИМЕЧАНИЕ

При использовании загрузочного USB-накопителя, созданного в программе Rufus, никаких меню загрузок выводиться не будет — загрузится сразу приветственное окно установщика ОС Ubuntu.

Выберите третий вариант загрузки **Try Ubuntu without installing** и нажмите клавишу <Enter>. Начнется загрузка операционной системы.

#### ПРИМЕЧАНИЕ

При запуске ОС Ubuntu с загрузочного носителя (DVD-диска либо USB) установщик запустит процесс проверки корректности образа. Отменить проверку можно нажатием

комбинации клавиш <Ctrl>+<C>, т. к. сам процесс проверки целостности образа не обязателен.

На данном этапе никаких дополнительных действий от вас не требуется, и вам нужно только дождаться загрузки ОС Ubuntu в Live-режиме (рис. 3.2).

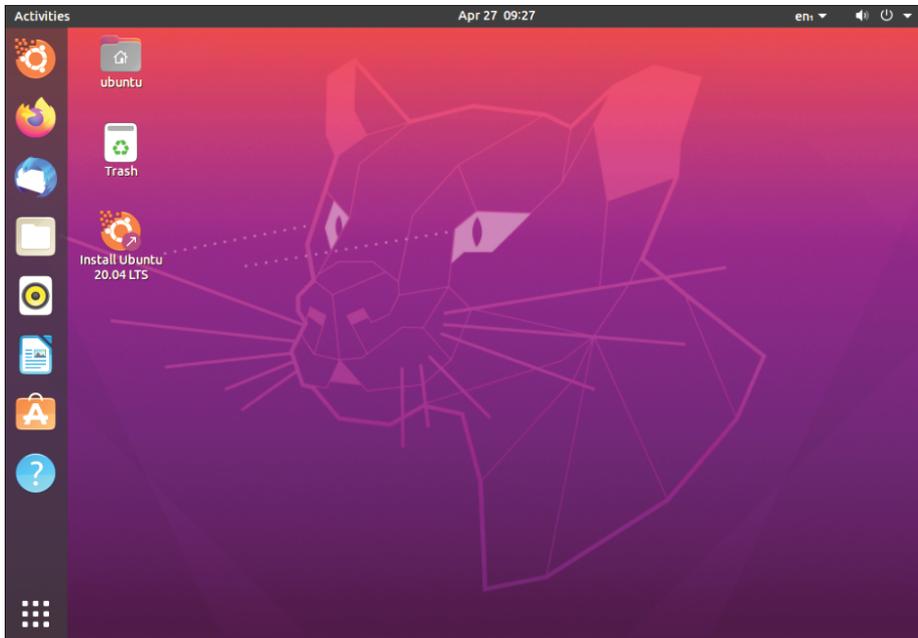


Рис. 3.2. ОС Ubuntu загрузилась в Live-режиме

Теперь ОС Ubuntu загружена в Live-режиме, который позволит вам сделать разметку диска для установки операционной системы. Но об этом чуть позже, а сейчас рассмотрим вариант загрузки с DVD-привода, который идентичен загрузке установочного образа с USB-накопителя, созданного в программе Rufus.

#### **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 откроется приветственное окно установщика ОС Ubuntu (рис. 3.3).

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

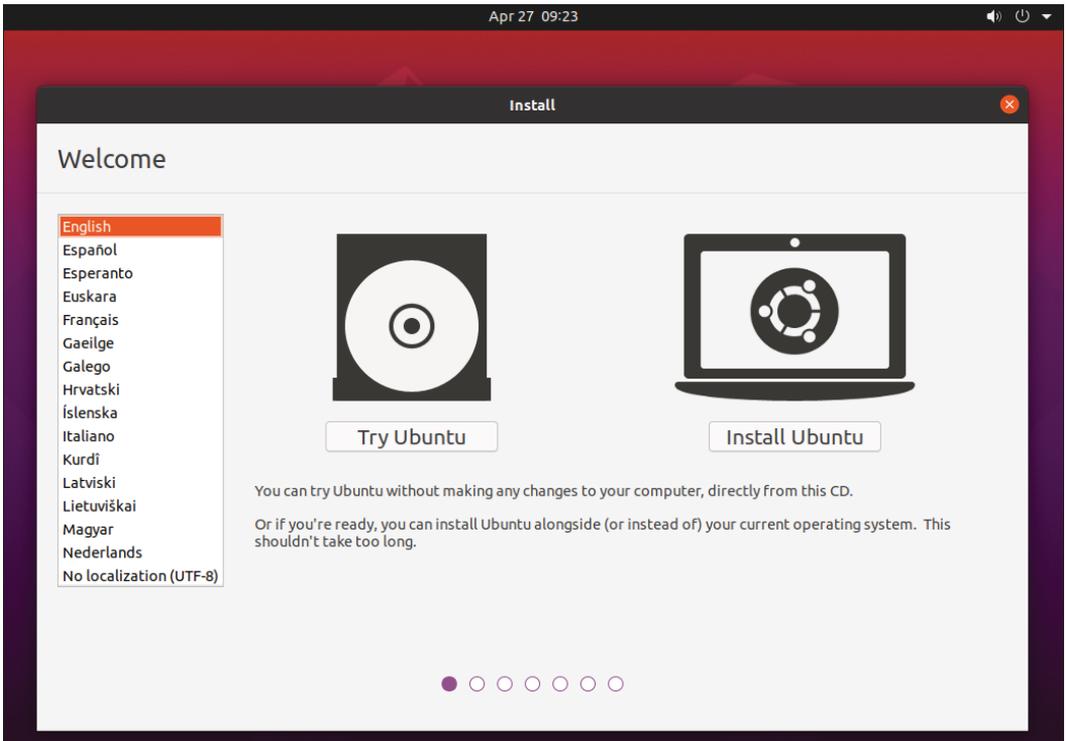


Рис. 3.3. Приветственное окно установщика Ubuntu

## 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 дефрагментировали раздел, 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.

## Программа GParted

После запуска ОС Ubuntu в Live-режиме нам нужно освободить место на жестком диске для установки Ubuntu Linux. Выделять свободное пространство мы будем на диске 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 **Show Applications**, которая расположена в левом нижнем углу на панели задач (по типу кнопки **Пуск** в ОС Windows) — рис. 3.4.

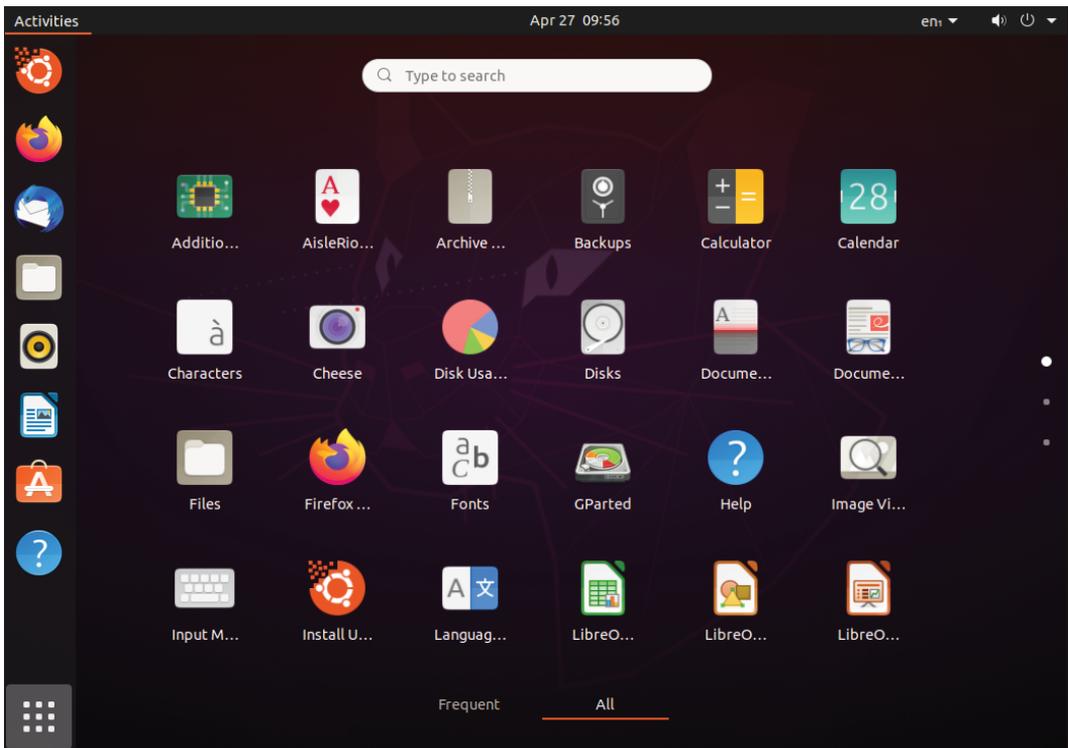


Рис. 3.4. Запуск меню поиска приложений

Затем в строке поиска введите слово `gparted`. В ответ на это главное меню мгновенно отобразит в качестве результата значок программы GParted Partition Editor (далее — просто GParted) — рис. 3.5.

Запустите найденную программу GParted. Она поможет нам разметить пространство жесткого диска.

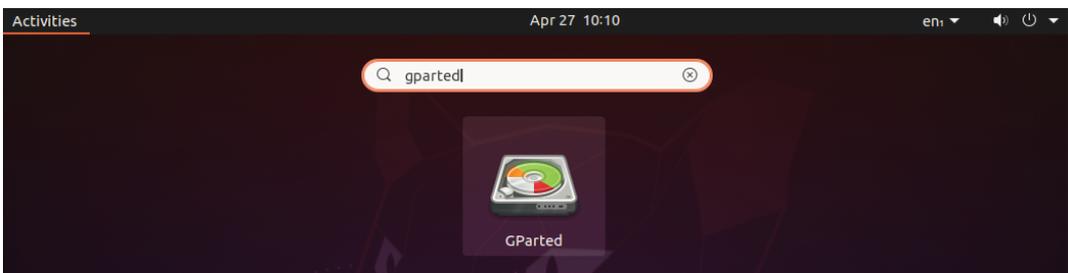


Рис. 3.5. Результат поиска программы GParted Partition Editor

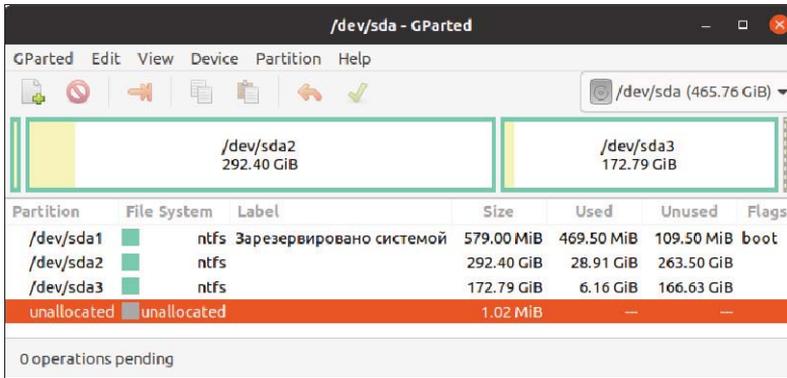


Рис. 3.6. Разметка жесткого диска в программе GParted

Разметка жесткого диска с двумя разделами C: и D: в общем виде выглядит так, как показано на рис. 3.6.

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

- ◆ /dev/sda1 — раздел, автоматически создаваемый при установке ОС Windows 10 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 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 отсутствовали какие-либо копии на других носителях. Именно поэтому рекомендуется сделать резервную копию всей важной для вас информации и только после этого приступить к процессу разметки жесткого диска в программе GParted.

Так как установить ОС Ubuntu прямо в раздел, созданный ОС Windows, невозможно, 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. 3.7).

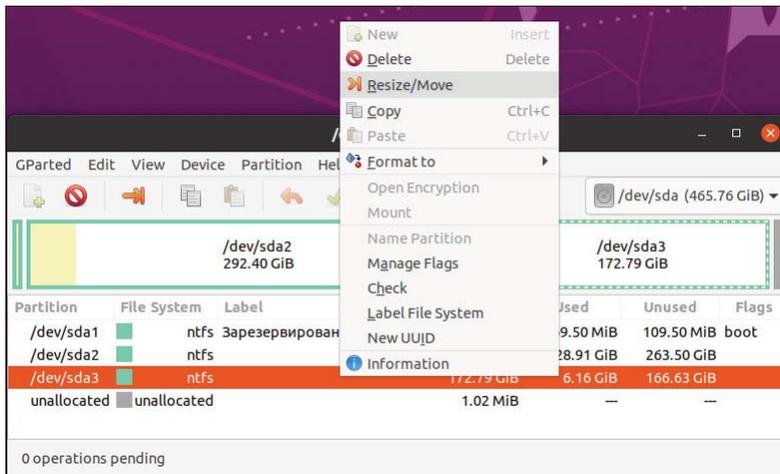


Рис. 3.7. Изменение размера раздела в программе GParted

Давайте выделим 50 Гбайт. Для этого в поле счетчика **Free space following (MiB)** введите значение 50000, т. е. размер в мегабайтах, и после этого нажмите кнопку **Resize/Move** (рис. 3.8). Этим действием мы выделяем свободное пространство с диска D:, которое нам понадобится для установки ОС Ubuntu. Получится примерно такая картина, как на рис. 3.9: у нас появился неразмеченный раздел, который значится как **unallocated**. Именно его мы и будем сейчас разметать.

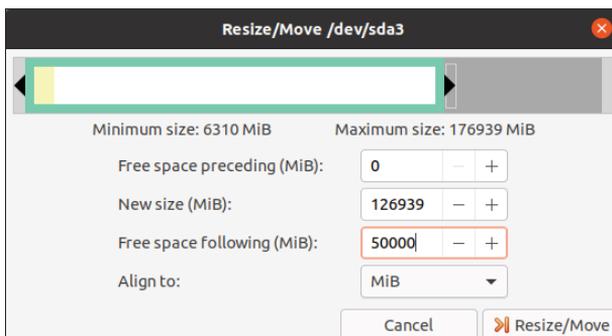


Рис. 3.8. Выделение размера для установки ОС Ubuntu

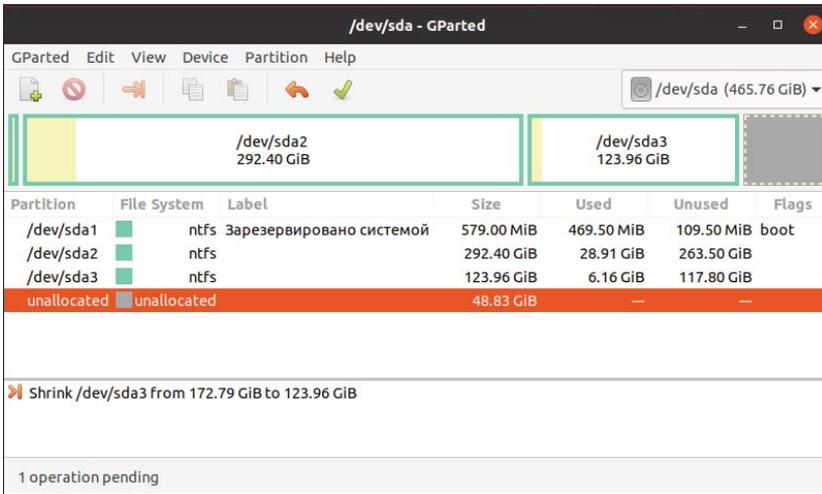


Рис. 3.9. Программа GParted выделила 50 Гбайт с диска D:

Обратите внимание, что у нас уже имеются три первичных раздела, а, как мы уже знаем, MBR не позволяет создавать больше четырех первичных разделов. Если неразмеченный раздел пометить как *первичный*, то нам больше не удастся создать дополнительных разделов, и при попытке создания будет выведено сообщение об ошибке (рис. 3.10).



Рис. 3.10. 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. 3.11).

В раскрывающемся списке **Create as** выберите вариант **Extended Partition** и нажмите кнопку **Add** для сохранения изменений. Это позволит нам создать расши-

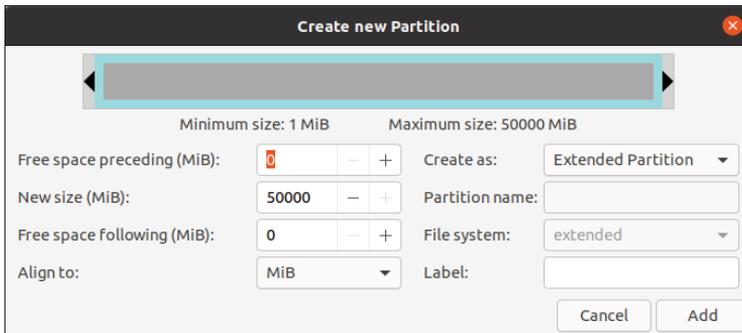


Рис. 3.11. Создание расширенного раздела

ренный 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. 3.12.

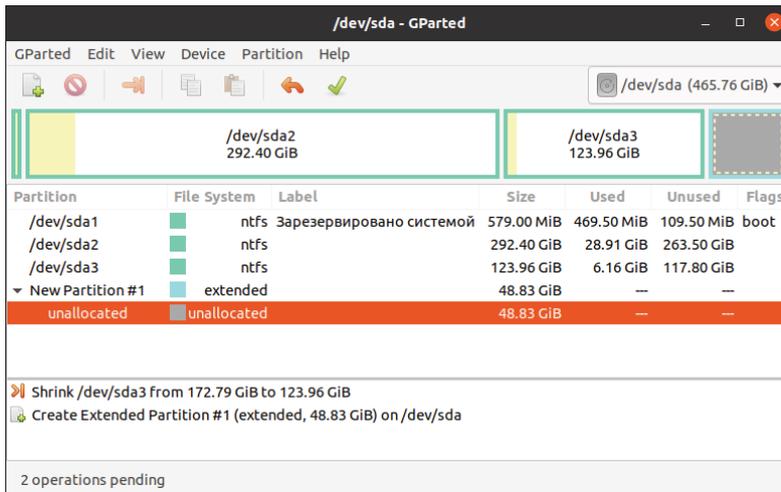


Рис. 3.12. Новый раздел помечен как расширенный

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 172,79 Гбайт на 123,96 Гбайт. Второй очередью задания создается расширенный раздел размером в 48,83 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 зададим 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**.

В поле счетчика **New size (MiB)** установите размер файла подкачки (рис. 3.13). В нашем случае это 4500 Мбайт. Затем в раскрывающемся списке **File system** выберите значение **linux-swap**, которое позволит пометить созданный раздел как файл подкачки. Для подтверждения создания раздела нажмите кнопку **Add**.

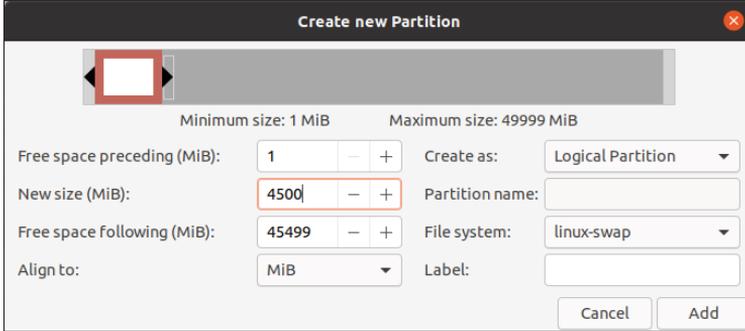


Рис. 3.13. Создание файла подкачки

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. 3.14).

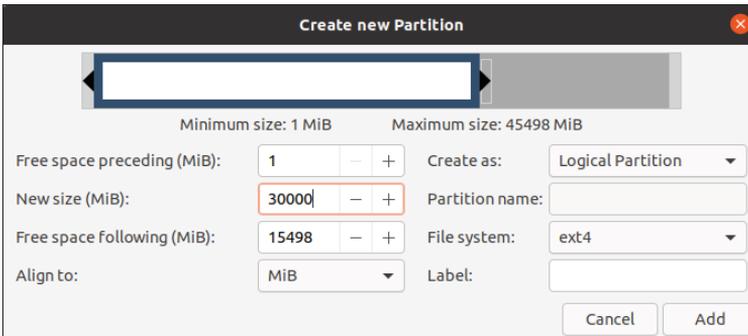


Рис. 3.14. Создание корневого раздела файловой системы

Давайте для корневой файловой системы выделим 30 Гбайт. Для этого в поле счетчика **New size (MiB)** установите значение равным 30000. 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**.

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

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

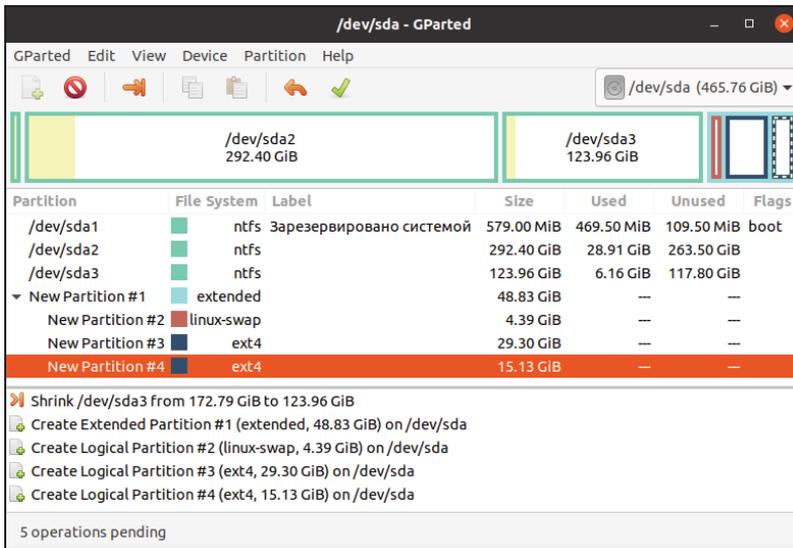


Рис. 3.15. Финальный этап создания разделов для установки ОС Ubuntu

В итоге у нас получится примерно такая конфигурация, как показано на рис. 3.15.

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. 3.16).

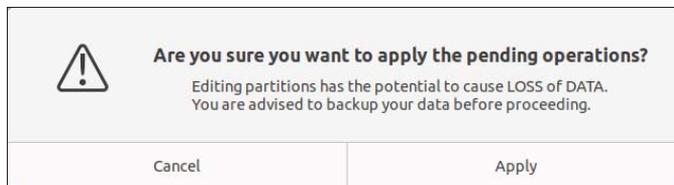


Рис. 3.16. Подтверждение проделанных изменений

Нажмите кнопку **Apply**, и появится окно, отображающее ход выполнения изменений (рис. 3.17).

Этот процесс занимает не так уж много времени, однако если ваш жесткий диск большого размера, то можно сходить попить чай или запустить браузер Firefox прямо из Live-режима и почитать что-нибудь в Интернете.

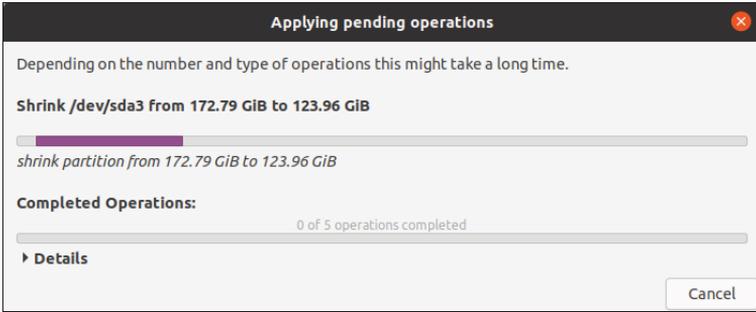


Рис. 3.17. Процесс разметки созданных разделов

По окончании процесса разметки жесткого диска будет выведено окно с сообщением об успешном процессе всех операций (рис. 3.18).

Теперь разметка жесткого диска выглядит так, как показано на рис. 3.19.

Как оказалось, нет ничего сложного в ручной разметке жесткого диска. Все операции выполняются довольно легко и интуитивно понятны. Главное при этом —

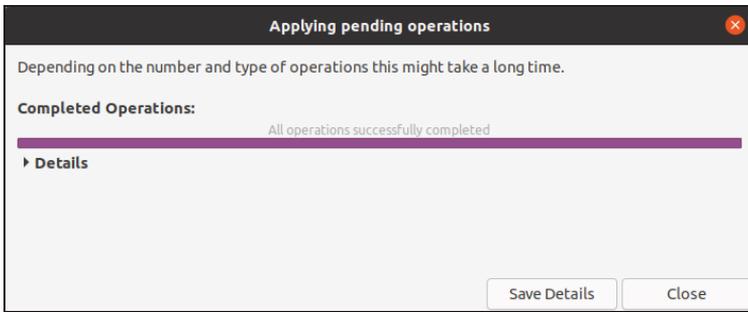


Рис. 3.18. Программа GParted сообщила об успешном завершении всех операций

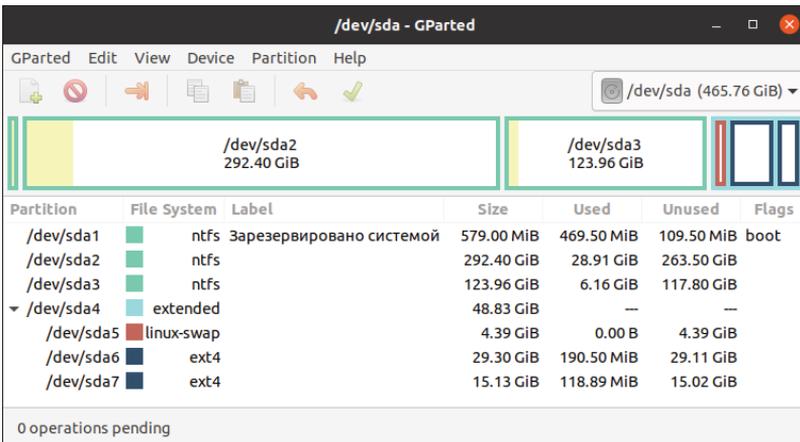


Рис. 3.19. Созданные разделы на жестком диске

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.

### *ПРИМЕЧАНИЕ*

Напомним, что при запуске ОС Ubuntu с загрузочного носителя установщик предложит проверить целостность образа. Отменить эту необязательную проверку можно нажатием комбинации клавиш <Ctrl>+<C>.

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.

На рабочем столе присутствуют три ярлыка:

- ◆ **ubuntu;**
- ◆ **Trash;**
- ◆ **Install Ubuntu 20.04 LTS.**

Сейчас нас интересует третий ярлык — **Install Ubuntu 20.04 LTS**, и мы сейчас узнаем, для чего он предназначен.



Рис. 4.1. Рабочий стол ОС Ubuntu

## Установка операционной системы из Live-режима

Для запуска процесса установки операционной системы дважды щелкните левой кнопкой мыши по ярлыку **Install Ubuntu 20.04 LTS**.

Запустится мастер установки ОС Ubuntu (рис. 4.2).

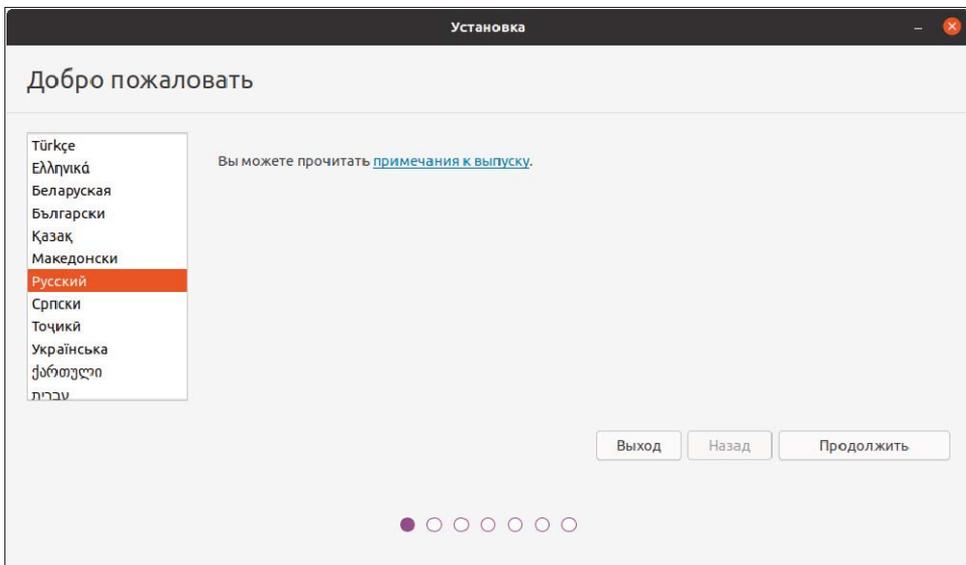


Рис. 4.2. Мастер установки ОС Ubuntu предлагает выбрать язык установки ОС

Здесь нам предлагается выбрать основной язык для ОС Ubuntu. В левой части окна выберите вариант **Русский** и нажмите кнопку **Продолжить**. После чего нам будет предложено выбрать раскладку клавиатуры (рис. 4.3).

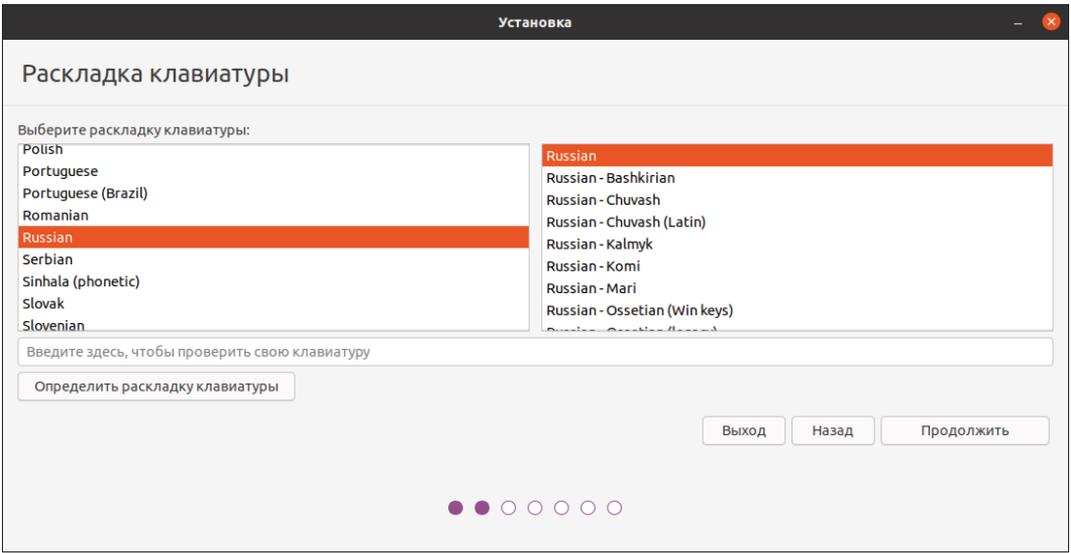


Рис. 4.3. Выбор раскладки клавиатуры и используемого языка

В левой части окна выберите **Russian**. В правой части окна автоматически будет выбран пункт **Russian**. Нажмите кнопку **Продолжить**.

На следующем этапе установки вам будет предложено загрузить обновления во время установки Ubuntu и установить стороннее программное обеспечение для видеокарт и сетевого оборудования, такого как Wi-Fi, а также загрузить дополнительное программное обеспечение для поддержки различных медиаформатов (рис. 4.4).

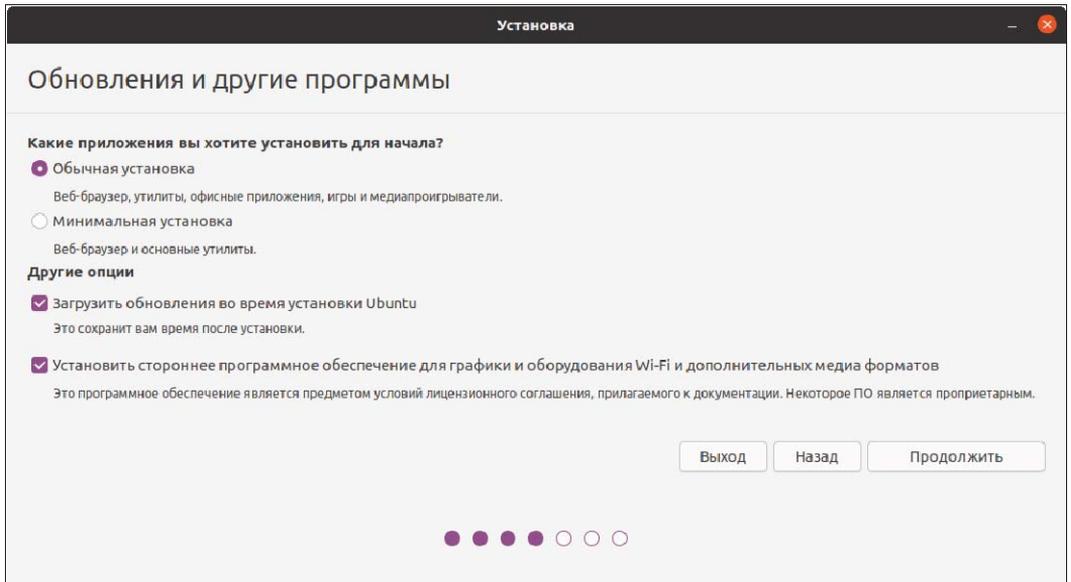


Рис. 4.4. Выбор обновлений и стороннего программного обеспечения

Если ваш компьютер подключен к Интернету, можно загрузить обновления во время установки операционной системы. Но это не обязательно, 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 шаг 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. 4.5).

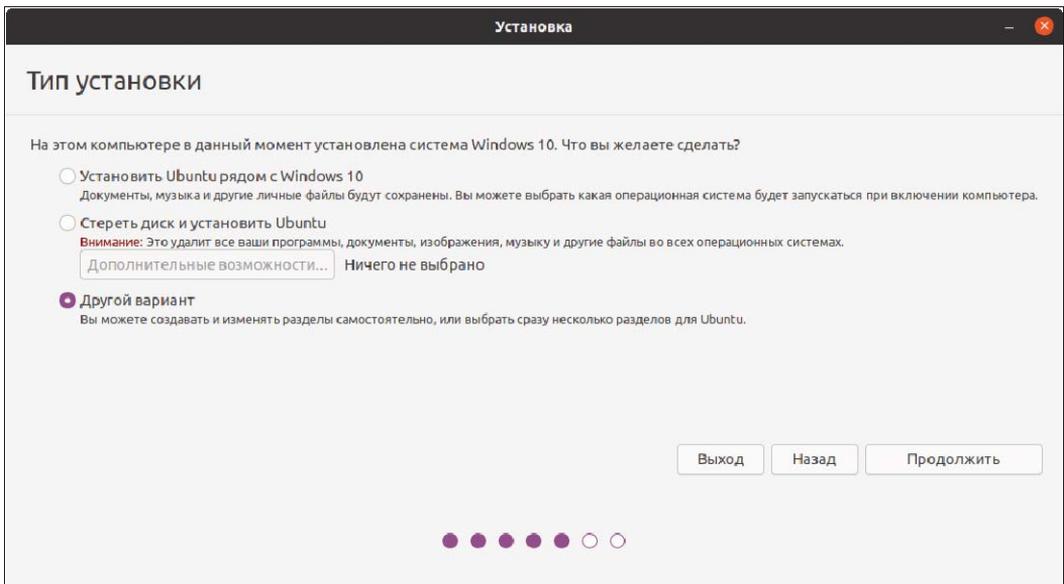


Рис. 4.5. Выбор типа установки операционной системы

Здесь нам предлагаются на выбор три типа установки.

- ◆ **Установить Ubuntu рядом с Windows 10.** 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 подходит, т. к. мы уже самостоятельно разметили разделы диска.
- ◆ **Стереть диск и установить Ubuntu.** Этот вариант приемлем 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. 4.6).

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

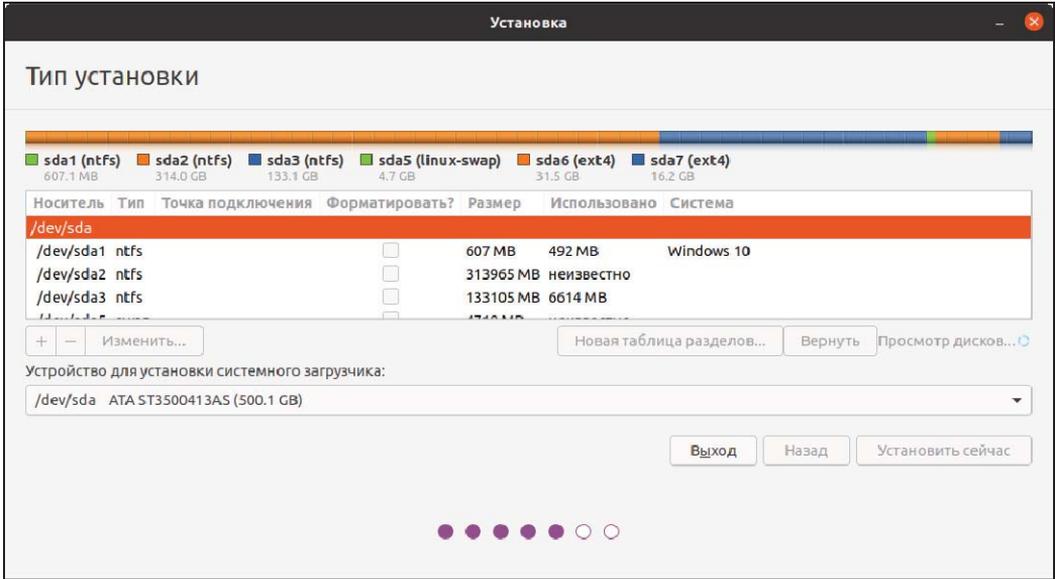


Рис. 4.6. Окно со списком разделов жесткого диска

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. 4.7).

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

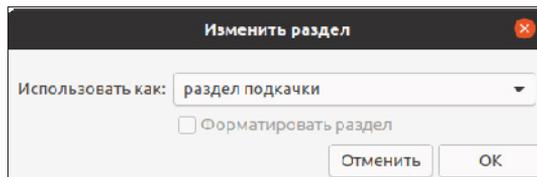


Рис. 4.7. Изменение раздела для файла подкачки

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, в котором необходимо задать нужные нам значения (рис. 4.8).

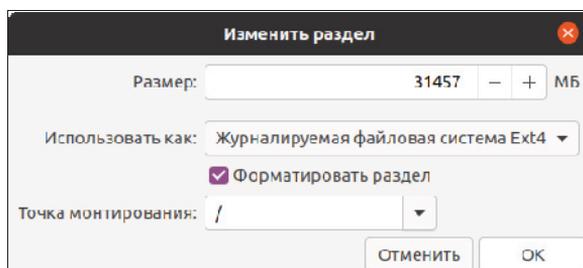


Рис. 4.8. Редактирование раздела для корневой файловой системы

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. 4.9.

Поскольку 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 внесенных изменений (рис. 4.10).

Нажмите кнопку **Продолжить**. Появится окно выбора часового пояса компьютера (рис. 4.11). Укажите часовой пояс, выбрав нужный город на карте либо введя нужное значение в поле ввода, 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.

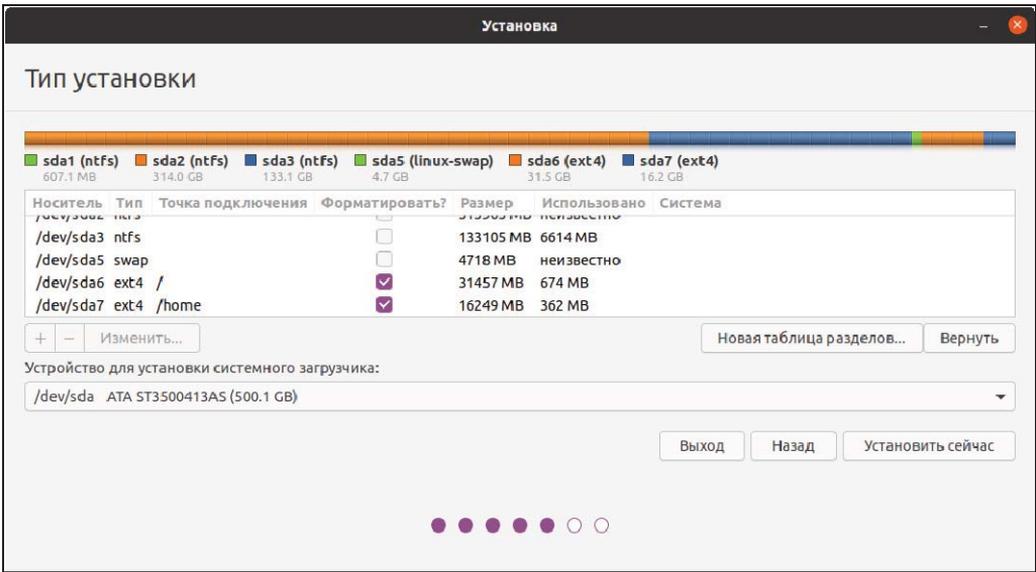


Рис. 4.9. Окончательная разметка разделов жесткого диска

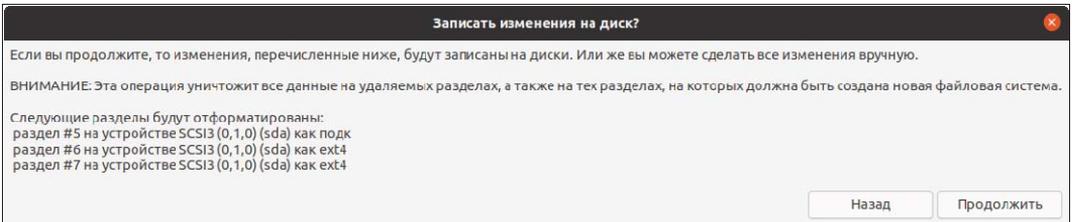


Рис. 4.10. Окно подтверждения проделанных изменений

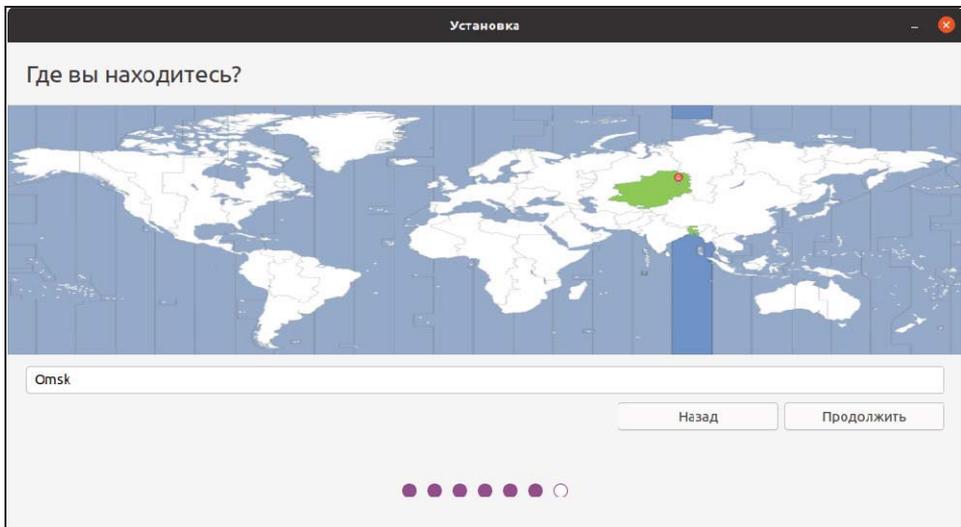


Рис. 4.11. Выбор часового пояса компьютера

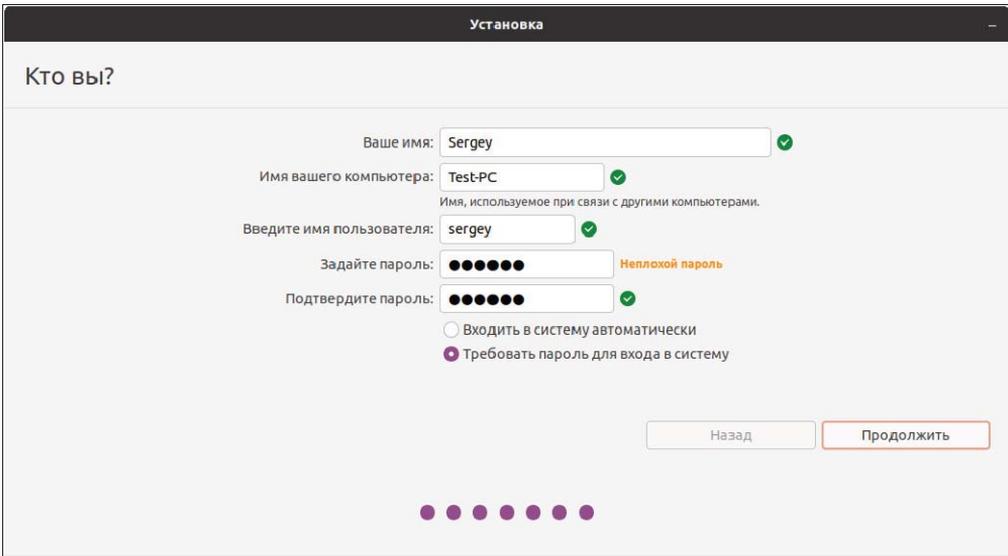


Рис. 4.12. Создание пользователя компьютера и его пароля

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**, нужно вводить 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.

### СОВЕТ

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**.

### ПРИМЕЧАНИЕ

В более ранних версиях ОС Ubuntu Linux установщик предлагал отметить флажок **Шифровать мою домашнюю папку**. Шифрование папки предполагает защиту ваших личных данных извне, т. е. если кто-то попытается прочитать ваши данные, загрузившись с Live-CD, то это ему не удастся сделать, т. к. данные будут зашифрованы.

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

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

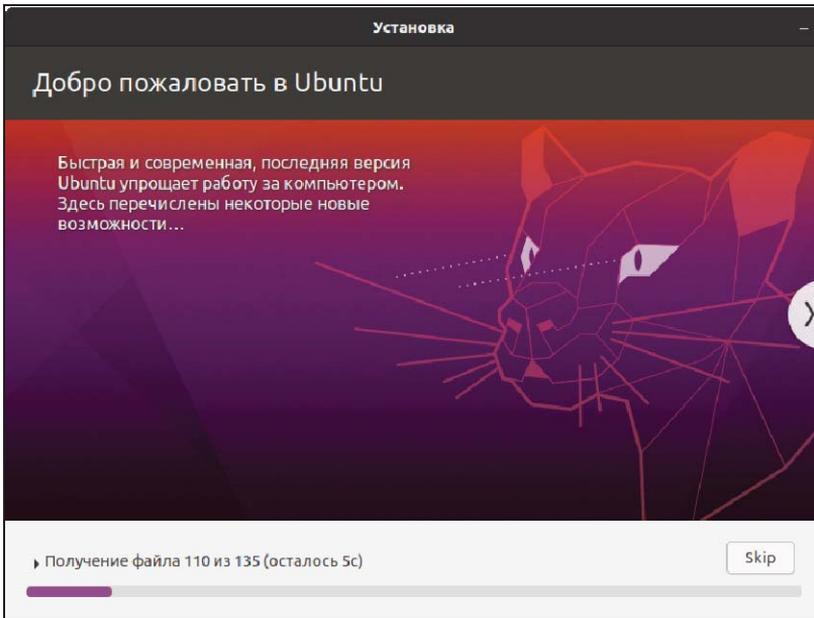


Рис. 4.13. Начался процесс установки ОС Ubuntu Linux

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

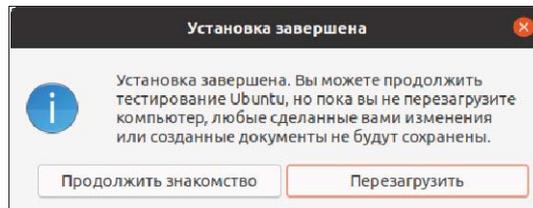


Рис. 4.14. Установщик сообщил об успешной установке ОС Ubuntu

Нажмите кнопку **Перезагрузить** и извлеките из компьютера установочный 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: ОС Ubuntu установлена, и теперь нам предстоит произвести ее первый запуск. Нажмите кнопку включения компьютера и дождитесь момента, когда после процедуры POST (Power-On Self-Test) появится окно меню загрузчика GRUB (рис. 5.1).

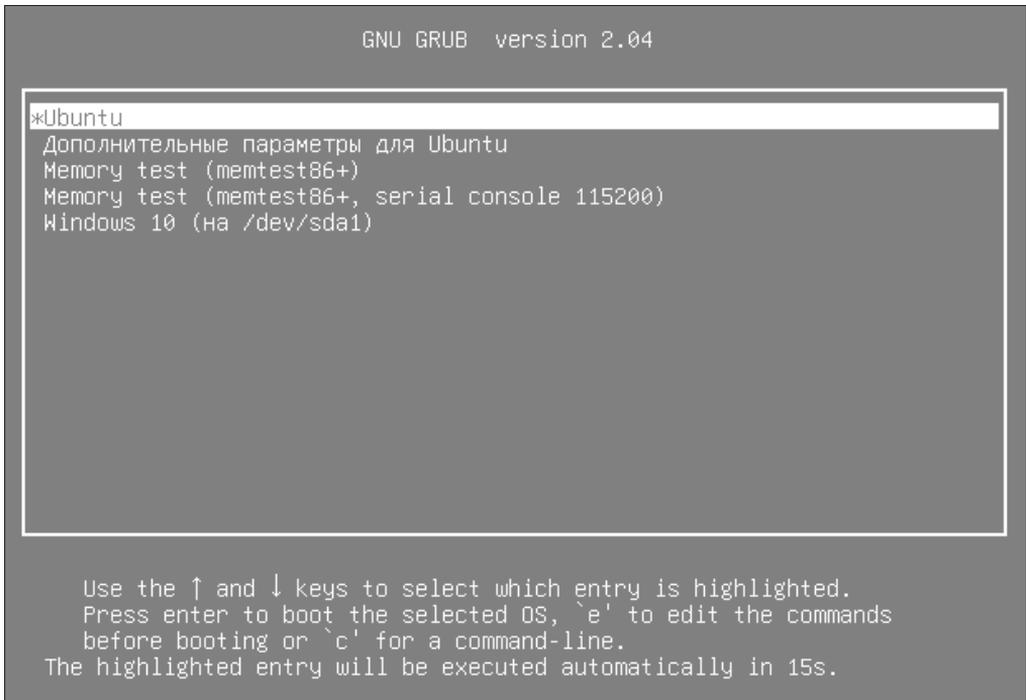


Рис. 5.1. Меню загрузчика GRUB

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

В нашем случае GRUB по умолчанию загружает ОС Ubuntu (строка, выделенная по умолчанию), если не был выбран другой вариант загрузки. Если в течение определенного времени не выбрать другой вариант загрузки, 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 последний вариант позволяет запустить ОС Windows 10, которая изначально присутствовала на компьютере до установки ОС Ubuntu Linux..

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

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

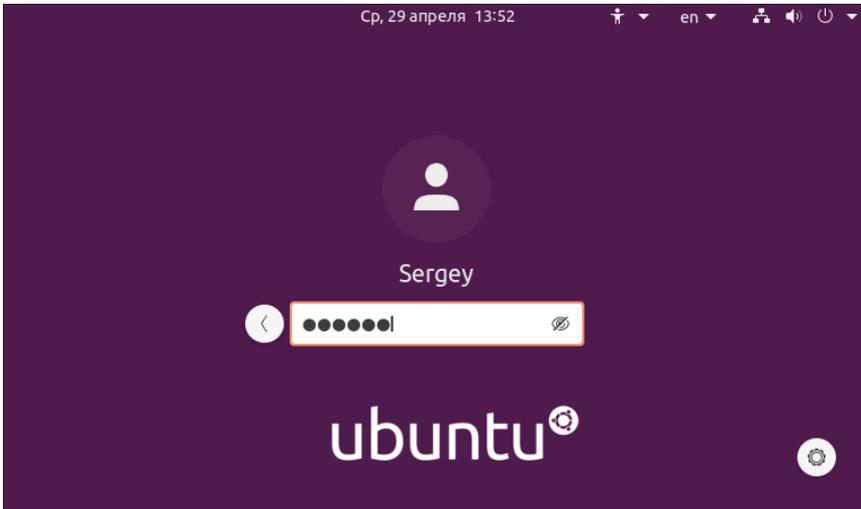


Рис. 5.2. ОС Ubuntu запрашивает пароль пользователя

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, по центру и в правом верхнем углу есть несколько значков:

- ◆ **Ср, 29 апреля 13:52** — отображение текущей даты и времени;
- ◆  — выбор специальных возможностей; например, можно включить экранную клавиатуру, увеличить текст или настроить контрастность;

- ◆ **en** — индикатор языка клавиатуры позволит сменить язык для требуемого ввода символов пароля;
- ◆  — группа значков, позволяющая отобразить или выбрать текущие настройки доступа к Интернету, настроить уровень громкости звуков для всей ОС в целом, управлять текущим сеансом (выключить компьютер или завершить текущий сеанс).

Ну что ж, введите свой пароль, заданный при установке ОС, и нажмите клавишу <Enter>. Вы попадете на рабочий стол ОС Ubuntu (рис. 5.3).

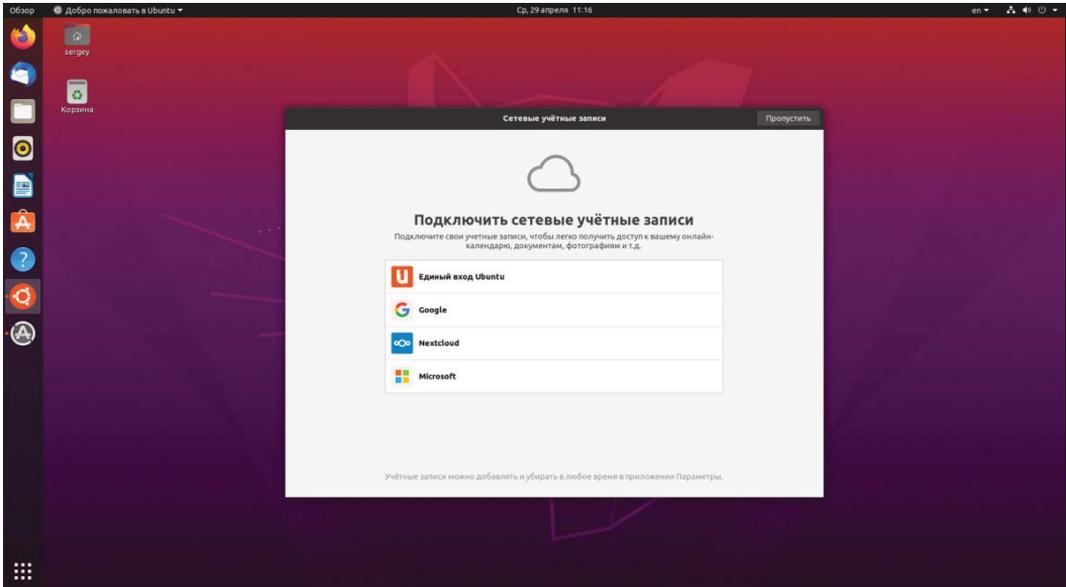


Рис. 5.3. Рабочий стол ОС Ubuntu

Рабочий стол выглядит примерно так же, как при запуске в Live-режиме. Можно заметить, что исчез значок **Install Ubuntu 20.04 LTS** и интерфейс ОС теперь на русском языке.

При первом запуске ОС Ubuntu предлагает нам подключить сетевую учетную запись, однако это можно сделать и позже в параметрах настройки системы. Поэтому нажмите кнопку **Пропустить** в верхней части окна. После этого вам будет предложено настроить возможность использования **Livepatch** (обновления системы без перезагрузки). Нажмите кнопку **Далее**, чтобы пропустить данный шаг. Затем ОС Ubuntu предложит настроить отправку отчетов об использовании системы. Выберите любой пункт, который вас устраивает, и нажмите кнопку **Далее**.

На следующем шаге система запросит разрешение на определение местоположения вашего компьютера. Для того чтобы пропустить этот запрос, также нажмите кнопку **Далее** для перехода к последнему шагу завершения предварительной настройки ОС Ubuntu Linux. На последнем шаге будет предложено установить необходимые нам приложения. Мы займемся этим позже, поэтому нажмите кнопку **Готово**.

Теперь 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 локальной сети (если беспроводное соединение не было установлено) .

#### **ПРИМЕЧАНИЕ**

Как мы уже знаем, данный значок входит в одну группу вместе с регулировкой громкости звуков и управлением питания компьютера.

Нажмите данную кнопку (группу значков), затем разверните группу **Wi-Fi-сеть** и выберите пункт **Выбрать сеть**. Отобразится список доступных подключений (рис. 5.4). В этом меню отображаются доступные беспроводные подключения, а также уже активные (установленные) соединения.

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

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

Для того чтобы отключить текущее беспроводное соединение, нажмите на группу со значком Wi-Fi, затем разверните установленное подключение, нажав по нему

левой кнопкой мыши, и выберите команду **Выключить**. Адаптер Wi-Fi будет отключен. Для того чтобы снова включить Wi-Fi, проделайте то же самое, что и при отключении, нажав на пункт меню **Включить**. Адаптер Wi-Fi будет включен, и сохраненное ранее подключение к беспроводной сети автоматически установится.

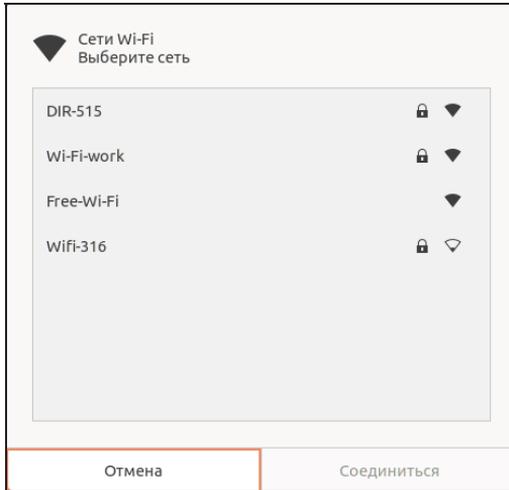


Рис. 5.4. Программа NetworkManager отображала список доступных подключений

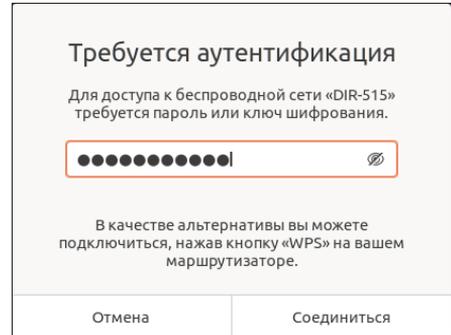


Рис. 5.5. Запрос пароля для подключения к сети Wi-Fi

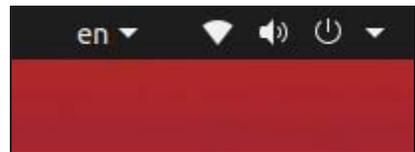


Рис. 5.6. Соединение Wi-Fi установлено

## Настройка подключений в окне настроек системы

Для настройки проводного или беспроводного подключения нужно щелкнуть на рабочем столе правой кнопкой мыши и выбрать пункт контекстного меню **Параметры**. Откроется окно настройки параметров системы (аналог Панели управления в ОС Windows). После этого в левой части окна вы можете выбрать требуемое вам беспроводное либо проводное соединение (рис. 5.7).

В нашем случае в окне редактора соединений имеется установленное проводное подключение. Если перейти на вкладку **Wi-Fi**, мы увидим сохраненное ранее подключение к беспроводной сети.

Для того чтобы добавить новое проводное соединение, нужно нажать кнопку с изображением знака +.

Редактирование параметров установленного проводного соединения осуществляется путем нажатия на кнопку с изображенной шестеренкой .

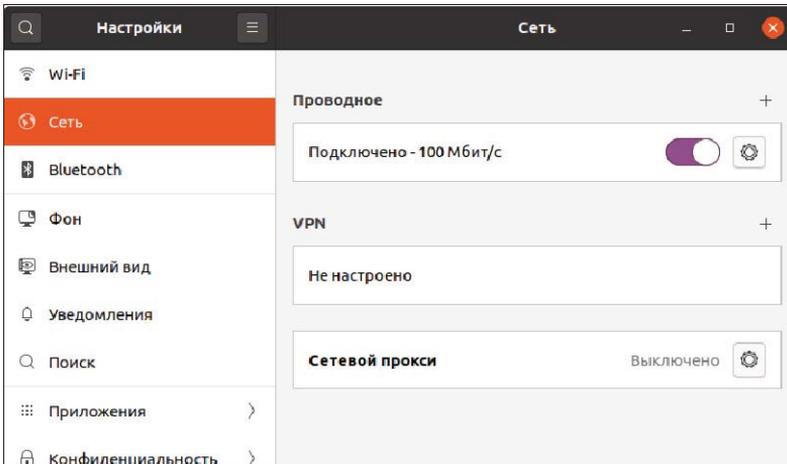


Рис. 5.7. Редактор сетевых соединений в параметрах системы

## 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 щелкнув правой кнопкой по рабочему столу и выбрав команду **Открыть в терминале**. Откроется окно программы Терминал, в котором и нужно ввести приведенную выше команду (рис. 5.8).

 A screenshot of a terminal window titled "sergey@Test-PC: ~". The terminal shows the command `lspci -knn | grep "Eth\|Net" -A2` and its output. The output lists two network controllers: a Qualcomm Atheros AR8151 v2.0 Gigabit Ethernet controller and a Ralink corp. RT5360 Wireless 802.11n 1T/1R network controller. The terminal text is as follows:
 

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

sergey@Test-PC:~$ lspci -knn | grep "Eth\|Net" -A2
01:00.0 Ethernet controller [0200]: Qualcomm Atheros AR8151 v2.0 Gigabit Ethernet
t [1969:1083] (rev c0)
Subsystem: ASUSTeK Computer Inc. AR8151 v2.0 Gigabit Ethernet [1043:847e
]
Kernel driver in use: atl1c
Kernel modules: atl1c
03:01.0 Network controller [0280]: Ralink corp. RT5360 Wireless 802.11n 1T/1R [1
814:5360]
Subsystem: D-Link System Inc DWA-525 Wireless N 150 Desktop Adapter (rev
.A2) [1186:3c05]
Kernel driver in use: rt2800pci
sergey@Test-PC:~$
```

Рис. 5.8. Терминал вывел список установленных сетевых адаптеров

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

```
01:00.0 Ethernet controller [0200]: Qualcomm Atheros AR8151 v2.0 Gigabit Ethernet [1969:1083] (rev c0)
```

...

```
03:01.0 Network controller [0280]: Ralink corp. RT5360 Wireless 802.11n 1T/1R [1814:5360]
```

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

- ◆ Atheros AR8151 драйверы для Ubuntu;
- ◆ Ralink RT5360 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.

Для этого откройте меню настроек, щелкнув правой кнопкой по рабочему столу и выбрав пункт меню **Параметры**. Перейдите на вкладку **Сеть** и в разделе **Проводное** нажмите кнопку с изображенным знаком + (рис. 5.9).

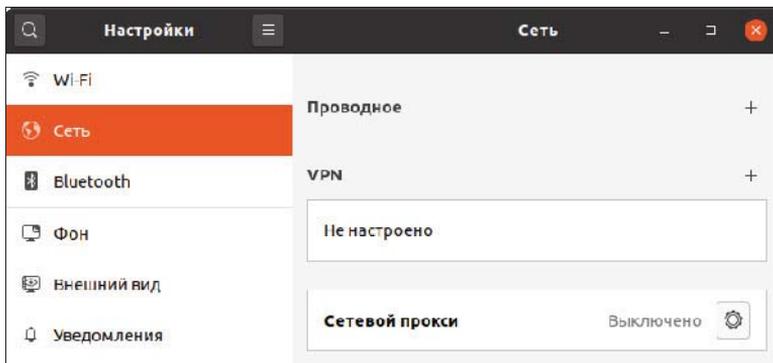


Рис. 5.9. Создание нового подключения

Откроется окно создания нового профиля проводного соединения (рис. 5.10).

Для того чтобы создать профиль подключения, нажмите кнопку **Добавить**. Профиль проводного подключения будет создан.

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/PPPoE

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 (модеме) 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.

В ранних версиях ОС Ubuntu Linux настройка DSL-подключения производилась напрямую из программы NetworkManager. Однако время не стоит на месте, и, как показывает практика, DSL-подключение становится такой редкостью, как, например, использование дискет. По этой причине, за редкой необходимостью настройки такого подключения, разработчики ОС Ubuntu убрали этот пункт из меню настроек подключения к сети. В случае если такое подключение необходимо создать, эта возможность все еще существует.

Итак, для начала вставьте Ethernet-кабель в соответствующий разъем на сетевой карте вашего компьютера и откройте Терминал, нажав сочетание клавиш `<Ctrl>+<Alt>+<T>`, а затем выполните команду:

```
nmcli con edit type pppoe con-name "Connection DSL"
```

Запустится интерактивный режим создания DSL-подключения (рис. 5.11).

В этом интерактивном режиме выполните следующую команду:

```
set pppoe.username user0342342
```

```

sergey@Test-PC: ~
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

sergey@Test-PC:~$ nmcli con edit type pppoe con-name "Connection DSL"

===| интерактивный редактор соединений nmcli |===

Добавление нового соединения «pppoe»

Для просмотра доступных команд введите «help» или «?»
Введите «print», чтобы показать все свойства соединения.
Для просмотра описания свойства введите «describe [<параметр>.<свойство>]».

Разрешается изменить следующие параметры: connection, pppoe, 802-3-ethernet (
ethernet), ppp, 802-1x, ethtool, match, ipv4, ipv6, tc, proxy
nmcli>

```

Рис. 5.11. Создание DSL-соединения

Вместо строки `user0342342` укажите ваш логин (имя пользователя), который вам выдал ваш провайдер.

Для сохранения внесенных изменений введите соответствующую команду:

```
save
```

Для подтверждения изменений нажмите клавишу `<Y>`.

Соединение будет создано (рис. 5.12).

Для выхода из режима создания DSL-подключения выполните команду:

```
quit
```

```

sergey@Test-PC: ~
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

sergey@Test-PC:~$ nmcli con edit type pppoe con-name "Connection DSL"

===| интерактивный редактор соединений nmcli |===

Добавление нового соединения «pppoe»

Для просмотра доступных команд введите «help» или «?»
Введите «print», чтобы показать все свойства соединения.
Для просмотра описания свойства введите «describe [<параметр>.<свойство>]».

Разрешается изменить следующие параметры: connection, pppoe, 802-3-ethernet (ethe
rnet), ppp, 802-1x, ethtool, match, ipv4, ipv6, tc, proxy
nmcli> set pppoe.username user0342342
nmcli> save
Вы собираетесь сохранить соединение с установленным параметром «autoconnect=yes»,
что может привести к его немедленной активации.
Сохранить? (yes/no) [yes] y
Соединение «Connection DSL» (3c578b99-73b2-464e-8185-413358507779) сохранено.
nmcli>

```

Рис. 5.12. Сохранение параметров DSL-соединения

Теперь созданное DSL-подключение будет отображаться в списке проводных соединений в окне параметров настроек системы. Дальнейшее редактирование созданного подключения осуществляется там же.

В случае, когда DSL-подключение более не нужно или вы ошибочно создали слишком много подключений и хотите удалить лишние, запустите редактор сетевых соединений, выполнив следующую команду в Терминале:

```
nm-connection-editor
```

Откроется окно редактирования сетевых соединений (рис. 5.13).

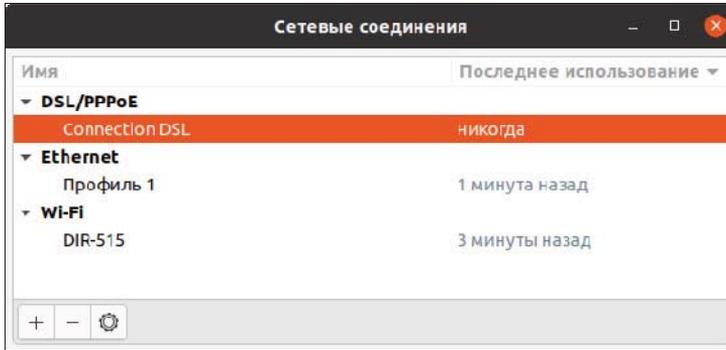


Рис. 5.13. Список сетевых соединений

В этом окне перечислены все созданные подключения, в том числе установленные подключения к сети Wi-Fi.

Для того чтобы удалить необходимое подключение, выделите его и внизу окна нажмите кнопку с изображением знака "минус". Появится окно подтверждения удаления. Нажмите соответствующую кнопку или кнопку **Отменить** для отмены удаления соединения.

Стоит отметить, что в окне **Сетевые соединения** вы также можете вызвать окно редактора настроек того или иного подключения, нажав кнопку с изображением шестеренки в самом низу окна.

Также в этом окне можно отследить статус созданного подключения: использовалось оно или нет. Данный статус отображается напротив самого подключения и позволяет косвенно понять, было ли на самом деле установлено подключение к Интернету или только создано, но не установлено.

## Особый случай подключения через DSL/PPPoE

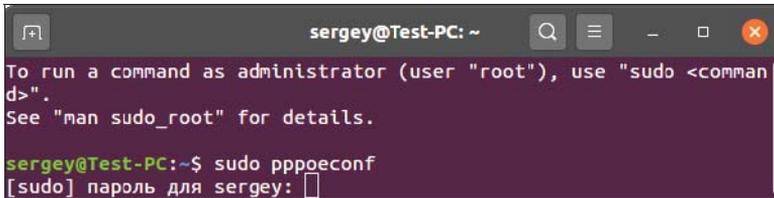
Встречаются ситуации, когда приведенный выше способ создания DSL-соединения не работает. То есть вы создали соединение, но оно отказывается работать. В этом случае можно воспользоваться альтернативным способом настройки соединения. Для этого запустите Терминал и выполните следующую команду:

```
sudo pppoeconf
```

**ПРИМЕЧАНИЕ**

Запустить Терминал можно, нажав комбинацию клавиш <Ctrl>+<Alt>+<T> либо щелкнув на рабочем столе правой кнопкой мыши и выбрав команду **Открыть в терминале**.

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



```
sergey@Test-PC: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
sergey@Test-PC:~$ sudo pppoeconf  
[sudo] пароль для sergey: 
```

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.

Здесь нас ожидает одна неприятность, которая заключается в том, что пакет (приложение) `pppoeconf` больше не поставляется вместе с установочным образом дистрибутива ОС Ubuntu Linux. Об этом будет свидетельствовать сообщение Терминала `"sudo: pppoeconf: команда не найдена"`. Однако этот пакет можно установить с помощью команды:

```
sudo apt install pppoeconf
```

Выполните данную команду, подтвердив установку пакета нажатием клавиши <Y> в ответ на запрос уставщика. После окончания установки повторите запуск команды `sudo pppoeconf`.

Запустится 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).

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

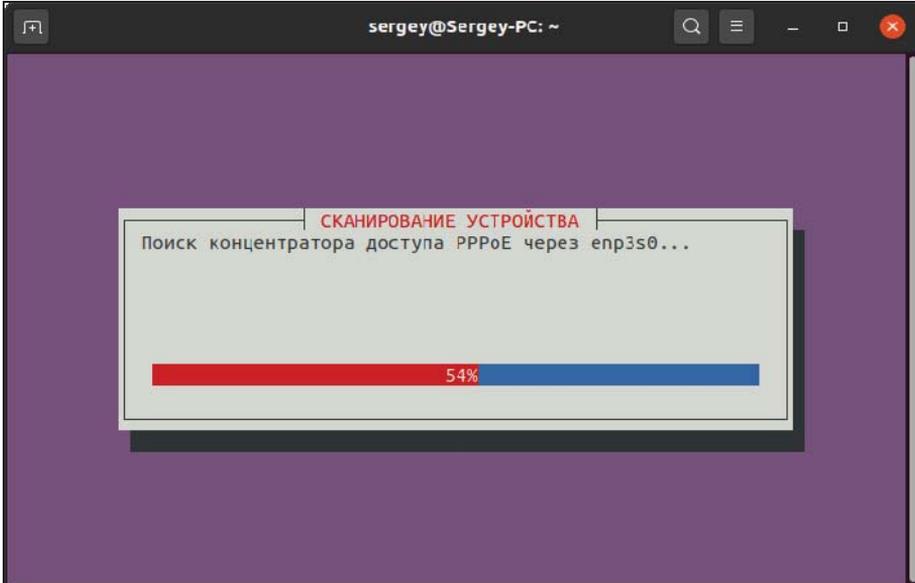


Рис. 5.15. Программа настройки PPPoE-соединений

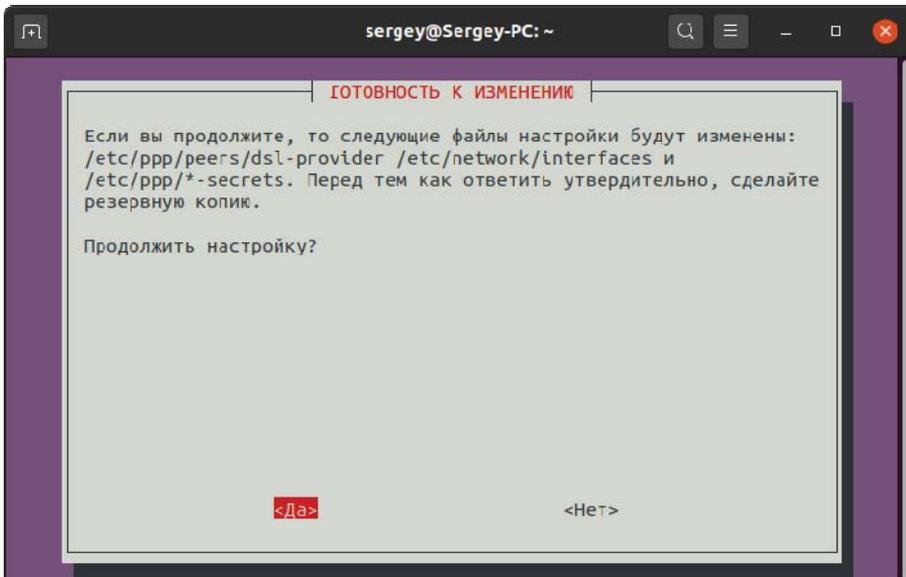


Рис. 5.16. Программа rppoesconf сообщила о готовности к внесению изменений

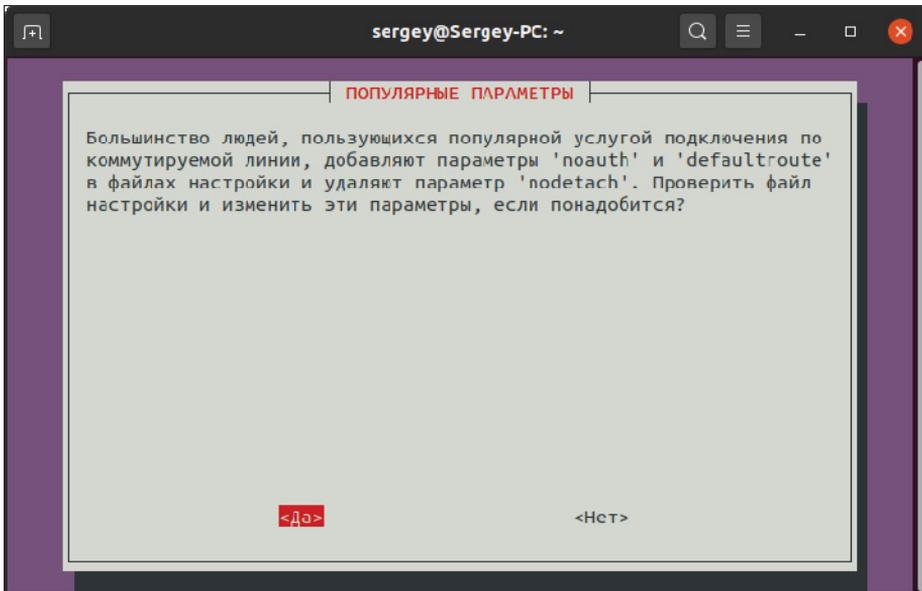


Рис. 5.17. Использование дополнительных опций для соединения

по умолчанию) и **noauth** (не требовать аутентификации от PPPoE-сервера) — рис. 5.17.

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>**.

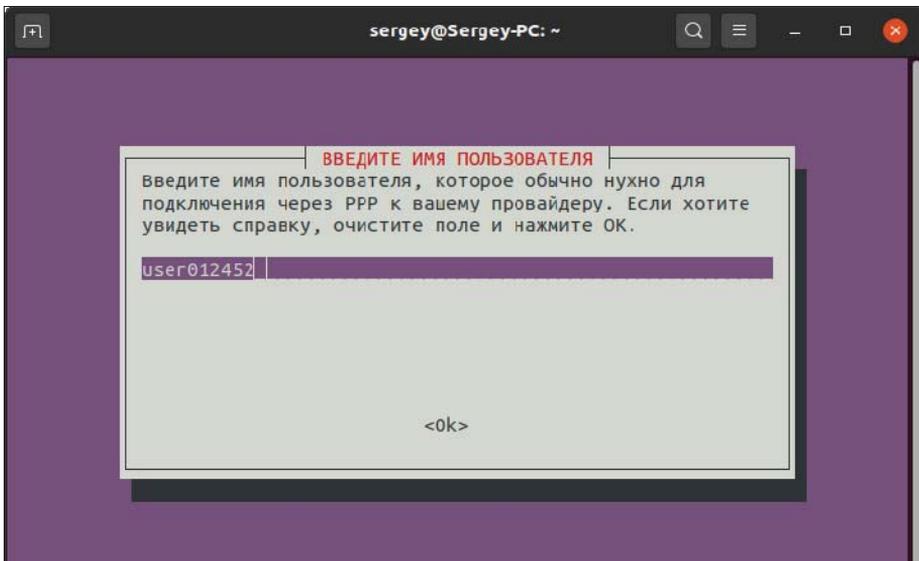


Рис. 5.18. Программа настройки запросила логин для подключения

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

Символы пароля отображаются во избежание ошибок при вводе пароля. Будьте внимательны, т. к. в случае ошибочного задания пароля вам придется заново производить настройку. Введите ваш пароль и нажмите клавишу <Enter>.

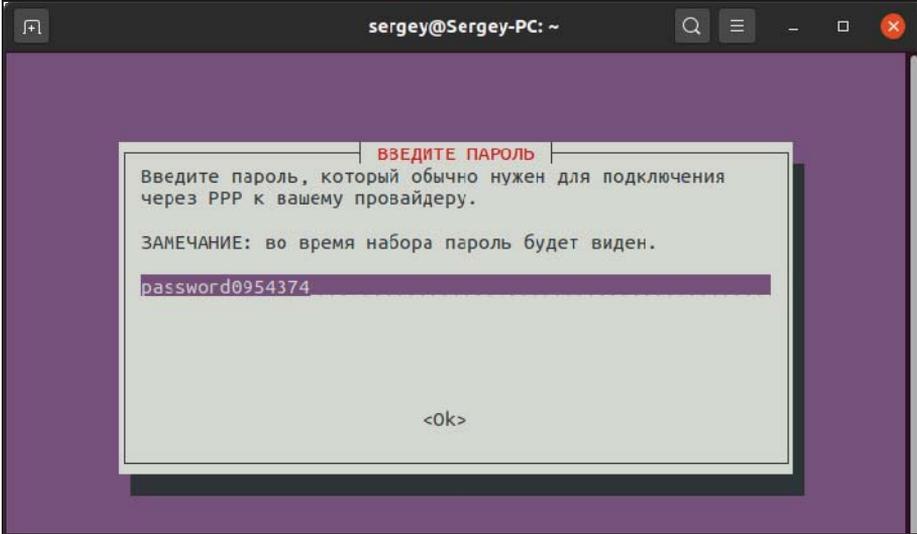


Рис. 5.19. Запрос ввода пароля

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

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

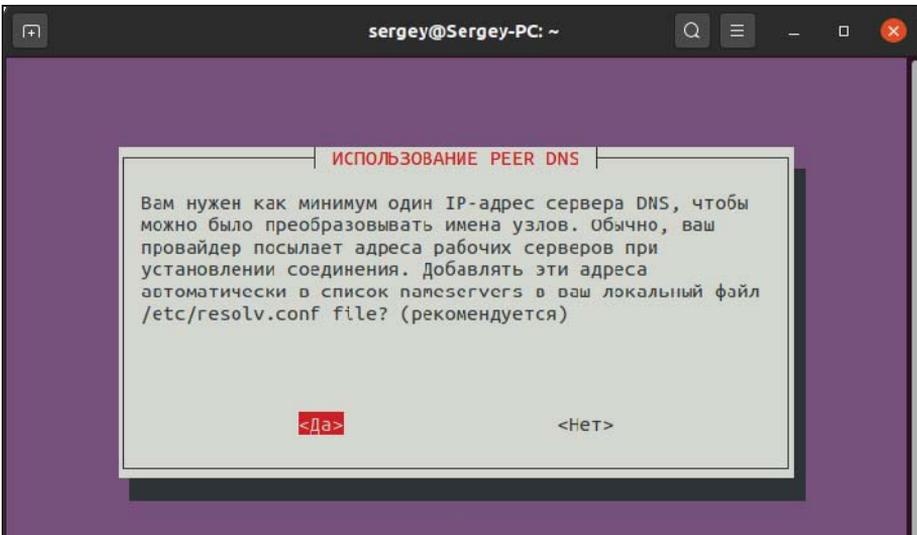


Рис. 5.20. Программа настройки запрашивает добавление адреса DNS-сервера

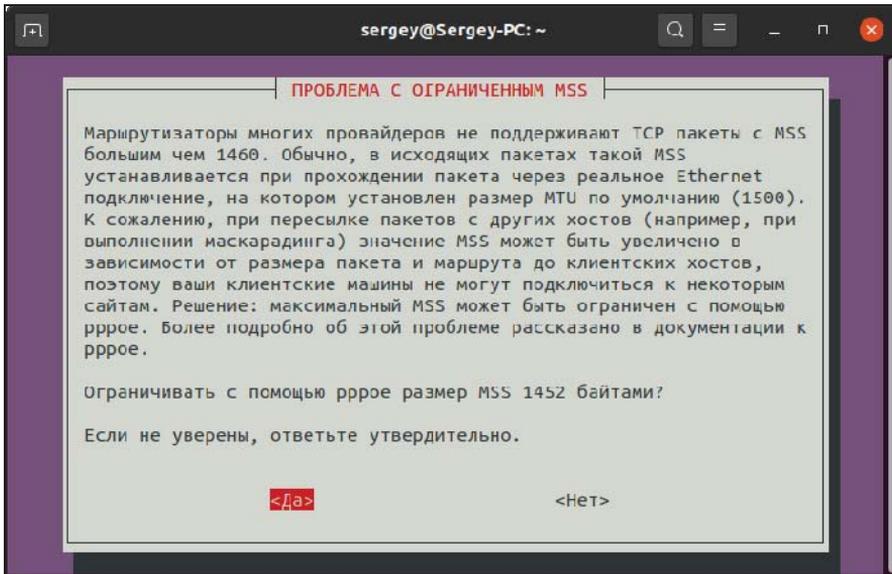


Рис. 5.21. Ограничение размера MSS

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).

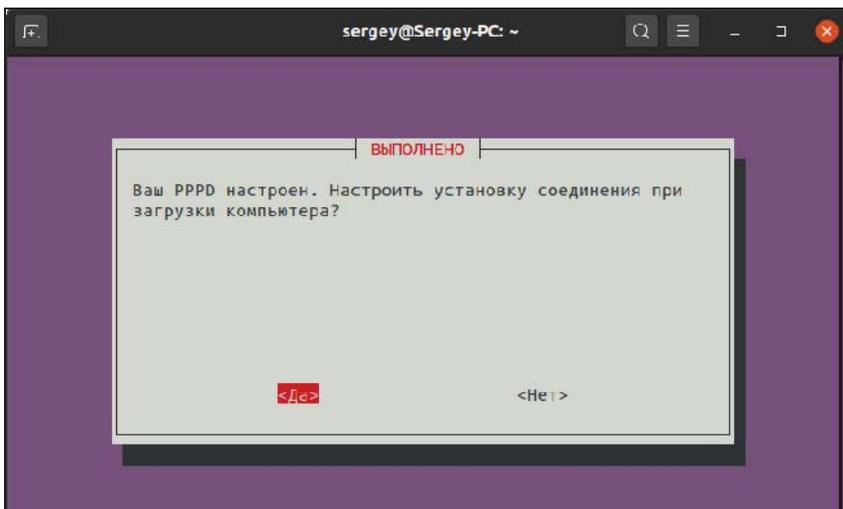


Рис. 5.22. Программа настройки предлагает устанавливать соединение при загрузке ОС

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

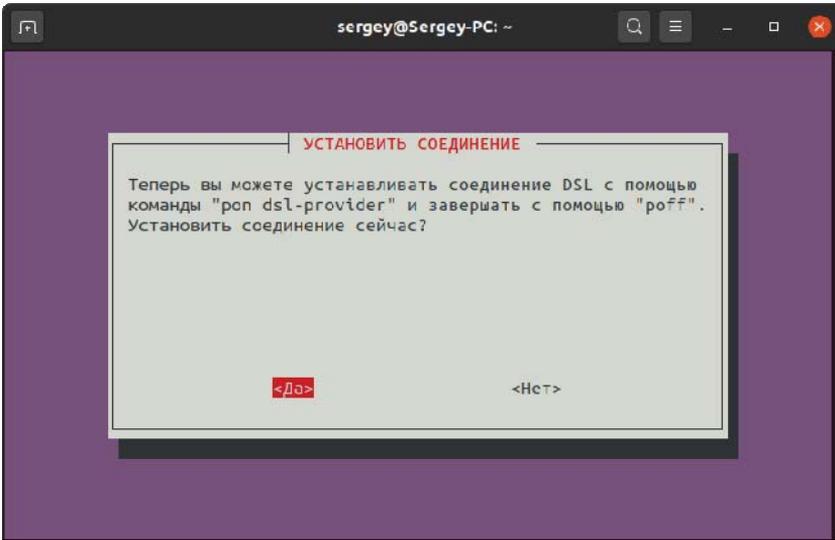


Рис. 5.23. Запрос установки соединения

**ПРИМЕЧАНИЕ**

В дальнейшем при 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).

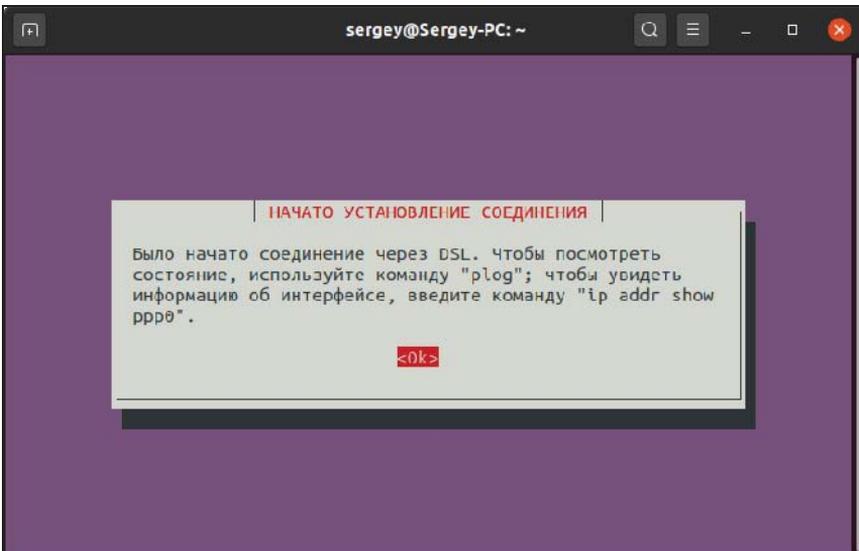


Рис. 5.24. Соединение установлено

Нажмите клавишу <Enter> для завершения настройки.

На этом настройка DSL/PPPoE-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, откройте Терминал и запустите окно сетевых соединений, выполнив уже знакомую нам команду:

```
nm-connection-editor
```

Нажмите на кнопку с изображением знака "плюс" и в открывшемся выборе соединения в раскрывающемся списке выберите вариант **Мобильная связь** (рис. 5.25).

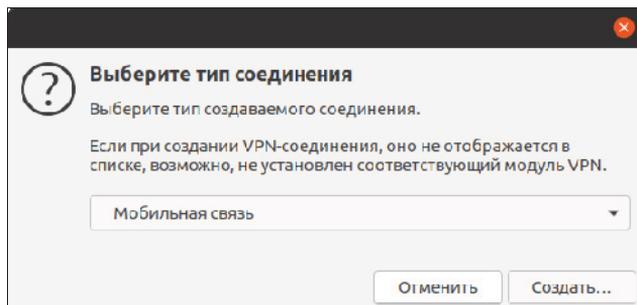


Рис. 5.25. Создание мобильного подключения

Нажмите кнопку **Создать...** Откроется окно создания мобильного подключения (рис. 5.26).

В раскрывающемся списке **Создать соединение для этого устройства мобильной связи** 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 очередном этапе настройки предлагается выбрать страну оператора. Так как мы живем в России, соответственно выбираем этот пункт (рис. 5.27).

Нажмите кнопку **Следующая**, чтобы выбрать самого оператора сотовой связи (рис. 5.28).

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

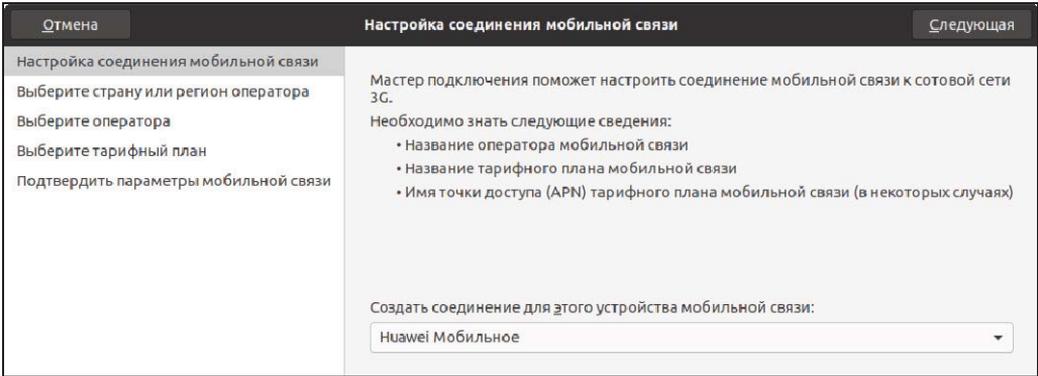


Рис. 5.26. Создание соединения через USB-модем

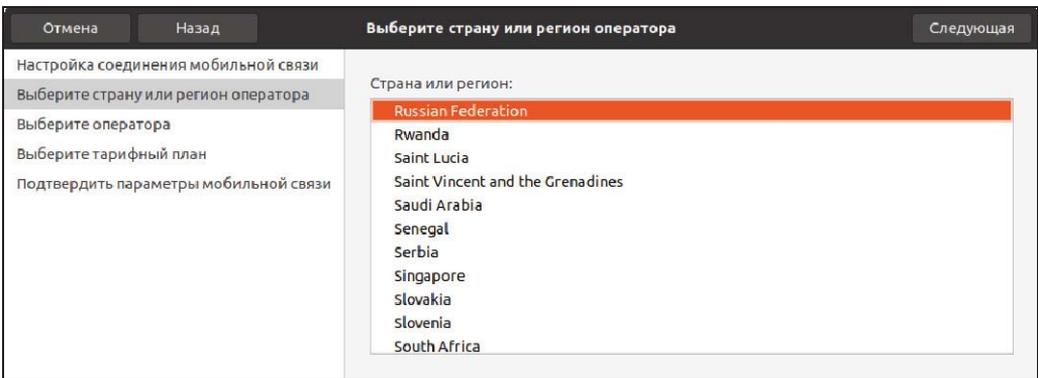


Рис. 5.27. Выбор страны оператора

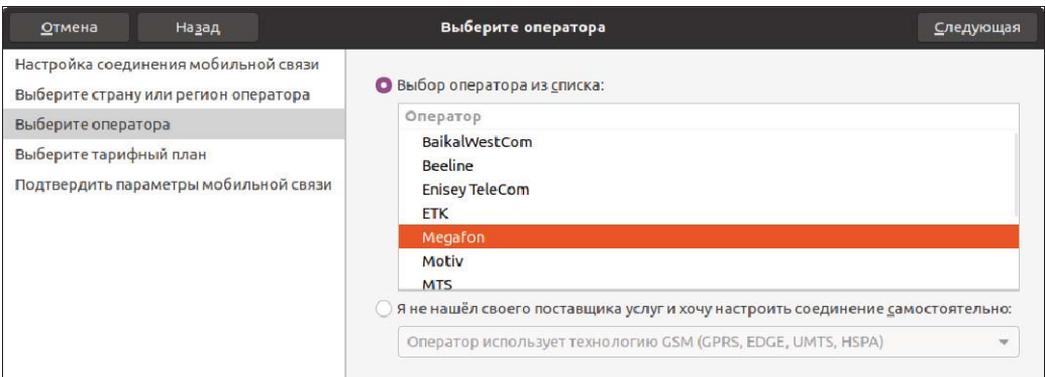


Рис. 5.28. Страница выбора оператора сотовой связи

Далее будет предложено выбрать тарифный план и точку доступа (APN) (рис. 5.29). Менять тарифный план и точку доступа не имеет смысла, т. к. точка доступа указана верно. Нажмите кнопку **Следующая**.

На следующем этапе предлагается применить сделанные настройки (рис. 5.30).

Отмена Назад **Выберите тарифный план** Следующая

Настройка соединения мобильной связи

Выберите страну или регион оператора

Выберите оператора

**Выберите тарифный план**

Подтвердить параметры мобильной связи

Выберите тарифный план:  
RUS

Выберите точку доступа (APN) этого тарифного плана:  
internet

**!** Внимание! Выбор неверного тарифного плана может привести к применению неправильного тарифа для мобильной связи или невозможности доступа в сеть.  
Если не уверены, какой тарифный план выбрать, уточните APN у оператора мобильной связи.

Рис. 5.29. Выбор точки доступа и тарифного плана

Отмена Назад **Подтвердить параметры мобильной связи** Применить

Настройка соединения мобильной связи

Выберите страну или регион оператора

Выберите оператора

Выберите тарифный план

**Подтвердить параметры мобильной связи**

Соединение мобильной связи настроено со следующими параметрами:  
Устройство: Huawei Мобильное  
Оператор: MegaFon, Russian Federation  
Тарифный план: RUS  
APN: internet

Рис. 5.30. Подтверждение заданных параметров

Изменение MegaFon RUS 1

Название соединения MegaFon RUS 1

Основное **Мобильная связь** Параметры PPP Прокси Параметры IPv4 Параметры IPv6

**Основной**

Number \*99#

Имя пользователя gdata

Пароль .....

**Дополнительно**

APN internet Изменить...

ИД сети

Включать роуминг, если домашняя сеть недоступна

PIN

Показывать пароли

Отменить Сохранить

Рис. 5.31. Дополнительные настройки мобильного Интернета

Нажмите кнопку **Применить**, и откроется последнее окно, в котором можно задать дополнительные настройки мобильного Интернета (рис. 5.31): 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 группу значков в правом верхнем углу рабочего стола или по команде `nm-connection-editor`.

## Возможные проблемы с созданием подключений

Бывают такие ситуации, когда не удастся установить соединение даже в окне редактора сетевых соединений. Причина кроется в том, что в файле `/etc/network/interfaces` появляются неоднозначные записи, которые вводят в замешательство программу по установке интернет-соединения. Так, следующее содержимое 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)
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto dsl-provider
iface dsl-provider inet ppp
pre-up /bin/ip link set enp3s0 up # line maintained by pppoeconf
provider dsl-provider

auto enp3s0
iface enp3s0 inet manual
```

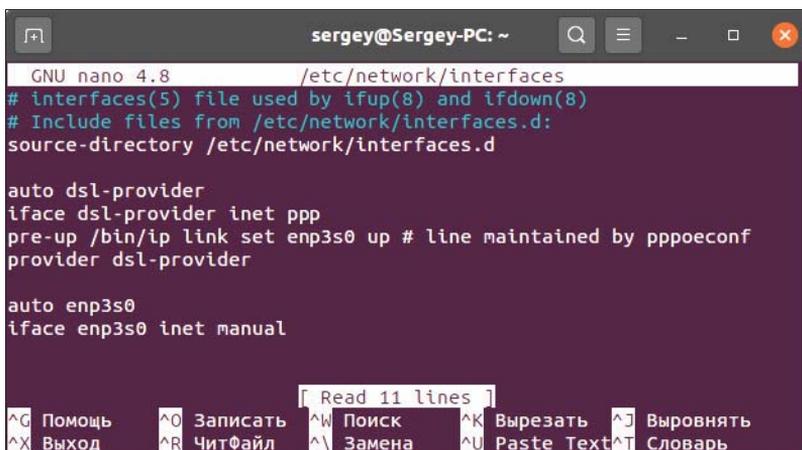
Для того чтобы исправить ситуацию, необходимо вручную отредактировать данный файл, но перед этим следует удалить все созданные соединения.

Запустите редактор сетевых соединений программы NetworkManager, выполнив команду `nm-connection-editor` в Терминале, и по очереди удаляйте каждое соединение. Затем в Терминале выполните следующую команду:

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

Вы увидите запрос вашего пароля, после ввода которого файл `/etc/network/interfaces` станет доступен для редактирования (рис. 5.32).

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.



```
GNU nano 4.8 /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto dsl-provider
iface dsl-provider inet ppp
pre-up /bin/ip link set enp3s0 up # line maintained by pppoeconf
provider dsl-provider

auto enp3s0
iface enp3s0 inet manual
```

Рис. 5.32. Файл /etc/network/interfaces открыт для редактирования

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

```
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d
```

**ПРИМЕЧАНИЕ**

Для перемещения по файлу используйте клавиши-стрелки, а для редактирования — соответствующие клавиши <Backspace> или <Del>.

После редактирования необходимо сохранить файл, нажав сочетание клавиш <Ctrl>+<O> (либо <F3>), а затем клавишу <Enter>. И выйдите из редактора Nano, нажав сочетание клавиш <Ctrl>+<X>.

После этого закройте Терминал и создайте нужное вам подключение.

## Выводы

Мы не только обсудили наиболее популярные типы подключений к Интернету, но и научились создавать такие подключения. Кроме того, мы 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 в Интернете.



## 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 **Запуск приложений** на панели задач и в строке поиска введите *обновление*. В ответ на это в результатах поиска появится программа Обновление приложений (рис. 6.1).

Запустите программу Обновление приложений. 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 также 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 **Подтвердить** (рис. 6.3).

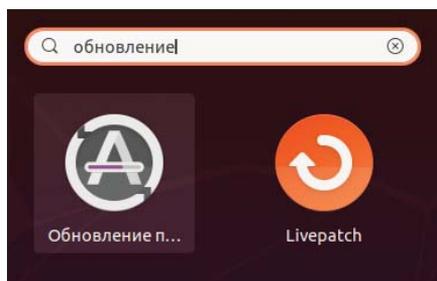


Рис. 6.1. Поиск программы Обновление приложений

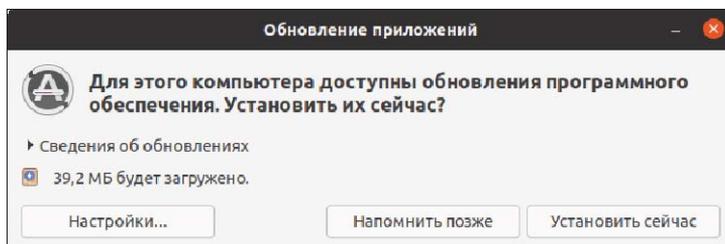


Рис. 6.2. Программа Обновление приложений сообщила о готовности обновлений к загрузке

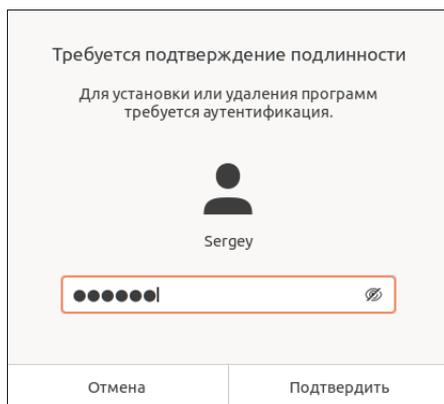


Рис. 6.3. Запрос ввода пароля

#### 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 — дорогое удовольствие.

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

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).

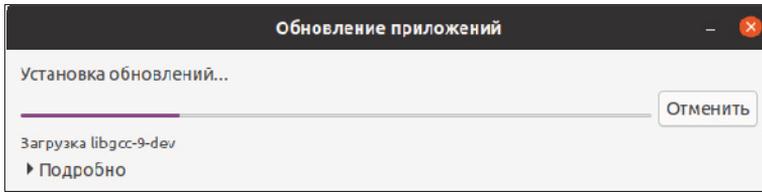


Рис. 6.4. Процесс загрузки обновлений

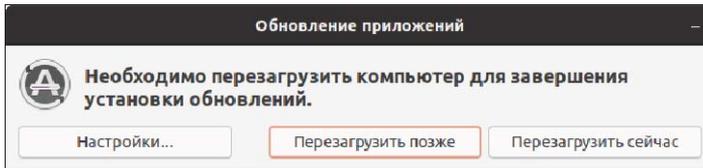


Рис. 6.5. Установка обновлений завершена

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).

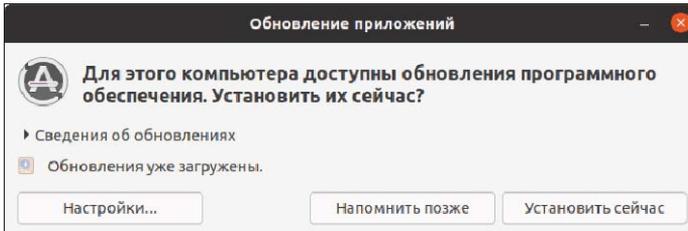


Рис. 6.6. Доступны обновления программного обеспечения

Для того чтобы своевременно получать актуальные пакеты обновлений, нужно последовательно выполнить в Терминале такие команды:

```
sudo apt update
sudo apt upgrade
```

При вводе первой команды появится запрос на ввод вашего пароля и будут загружены списки пакетов, содержащихся в репозиториях, а при выполнении второй команды от вас потребуются подтверждение ваших намерений для установки найденных обновлений. 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 пользователя. Разработчики дистрибутива Ubuntu не имеют возможности включить в загрузочный дистрибутив все языки (иначе загрузочный размер дистрибутива будет огромным), да это и не нужно, потому что всегда имеется возможность загрузить локализованные пакеты из Интернета.

Для загрузки языковых пакетов щелкните правой кнопкой мыши на рабочем столе и в контекстном меню выберите пункт **Параметры**. Откроется окно **Настройки** (рис. 6.7).

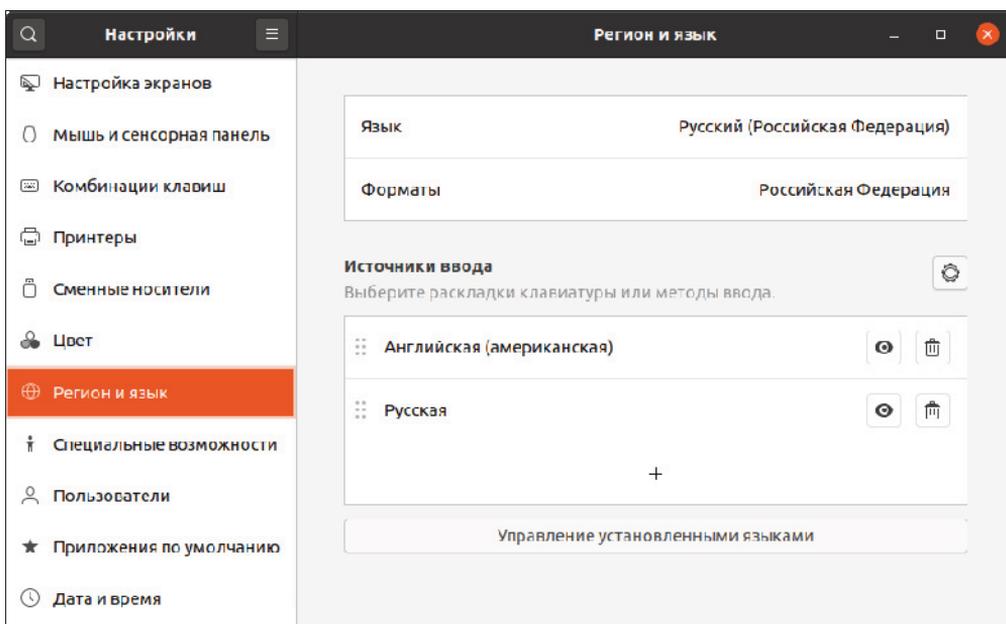


Рис. 6.7. Окно настройки параметров системы

Данное окно напоминает собой Панель управления, которую мы привыкли видеть в ОС Windows. В нем нас интересует раздел **Регион и язык**, в котором необходимо найти кнопку **Управление установленными языками**. Нажмите эту кнопку. 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).

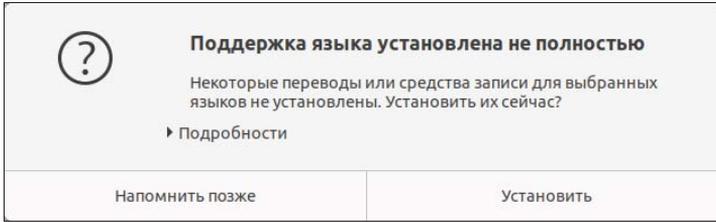


Рис. 6.8. Требуется установка языковых пакетов

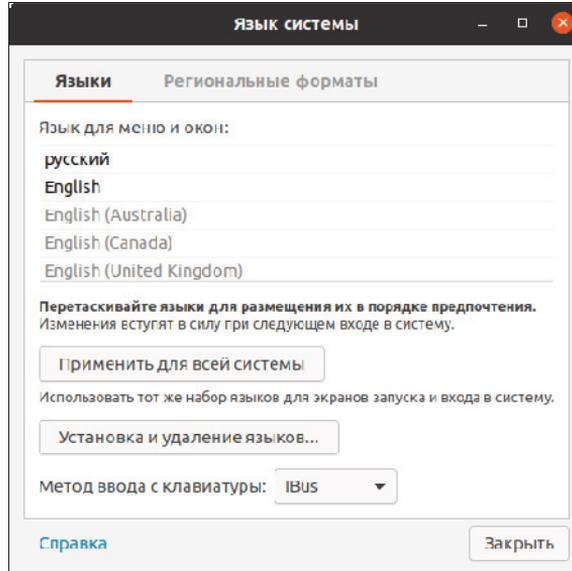


Рис. 6.9. Установка языковых пакетов завершена

**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 этого не требуется, если вы не используете язык, отличный от русского и английского.

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

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 группу значков системного меню (изображение тумблера) и в разворачивающемся списке выберите пункт **Выключить/Завершить сеанс**, а затем пункт **Завершить сеанс**. После входа в систему интерфейс ОС уже будет на русском языке.

## **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, теперь вы сможете самостоятельно менять язык интерфейса операционной системы на необходимый вам, а также устанавливать нужные 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 GNOME.

GNOME — это свободная графическая оболочка среды рабочего стола, используемая практически во всех UNIX-подобных операционных системах. Все, что вы видите при загрузке ОС Ubuntu, — это и есть графическая оболочка GNOME. По сути, графическую оболочку GNOME можно сравнить с Проводником в ОС Windows.

Помимо графической оболочки GNOME, в ОС на базе Linux используются такие оболочки, как KDE, XFCE4, LXDE и др.

### **ПРИМЕЧАНИЕ**

Графическую оболочку GNOME еще называют графическим окружением рабочего стола.

Так, вместо графической среды GNOME можно использовать любую графическую оболочку, которая совместима вашей ОС, например Cinnamon, MATE или XFCE4, либо оставить графическую среду GNOME без каких-либо изменений.

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

## Рабочий стол в GNOME

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

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

- ◆ панель основного меню;
- ◆ панель задач;
- ◆ рабочее место.

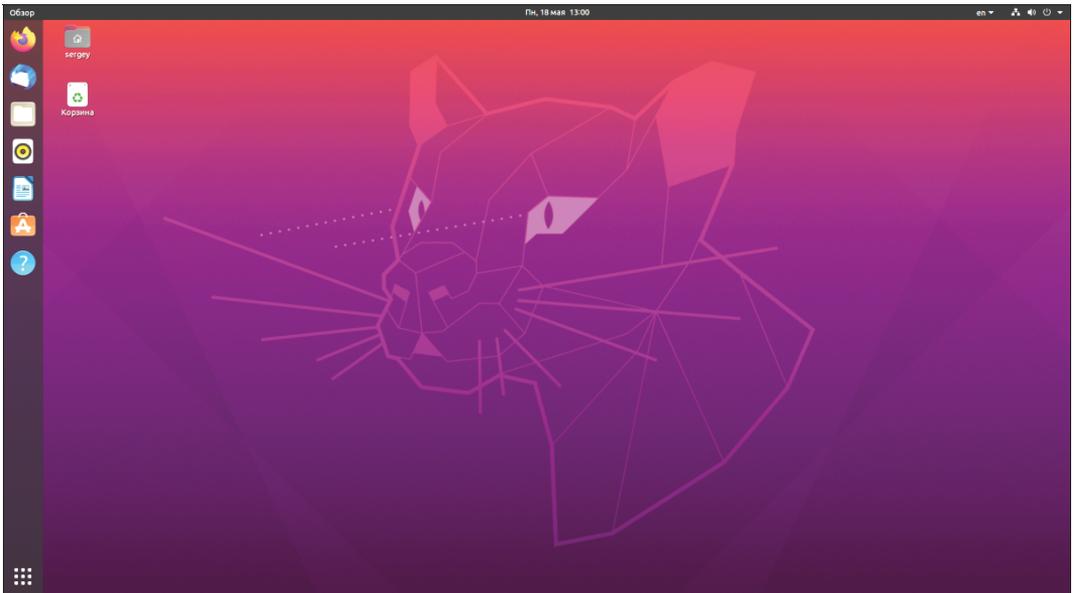


Рис. 7.1. Рабочий стол в графической оболочке GNOME

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

## Панель основного меню

*Панель основного меню* в ОС Ubuntu — это уже знакомая нам темная полоска вверху экрана (рис. 7.2), которая очень похожа на панель задач в ОС Windows, но с небольшими концептуальными отличиями.

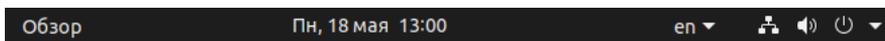


Рис. 7.2. Панель меню в графической оболочке GNOME

Однако панель основного меню устроена немного иначе и на первых порах использования кажется неудобной и непонятной. Давайте внесем ясность в то, как она работает в графической оболочке GNOME.

Левая часть панели меню задействована для нужд программы, которая запущена в тот или иной момент времени. Это означает, что в левой части меню, там, где мы привыкли видеть надпись "Обзор", во время работы с какой-либо программой, рядом будет располагаться панель меню самой программы. Звучит немного запутанно? 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.

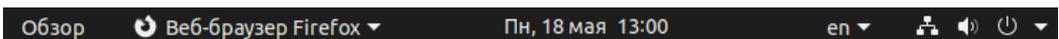


Рис. 7.3. Заголовок активного приложения

Теперь рядом с надписью "Обзор" появилась надпись "Веб-браузер Firefox", но и это еще не все. Если нажать на надпись **Обзор**, то отобразятся все открытые на данный момент приложения (рис. 7.4).

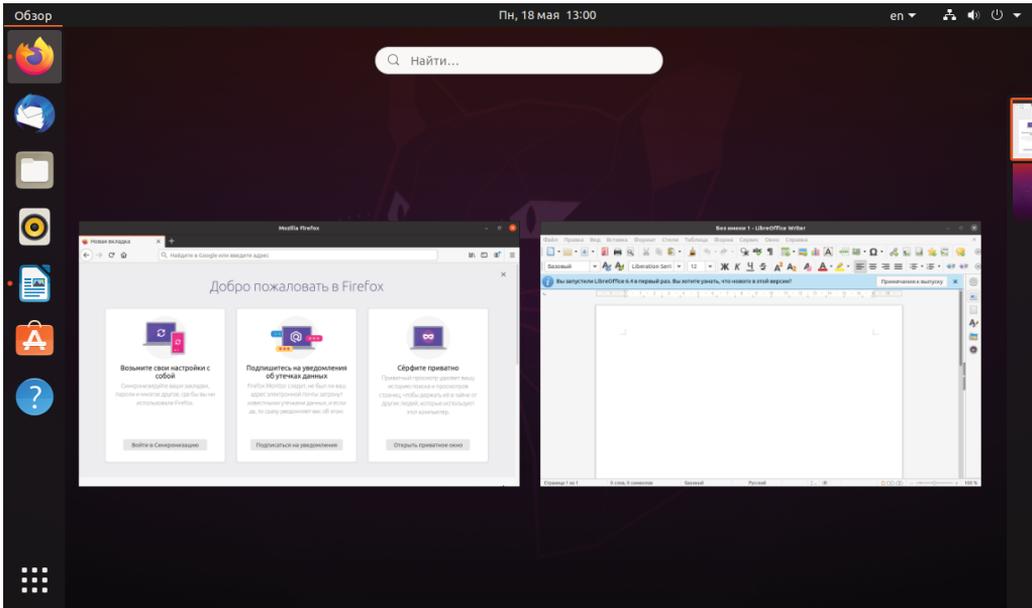


Рис. 7.4. Окна приложений

Надпись, а точнее кнопка **Обзор**, практически аналогична кнопке **Представление задач** в ОС Windows. С помощью этой кнопки очень легко можно переключаться между открытыми окнами в ОС Ubuntu Linux. Самым непривычным моментом для новичков на первых порах знакомства с ОС Ubuntu будет то, что данная панель расположена сверху, а не внизу экрана, как в ОС Windows.

Кнопки управления окном той или иной программы расположены ниже и выглядят так же, как и в ОС Windows. Например, первой идет кнопка *сворачивания окна*, затем кнопка *разворачивания окна* и только потом кнопка *закрывтия окна*.

По центру панели расположено меню уведомлений, которое совмещает в себе календарь с отображением даты и времени.

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



Рис. 7.5. Меню состояния

Данную панель часто называют *индикаторами*. 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 раскладки клавиатуры, который позволяет быстро ее переключить;
- ◆  — индикатор подключения к проводной сети. С этим значком мы уже познакомились при создании проводного соединения в программе NetworkManager. Данный значок присутствует при наличии проводного подключения;
- ◆  — индикатор подключения к беспроводной сети Wi-Fi. При установленном проводном подключении данный значок отсутствует;
- ◆  — индикатор звукового меню, который предоставляет быстрый доступ к управлению громкостью звука;
- ◆  — индикатор батареи, который присутствует на портативных устройствах, таких как ноутбук или планшет, и показывает текущее состояние батареи;
- ◆  — индикатор, позволяющий понять, что компьютер в данный момент работает от сети (не подключен к аккумулятору). Данный значок присутствует на настольных компьютерах. В портативных устройствах этот значок изображен в виде батареи;
- ◆  — значок разворачивающегося списка группы кнопок (индикаторов) управления сетью, громкостью и питанием компьютера. По сути, эти индикаторы являются одной общей кнопкой (группой кнопок).

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

## Панель задач

*Панель задач* — это еще одна особенность графической оболочки GNOME. Она представляет собой вертикальную панель, которая появляется сразу же после входа в ОС Ubuntu и расположена слева от рабочего стола (рис. 7.6).

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

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

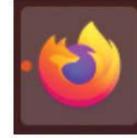
Как видите, после запуска приложения его значок поместился в полупрозрачную рамку, а слева от значка появилась небольшая точка. Одна точка означает, что запущена одна копия (окно) приложения, два — две копии приложения и т. д. Полупрозрачная рамка сообщает, что данное приложение в настоящий момент активно.



Рис. 7.6. Панель запуска в графической среде GNOME



а



б

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

### ПРИМЕЧАНИЕ

После нажатия на значок программы на панели основного меню сразу же появляется меню запущенной программы. Если переключиться или запустить другую программу, то на панели основного меню сразу же отобразится текущее меню активной программы. При этом меню неактивной программы не будет отображаться.

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

Однако это не весь набор спецэффектов значков приложений. Так, на некоторых значках программ могут появляться индикаторы хода выполнения определенной задачи, например при копировании файлов в файловом менеджере Nautilus.

На некоторых значках могут отображаться числа, но это в основном касается программ по обмену сообщениями. Таким способом программа сигнализирует вам о том, что имеются непрочитанные сообщения и т. п. Все зависит от функциональности той конкретной программы.

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

Открепляется значок по пункту меню **Удалить из избранного**.

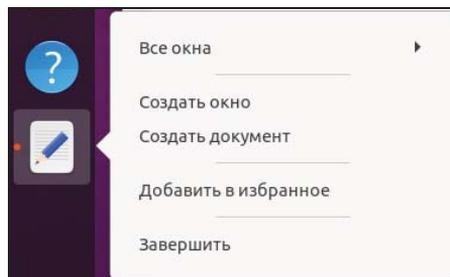


Рис. 7.8. Закрепление значка на панели задач

**ПРИМЕЧАНИЕ**

Количество пунктов меню также 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*, или рабочий стол, в ОС Ubuntu — это все оставшееся пространство на экране между панелью основного меню и панелью задач. Другими словами, бóльшая часть экрана и есть рабочее место.

При запуске ОС Ubuntu создается только одно рабочее место, однако добавить дополнительные рабочие места очень просто, чем в скором времени мы и займемся.

**ПРИМЕЧАНИЕ**

Более детальную настройку можно произвести путем установки дополнительных расширений для вашей графической оболочки. Такие программы называются *твиками* (от англ. tweaks). Скачать такие программы можно из стандартного репозитория Ubuntu. Чуть позже мы установим данную программу.

Добавленные рабочие места содержат одни и те же панель задач и панель основного меню. В общем виде рабочие места можно рассматривать как несколько мониторов, между которыми можно переключаться и которые содержат различные окна и запущенные приложения.

Рабочие места в ОС Ubuntu предназначены для упорядочивания окон запущенных программ. Бывает, что организовать работу легче всего, упорядочив окна по их типу, например, программы по работе с текстовыми документами, затем группа программ с графическими документами и отдельно запущенный файловый менеджер. Таким образом, лучше всего создать три рабочих стола и уже потом между ними переключаться.

Допустим, у вас открыт файловый менеджер Nautilus и вам необходимо создать еще одно рабочее место. Для добавления рабочего места нажмите кнопку **Обзор** и переместите курсор максимально в правую часть рабочего стола (рис. 7.9).

Вы увидите развернувшуюся часть рабочих мест. Для добавления новых рабочих мест дважды щелкните левой кнопкой мыши по самому нижнему рабочему месту (оно у нас без запущенных программ), и откроется пустой рабочий стол. Теперь запустите нужную вам программу, например браузер Mozilla Firefox, и он уже будет работать в пределах нового рабочего места.

Убедиться в том, что рабочее место было создано, можно описанным выше способом: нажав кнопку **Обзор** и переместив курсор вправо (рис. 7.10).

Для того чтобы переключиться на другое рабочее место, достаточно щелкнуть мышью по соответствующему рабочему месту. То же самое действие можно выполнить, нажав клавишу <Win> () и выбрав нужное рабочее место. Кроме

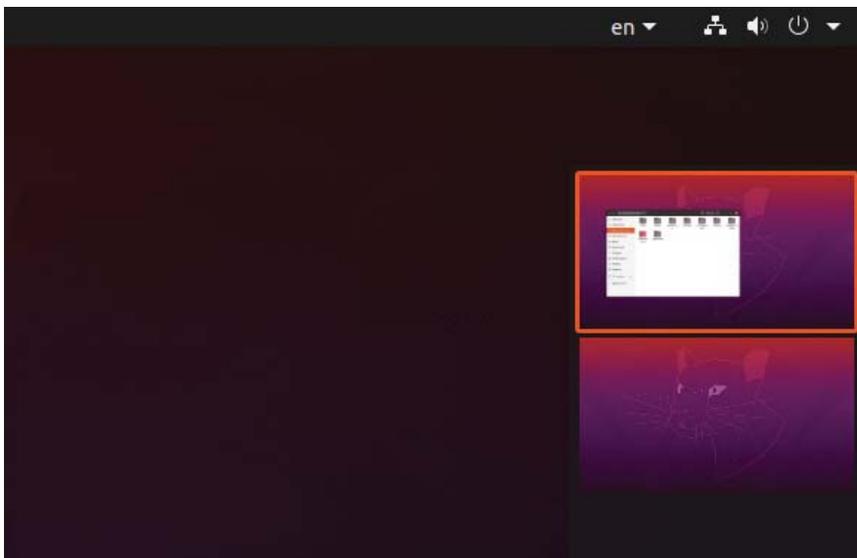


Рис. 7.9. Отображение списка рабочих мест

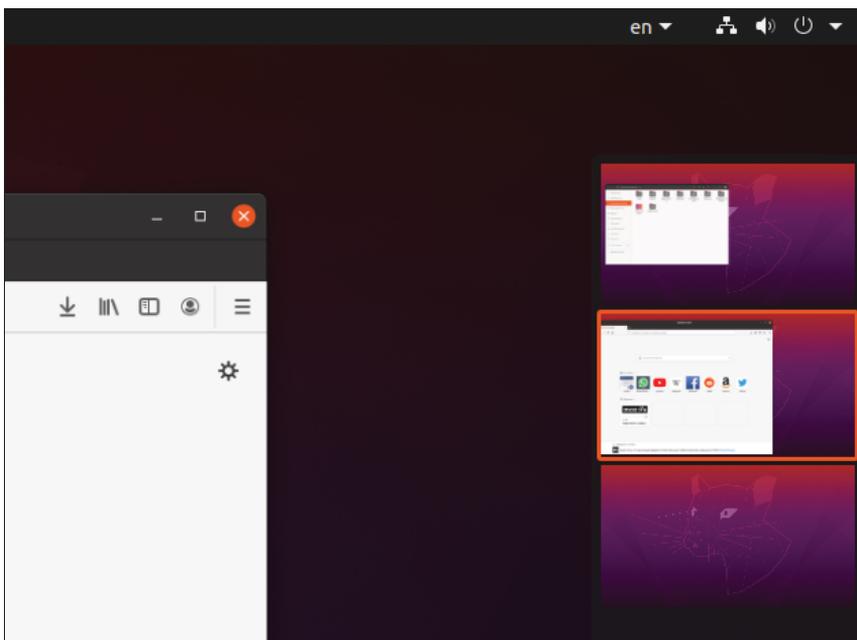


Рис. 7.10. Создано два рабочих места

того, переключаться между рабочими местами можно с помощью клавиш <Win>+<Page Up> и <Win>+<Page Down>.

Для того чтобы перенести запущенную программу из одной рабочей области в другую, сделайте активной данную программу и откройте список рабочих областей. Затем простым перетаскиванием поместите программу в требуемую рабочую область.

Как видите, функция создания и упорядочивания 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 программы из одной рабочей области в другую рабочую область окно программы выделяется полупрозрачной рамкой для лучшего восприятия при упорядочивании рабочих мест.

## Меню приложений

*Меню приложений*, или главное меню, скрывается за кнопкой запуска меню поиска и содержит список установленных приложений. Данное меню является одним из основных средств поиска и запуска приложений в графической оболочке GNOME.

Меню приложений открывается по нажатию клавиши <Win> либо по щелчку на значке **Показать приложения**, который расположен в левом нижнем углу рабочего стола (рис. 7.11).

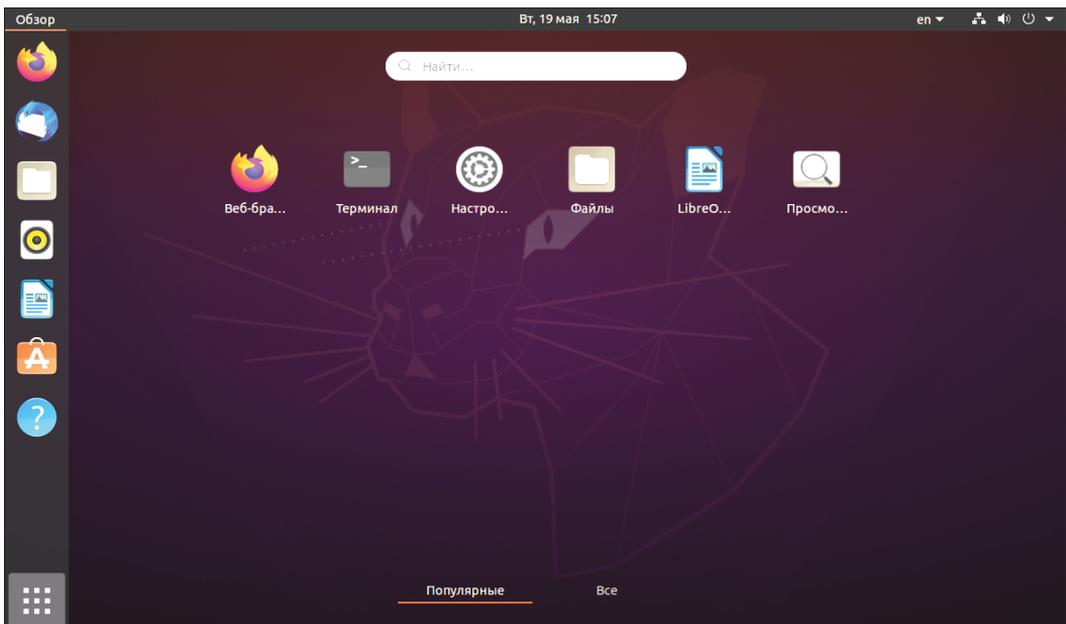


Рис. 7.11. Меню приложений

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 удобней было работать с операционной системой.

Давайте более детально рассмотрим главное меню.

По нажатию кнопки меню приложений открывается полупрозрачное окно со строкой поиска и отображением списка наиболее часто используемых программ.

Вверху окна располагается строка поиска, а внизу — вкладка с наиболее часто запускаемыми программами (вкладка **Популярные**), а также вкладка со списком всех установленных в системе программ (вкладка **Все**).

По мере ввода слова в строке поиска вам моментально будут предлагаться подходящие результаты поиска. К примеру, если в строке поиска начать вводить слово *настройки*, в результатах поиска будут предлагаться наиболее релевантные результаты. При этом по мере набора слова эти результаты будут сокращаться до наиболее подходящих по запросу.

Если ввести слово не полностью, например *наст*, то в результатах поиска отобразятся не только программы с подходящим названием, но и разделы пунктов меню в этих программах (рис. 7.12).

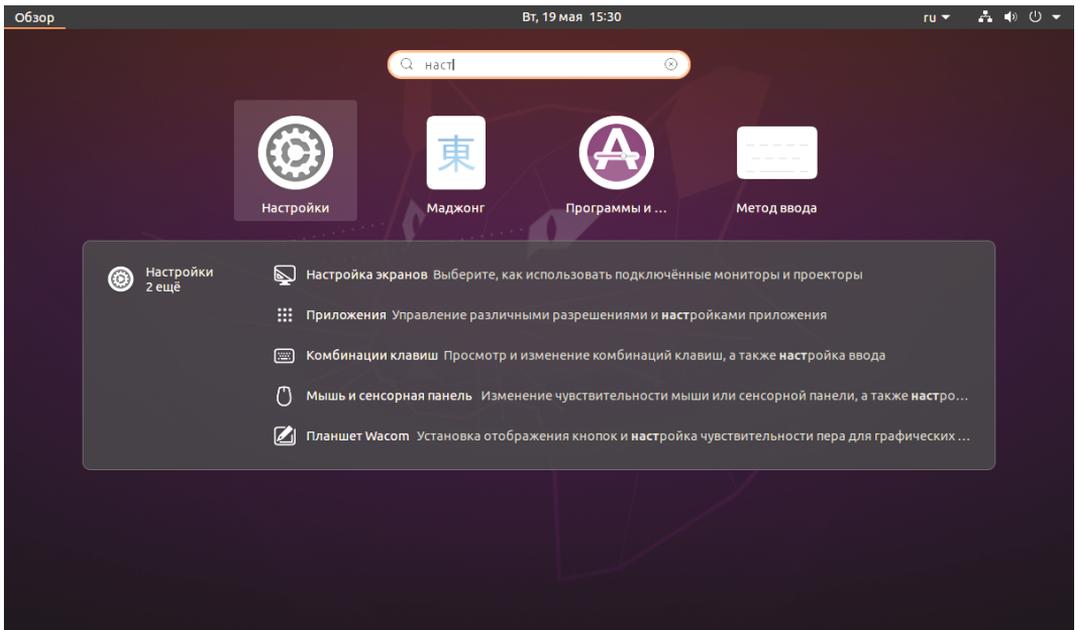


Рис. 7.12. Меню приложений предлагает подходящие команды

В левой части результатов поиска для удобства расположен быстрый доступ к строке поиска искомой программы. Так, если щелкнуть на пункте **Настройки 2 еще**, то откроется меню настроек системы со строкой поиска внутри программы.

## Быстрый запуск приложений

Быстрый запуск приложений — это дополнительная возможность запуска необходимой программы по ее названию.

Представим такую ситуацию: вы набираете текст в программе LibreOffice Writer, и вам необходимо сделать скриншот окна программы или ее части. При этом искать эту программу в пункте меню приложений вам совсем не хочется, т. к. это занимает больше времени, чем просто выполнить команду с клавиатуры. Как быть в такой ситуации?

Разработчики ОС Ubuntu предусмотрели такой вариант и реализовали возможность быстрого запуска программ.

К примеру, для того чтобы, не переключаясь в меню поиска приложений, сделать снимок части экрана, вам необходимо нажать сочетание клавиш `<Alt>+<F2>` и в строке ввода команды ввести команду `gnome-screenshot -a` и нажать клавишу `<Enter>` (рис. 7.13).

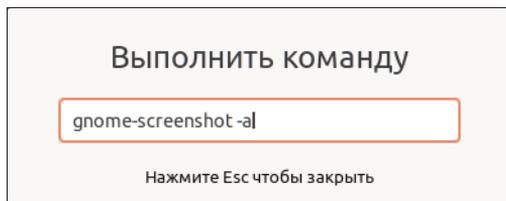


Рис. 7.13. Окно быстрого запуска программ

После нажатия клавиши `<Enter>` запустится программа для создания снимков экрана, которая закроется автоматически после выполнения.

## Выводы

Мы познакомились с графической средой рабочего стола под названием GNOME. Мы узнали, что для ОС Ubuntu существует несколько графических оболочек, а среда рабочего стола GNOME является стандартной для большинства дистрибутивов на базе Linux.

Мы научились работать с панелью задач, а также с панелью основного меню, выяснили, как создавать рабочие места и работать с ними. Эти знания помогут нам в дальнейшем наиболее продуктивно организовать свою работу. Кроме этого, мы научились быстро запускать программы, не прибегая к дополнительным манипуляциям, которые отнимают время.



## 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.

Окно настройки параметров системы можно вызвать следующими способами:

- ◆ нажать кнопку запуска приложений на панели задач, в строке поиска ввести *настройки* и после этого открыть соответствующее окно настройки системы;
- ◆ правой кнопкой мыши щелкнуть по рабочему столу и выбрать пункт меню **Параметры** для открытия окна настроек системы;
- ◆ нажать на группу значков основного меню и выбрать пункт **Настройки** для открытия соответствующего окна;
- ◆ ввести команду `gnome-control-center` в окне программы Терминал.

Мы воспользуемся четвертым вариантом, т. к. он является наиболее "линуковским". Итак, в Терминале выполните команду:

```
gnome-control-center
```

### **ПРИМЕЧАНИЕ**

Терминал запускается по нажатию клавиш <Ctrl>+<Alt>+<T>.

Откроется окно **Настройки** (рис. 8.1), которое напоминает Панель управления в ОС Windows. Действительно, идеология одна и та же.

В этом окне вы найдете пять основных групп настроек:

- ◆ настройки сетевого подключения — вкладки **Wi-Fi**, **Сеть** и **Bluetooth**, которые позволяют управлять сетевыми подключениями;
- ◆ персонализация — вкладки **Фон**, **Внешний вид**, **Уведомления**, **Поиск**, позволяющие персонализировать ваше рабочее место и настроить показ уведомлений, установленных в системе уведомлений;

- ◆ настройки системы — вкладки **Приложения**, **Конфиденциальность**, **Сетевые учетные записи** и **Общий доступ**, позволяющие настроить общий доступ к вашему компьютеру, задать ассоциацию файлов для того или иного приложения, а также настроить синхронизацию с сетевыми учетными записями;
- ◆ настройки оборудования — вкладки **Звук**, **Электропитание**, **Настройка экранов**, **Мышь и сенсорная панель**, **Комбинации клавиш**, **Принтеры**, **Сменные носители** и **Цвет**, позволяющие управлять настройками установленного в системе оборудования, например задавать громкость для системных звуков, управлять разрешением экрана вашего монитора, а также установленным в системе принтером;
- ◆ настройки рабочего окружения — вкладки **Регион и язык**, **Специальные возможности**, **Пользователи**, **Приложения по умолчанию**, **Дата и время** и **О системе**, позволяющие управлять установленными в системе языками, добавлять новые учетные записи, а также производить настройку даты и времени.

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.

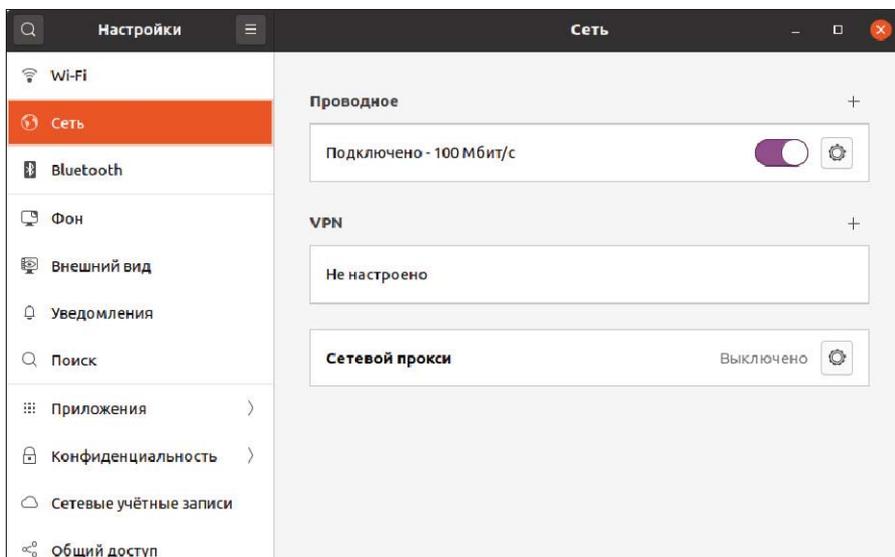


Рис. 8.1. Окно Настройки

## Настройка раскладки клавиатуры

По умолчанию в ОС Ubuntu смена раскладки клавиатуры выполняется нажатием сочетания 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  $\langle \text{Shift} \rangle + \langle \text{Alt} \rangle$  для смены раскладки. Поскольку 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  $\langle \text{Super} \rangle + \langle \text{Пробел} \rangle$  вместо  $\langle \text{Shift} \rangle + \langle \text{Alt} \rangle$ , но мы вправе поменять раскладку на свое усмотрение.

Давайте займемся сменой "горячих" клавиш раскладки клавиатуры. Откройте окно **Настройки**, если вы закрыли его, и перейдите на вкладку **Комбинации клавиш**. В группе **Ввод** будет отображен список используемых в текущий момент сочетаний клавиш для смены раскладки клавиатуры (рис. 8.2).

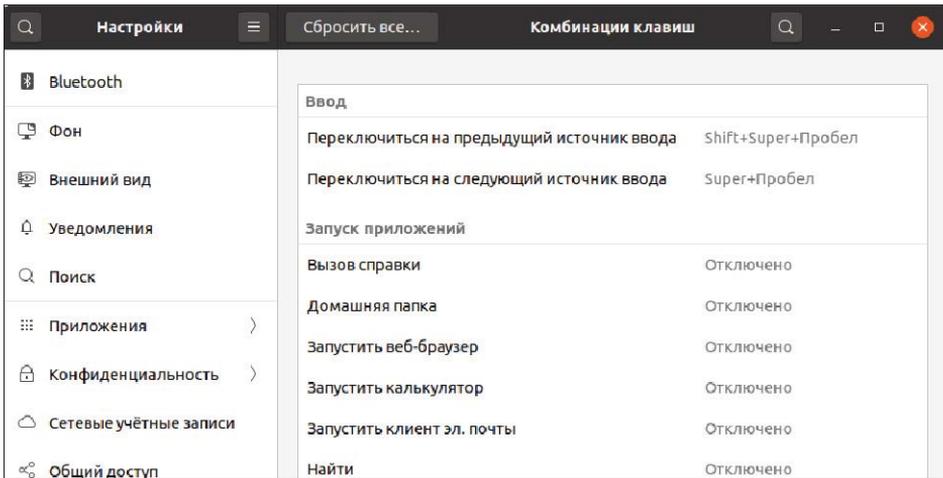


Рис. 8.2. Окно настроек "горячих" клавиш раскладки клавиатуры

Обратите внимание, что для удобства переключения раскладки клавиатуры можно назначить две комбинации клавиш, которые будут позволять переключать раскладку используемого языка не только на следующий, но и на предыдущий язык ввода. Это бывает полезно, когда в вашей системе присутствует более двух языков и вам приходится постоянно между ними переключаться.

Для смены "горячих" клавиш переключения раскладок клавиатуры щелкните по пункту **Переключиться на следующий источник ввода** и нажмите комбинацию клавиш  $\langle \text{Shift} \rangle + \langle \text{Alt} \rangle$  либо другое привычное вам сочетание.

И на этом этапе нас ожидает мелкая неприятность: привычное нам сочетание клавиш  $\langle \text{Shift} \rangle + \langle \text{Alt} \rangle$  более невозможно назначить, как это было в предшествующих версиях ОС Ubuntu. Но эта проблема легко решается с помощью программы GNOME Tweaks, которую для начала нужно установить из стандартного репозитория ОС Ubuntu. Давайте ее установим. Выполните в Терминале следующую команду:

```
sudo apt install gnome-tweaks
```

В ответ на это Терминал запросит ваш пароль и подтверждение установки. Нажмите клавишу <Y>, а затем <Enter>. Программа GNOME Tweaks будет установлена.

Теперь воспользуемся быстрым запуском установленного приложения. Для этого нажмите сочетание клавиш <Alt>+<F2>, ведите строку `gnome-tweaks` и нажмите клавишу <Enter>. Запустится окно дополнительных настроек GNOME (рис. 8.3).

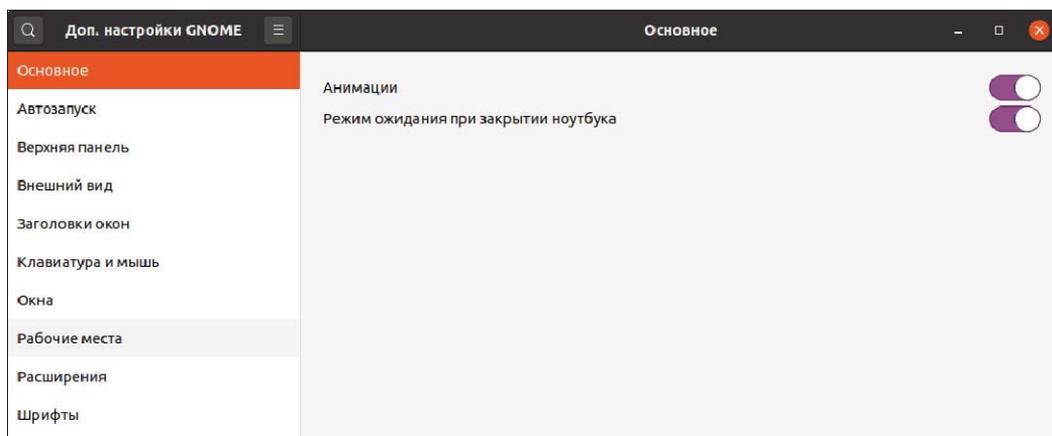


Рис. 8.3. Окно дополнительных настроек GNOME

Перейдите на вкладку **Клавиатура и мышь** и нажмите кнопку **Дополнительные параметры раскладки**. Откроется окно дополнительных параметров раскладки с разворачивающимся списком **Переключение на другую раскладку**, установите в нем флажок напротив опции <Alt>+<Shift> (рис. 8.4).

Окно дополнительных настроек можно закрыть.

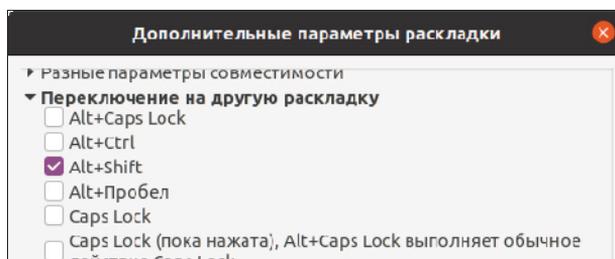


Рис. 8.4. Установка нужной раскладки

#### ПРИМЕЧАНИЕ

Обратите внимание, что в окне **Настройки** установленная раскладка не будет отображаться, т. к. она была настроена сторонней программой, а не системной.

Более детальную настройку клавиатуры можно выполнить в программе GNOME Tweaks, но разумнее всего будет использование системной программы **Параметры** для подобных настроек. Не зря же разработчики убрали эту возможность из системных настроек.

## Смена системного времени

Наверняка вы уже заметили, что при работе в ОС Ubuntu и ОС Windows системное время различное. Например, вы работаете ОС Ubuntu, а затем 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 (Coordinated Universal Time — всемирное координированное время). Это означает, что при настройке времени операционная система Ubuntu выставляла время UTC, а при выводе добавляла к нему смещение относительно текущего часового пояса, который вы выбрали при установке ОС.

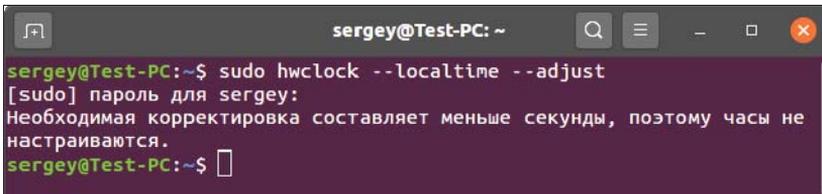
Допустим, ваш часовой пояс равен UTC+6 и текущее время равняется 19:42. ОС Ubuntu выставляет системное время относительно UTC и добавляет к нему смещение часового пояса. ОС Windows, в свою очередь, считывает системное время и выводит его как есть, т. е. без учета смещения относительно часового пояса,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

Такое поведение ОС Ubuntu удобно, но лишь если она является единственной ОС на компьютере. Если же она установлена по соседству с ОС Windows, то каждый раз настраивать время — дело не самое занятное. 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. 8.5).



```
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. 8.5. 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 при переключении из одной ОС в другую.

## 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 некоторых моделях ноутбуков, 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 включения 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 примерно следующая информация о видеокarte:

```
00:02:0 VGA compatible controller: Intel Corporation 2nd Generation Core Processor Family Integrated Graphics Controller (rev 09)
```

```
DeviceName: Intel® Graphics 3000
```

```
Subsystem: Hewlett-Packard Company 2nd Generation Core Processor Family Integrated Graphics Controller
```

```
''
```

```
01:00:0 VGA compatible controller: Advanced Micro Devices, Inc. [AMD/ATI] Thames [Radeon HD 7500M/7600M Series] (rev ff)
```

```
DeviceName: Radeon HD7670M
```

```
Kernel driver in use: radeon
```

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 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 dist-upgrade
```

По мере выполнения команды Терминал будет запрашивать 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 systemctl hibernate` для перевода компьютер в режим гибернации (спящий режим), `sudo systemctl 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 поможет выполнение следующей команды:

```
dconf reset -f /org/gnome/
```

Данная команда производит сброс настроек рабочего стола до настроек по умолчанию, таких как темы оформления окон, фонового рисунка, значков на панели задач. При этом личные данные, например документы на рабочем столе, удалены не будут.

В большинстве случаев выполнение данной команды позволяет решить проблему загрузки рабочего стола. Если вам не помогло это решение, то нужно искать причину до возникновения данной проблемы. Например, это может быть установка какой-либо программы, после которой появилась проблема загрузки рабочего стола.

## 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. 8.6).

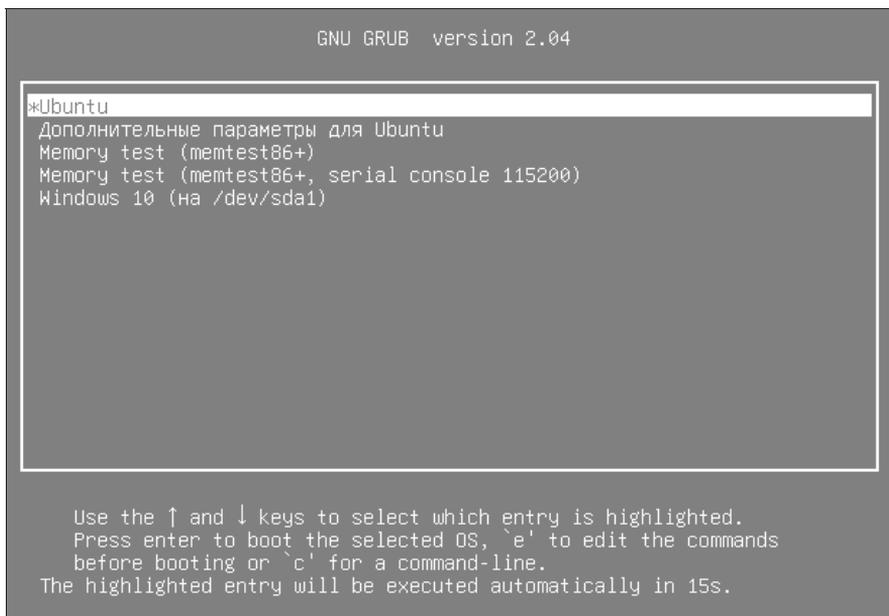


Рис. 8.6. Порядок загрузки ОС в загрузчике GRUB

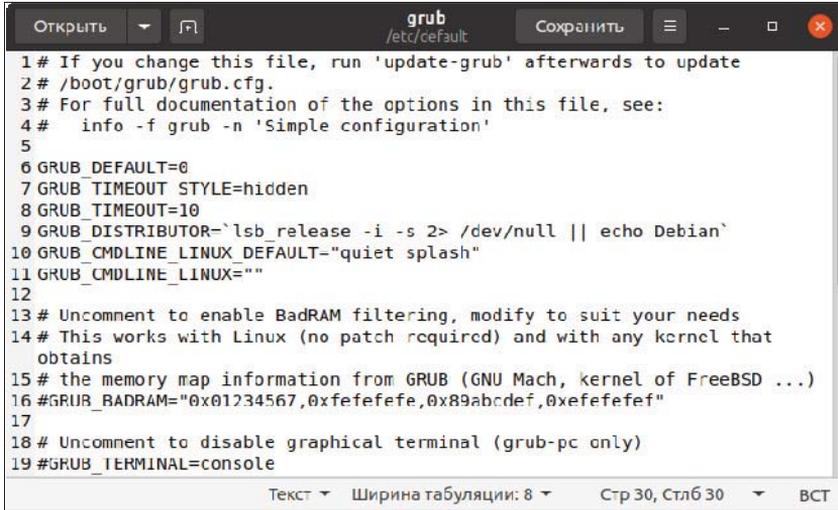
Да это уже знакомый нам загрузчик GRUB, который по умолчанию выделяет ОС Ubuntu первой в списке загрузки операционных систем. Но как быть, если мы хотим изменить этот порядок и сделать так, чтобы ОС Windows загружалась первой, а не ОС Ubuntu? Нужно отредактировать файл `/etc/default/grub` соответствующим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

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

Запустите Терминал и выполните следующую команду:

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

После ввода пароля откроется файл `/etc/default/grub` в текстовом редакторе Gedit (рис. 8.7).



```

Открыть  ▾  [🔍]  grub /etc/default  Сохранить  ☰  -  □  ×
1 # If you change this file, run 'update-grub' afterwards to update
2 # /boot/grub/grub.cfg.
3 # For full documentation of the options in this file, see:
4 #   info -f grub -n 'Simple configuration'
5
6 GRUB_DEFAULT=0
7 GRUB_TIMEOUT_STYLE=hidden
8 GRUB_TIMEOUT=10
9 GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
10 GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
11 GRUB_CMDLINE_LINUX=""
12
13 # Uncomment to enable BadRAM filtering, modify to suit your needs
14 # This works with Linux (no patch required) and with any kernel that
   obtains
15 # the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
16 #GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"
17
18 # Uncomment to disable graphical terminal (grub-pc only)
19 #GRUB_TERMINAL=console
Текст  ▾  Ширина табуляции: 8  ▾  Стр 30, Стлб 30  ▾  ВСТ

```

Рис. 8.7. Редактор Gedit открыл для редактирования файл `/etc/default/grub`

Нас интересует строка `GRUB_DEFAULT=0`, которая и задает приоритет загрузки ОС. Цифра `0` означает, что будет использована первая загрузочная 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. 8.6). 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 полуминуты, то присвойте параметру `GRUB_TIMEOUT` значение, равное `30`. Если указать значение равным `-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, используя программу GNOME Tweaks.

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

<b>Глава 9.</b>	Структура файловой системы
<b>Глава 10.</b>	Типы файлов в ОС Ubuntu
<b>Глава 11.</b>	Администратор и суперпользователь root
<b>Глава 12.</b>	Знакомство с Консолью и Терминалом
<b>Глава 13.</b>	Полезные команды Терминала
<b>Глава 14.</b>	Работа с привилегиями root
<b>Глава 15.</b>	Установка драйверов в ОС Ubuntu
<b>Глава 16.</b>	Установка принтера в ОС Ubuntu
<b>Глава 17.</b>	Установка и удаление программ в ОС Ubuntu
<b>Глава 18.</b>	Установка дополнительного программного обеспечения
<b>Глава 19.</b>	Запуск Windows-приложений в ОС Linux
<b>Глава 20.</b>	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 в консольном. Другое, менее распространенное название этого файлового менеджера — *Файлы* (от англ. *Files*).

Nautilus можно сравнить с Проводником 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 Midnight Commander, Thunar, PCManFM, Dolphin, Krusader, XFE и т. п. Все зависит от ваших личных предпочтений и привычек.

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

Запустить Nautilus можно следующими способами:

- ◆ щелкнуть мышью на панели задач по соответствующему значку;
- ◆ набрать в окне быстрого запуска `nautilus` и запустить данную 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,

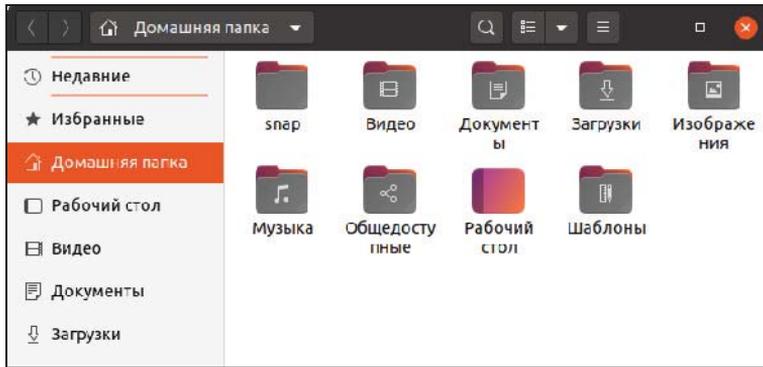


Рис. 9.2. Файловый менеджер Nautilus

в панели меню файлового менеджера (кнопка с изображением трех горизонтальных тире) установите флажок напротив пункта **Показывать скрытые файлы**, и вы их увидите (рис. 9.3). Имейте в виду, что скрытые файлы будут видны до тех пор, пока вы не сбросите только что установленный флажок. Если вы беспокоитесь, что можете случайно удалить важный скрытый файл, то лучше всего снять флажок и устанавливать его лишь в случае необходимости.

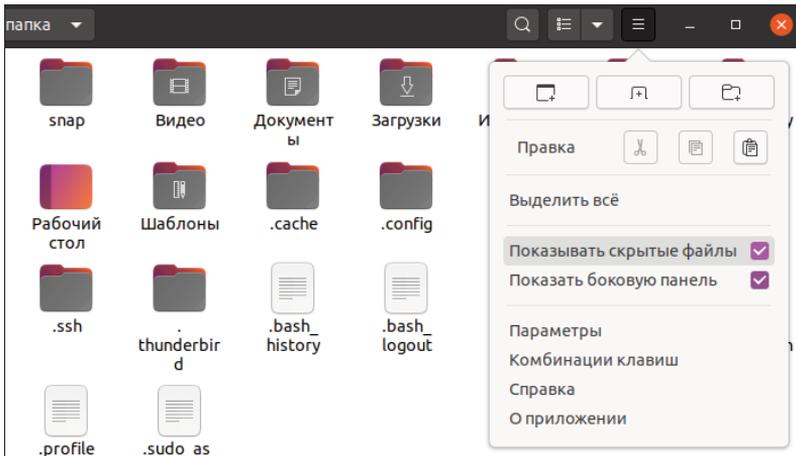


Рис. 9.3. Включение отображения скрытых файлов

После установки флажка для показа скрытых файлов они будут отображаться во всех каталогах ОС Ubuntu.

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

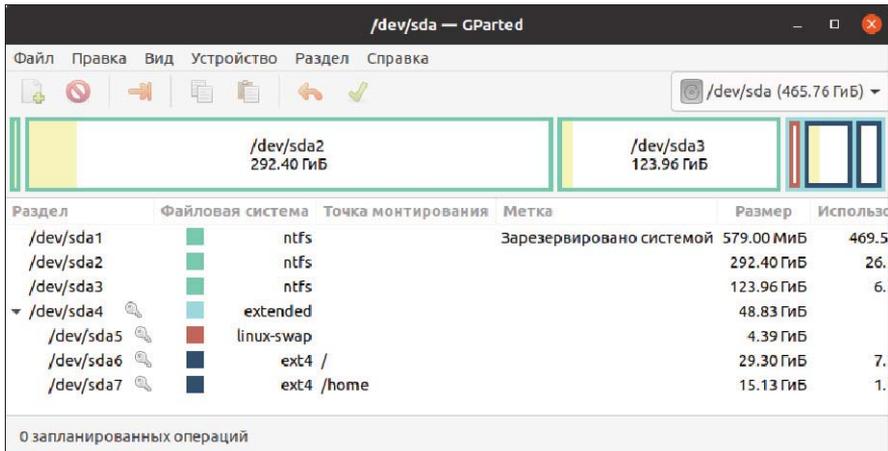


Рис. 9.4. Название устройств в программе GParted

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 нажмите на пункте **Другие места** и 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.

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

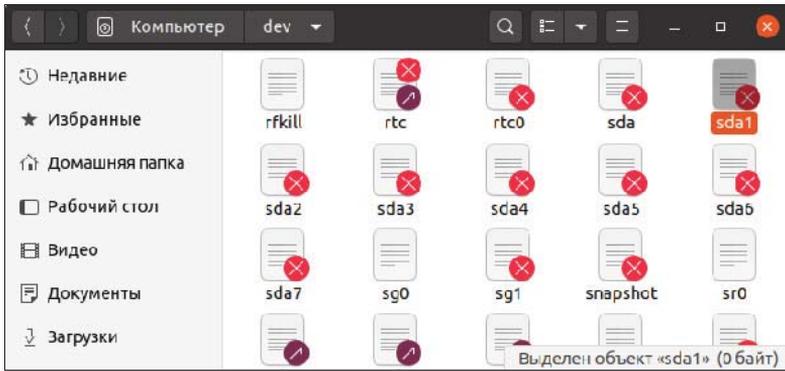


Рис. 9.5. Файлы устройств в каталоге /dev/

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 `sdc`. И так далее по аналогии.

С названием накопителей разобрались, теперь выясним, как выполняется нумерация 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, таблица разделов в любом случае будет такой, как на рис. 9.6.

<b>sda</b>	— основной жесткий диск
<b>sda1</b>	— основной раздел
<b>sda2</b>	— расширенный раздел
<b>sda5</b>	— логический раздел
<b>sda6</b>	— 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.

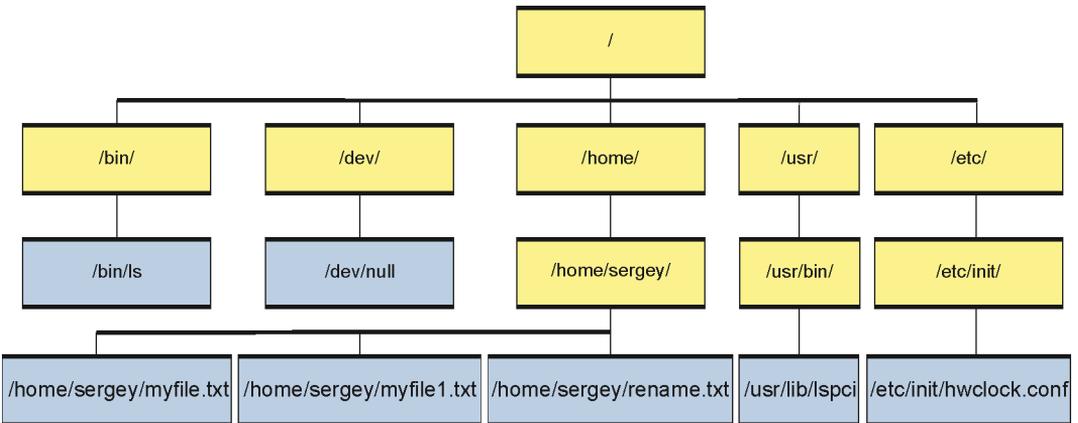


Рис. 9.7. Древовидная организация файловой системы

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

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).

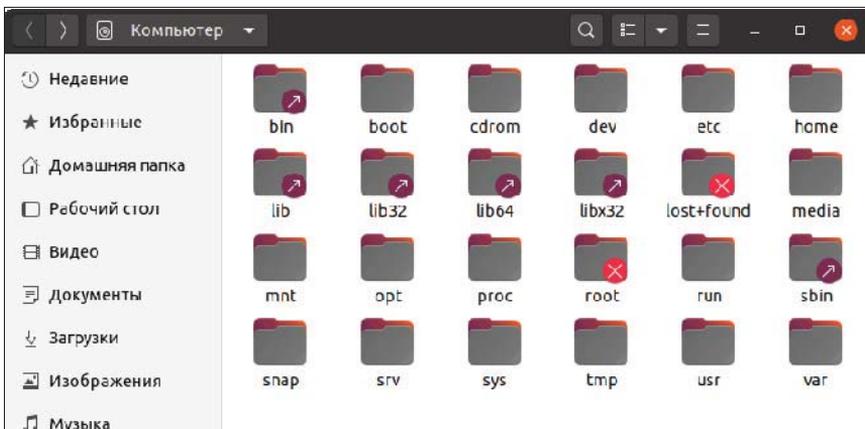


Рис. 9.8. Корень файловой системы

### ПРИМЕЧАНИЕ

Корень файловой системы можно развернуть следующим способом: запустите файловый менеджер Nautilus и в левой части окна нажмите на пункте **Другие места**, а затем 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 варьируется от установленных пакетов. Описывать каждый в отдельности не имеет смысла.

**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/	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/	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
/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/	Пользовательские точки монтирования
/opt/	Дополнительное программное обеспечение
/proc/	Виртуальная файловая система 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

Таблица 9.1 (окончание)

Каталог	Описание
/srv/	Каталог с данными, специфичными для окружения операционной системы
/sys/	Монтированная виртуальная файловая система 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/share/	Каталог с общими данными установленных программ
/usr/src/	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
/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).

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

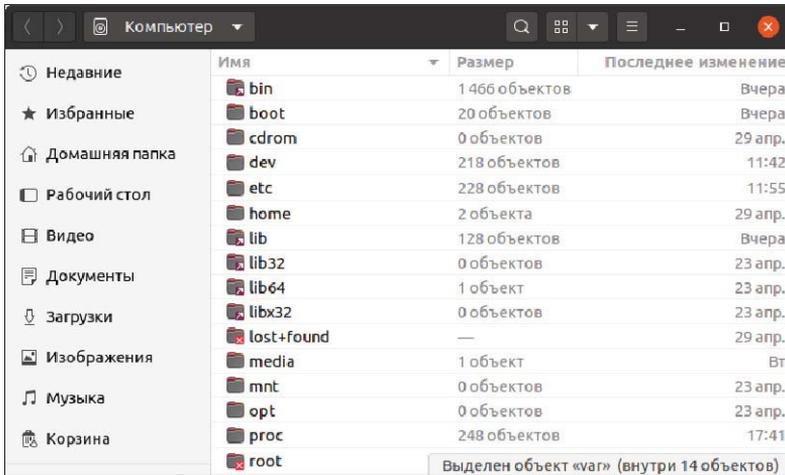


Рис. 9.9. Открыта корневая файловая система

## 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 базовые пользовательские программы, т. е. основные команды, которые необходимы пользователю в процессе работы с операционной системой (рис. 9.10).

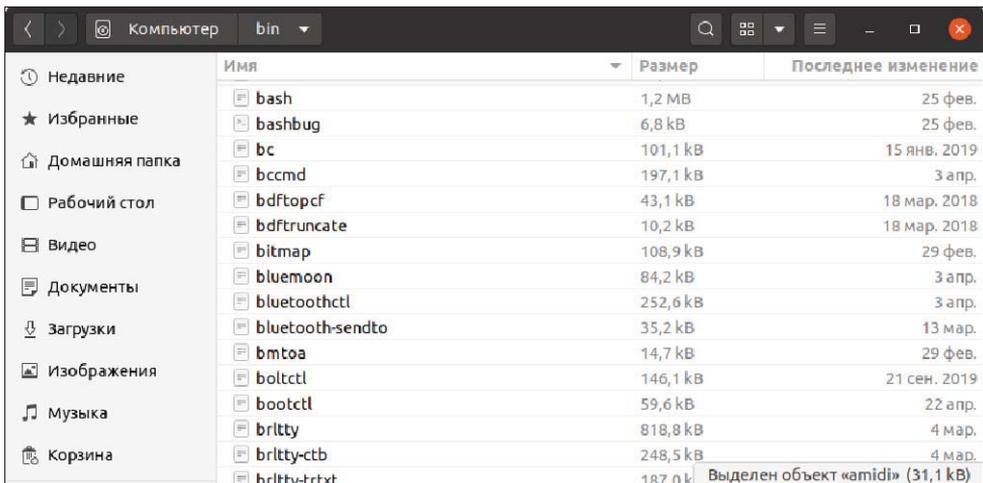


Рис. 9.10. Основные пользовательские программы

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. Он может быть изменен 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 cetera* — 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 сре-

ды 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 будут как минимум два каталога: `lost+found` (для хранения файлов, на которые нет ссылок) и каталог с именем вашей учетной записи (рис. 9.11).

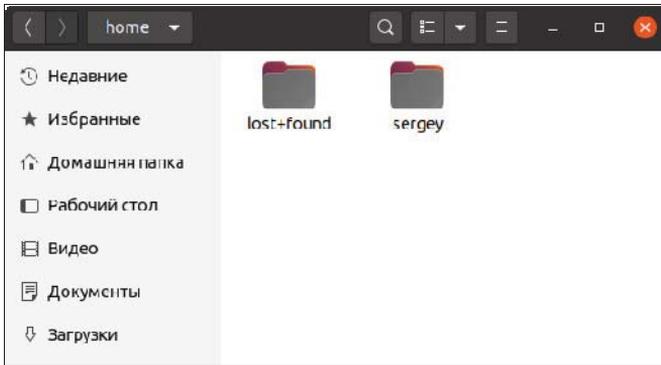


Рис. 9.11. Домашний каталог пользователя

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.

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 поскольку 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 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 этот факт обусловлен 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. Значит, 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, предоставляемых 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/ — один 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, согласно которому именуются жесткие диски 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 на основе анализа 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 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 именно находится внутри файла. Давайте на примере рассмотрим этот момент.

В домашнем каталоге пользователя создайте папку с именем Образец, щелкнув для этого правой кнопкой мыши и выбрав команду **Создать папку**. Затем, открыв Терминал, выполните следующую команду:

```
touch ~/Образец/ Sample.txt
```

По выполнении данной команды в директории ~/Образец/ будет создан простой текстовый файл 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 файлов списком, щелкнув несколько раз мышью по кнопке  в правом верхнем углу окна файлового менеджера Nautilus для выбора необходимого режима отображения (рис. 10.1).

Как вы могли заметить, на приведенном выше рисунке присутствует колонка **Тип**, а на вашем компьютере она отсутствует. Для того чтобы отобразить данную колон-

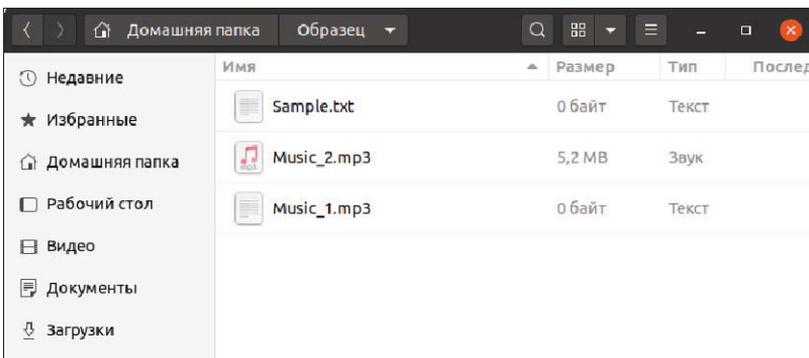


Рис. 10.1. Тип документа в файловом менеджере

ку, щелкните по заголовку колонки **Имя** и в контекстном меню установите флажок напротив пункта **Тип** (рис. 10.2).

После того как вы включили отображение колонки **Тип**, обратите внимание на тип файла в ней. Созданные нами документы `Sample.txt` и `Music_1.mp3` определяются как текстовые, а вот документ `Music_2.mp3` значится как **Звук**.

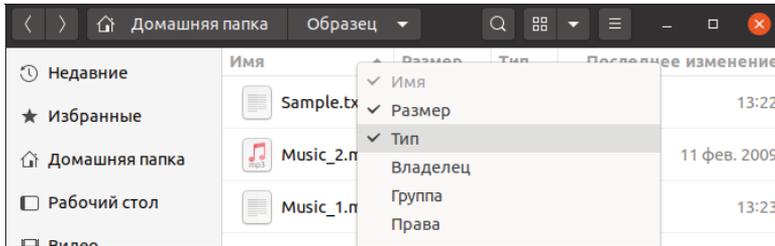


Рис. 10.2. Включение отображения колонки **Тип**

Давайте теперь переименуем `Music_2.mp3` в `Text.txt` и посмотрим, что получится (рис. 10.3).

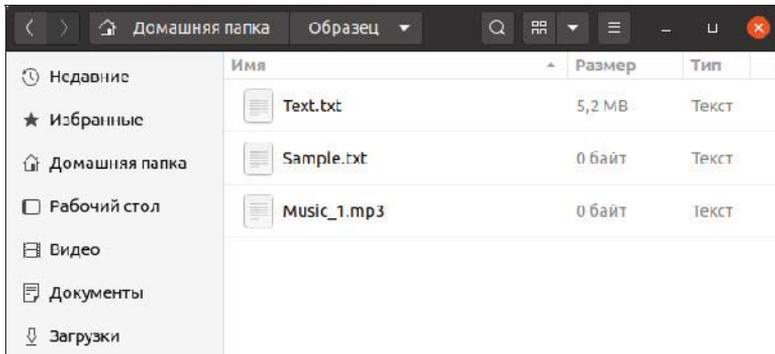


Рис. 10.3. Файловый менеджер определяет файл `Text.txt` как текстовый

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 содержимым, т. к. если открыть этот файл в текстовом редакторе, то вы увидите набор нечитаемых символов, которые по своей сути и являются текстовыми данными, хоть и не такими, каким вы их привыкли видеть. Как вы знаете, компьютеру совершенно не важно, какой тип файла перед нами: графический, текстовый или какой-то другой. Он видит лишь набор нулей и единиц, т. е. последовательность битов, из которой и получается тот или иной файл.

В других файловых менеджерах такой трюк со сменой расширения может не сработать, и файл будет определен именно по содержимому его заголовка, 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. 10.4).

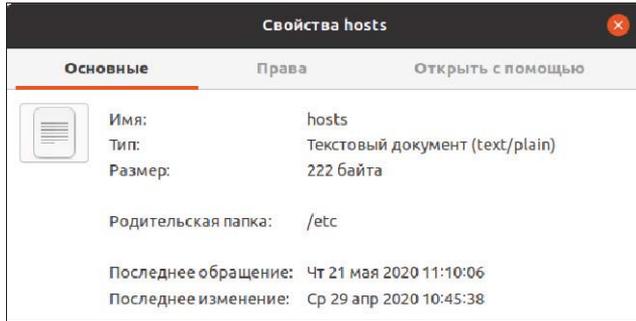


Рис. 10.4. Файл /etc/hosts определен как текстовый

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

Одна интересная особенность, которая кроется при работе с текстовыми файлами, созданными в ОС Windows, — это наличие так называемых кракозябр вместо текста (рис. 10.5).

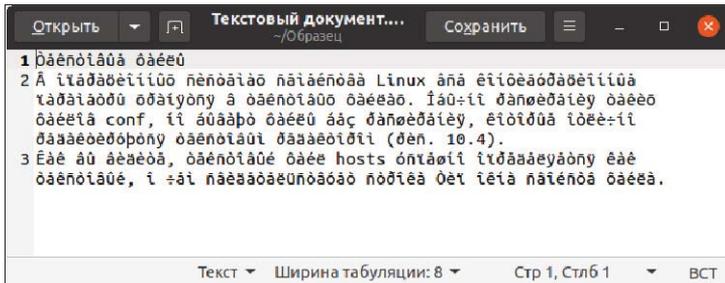


Рис. 10.5. Кракозябры вместо текста

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. 10.6).

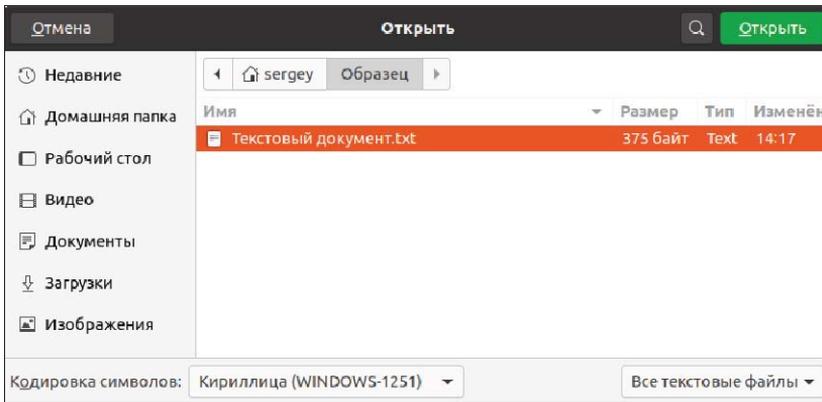


Рис. 10.6. Выбор кодировки для документов, созданных в ОС Windows

### 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. 10.7).

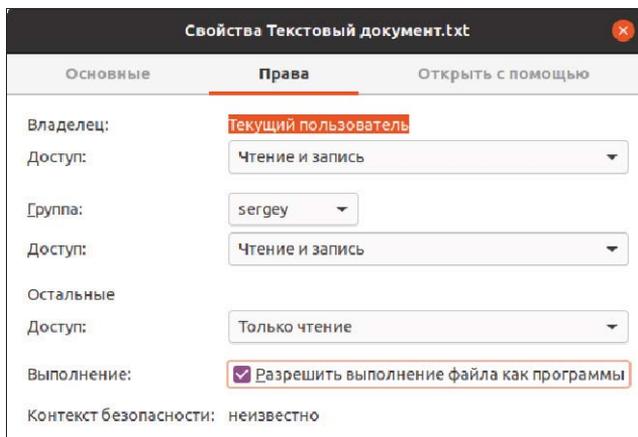


Рис. 10.7. Текстовый документ помечен как исполняемый

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 являются обычными текстовыми файлами, но содержат инструкции командного интерпретатора (рис. 10.8).

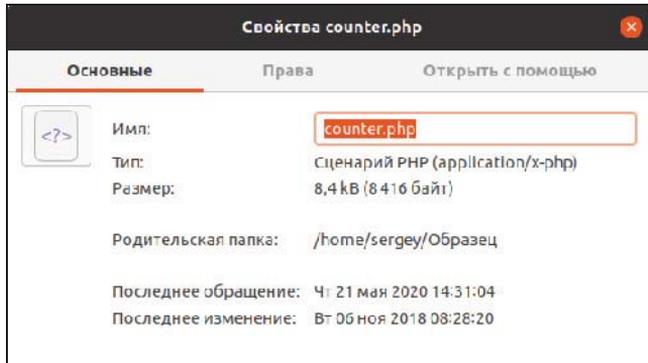


Рис. 10.8. Файл с инструкциями языка программирования PHP

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. Например, все тот же текстовый редактор Gedit вовсе не имеет расширения, но является исполняемым (рис. 10.9).

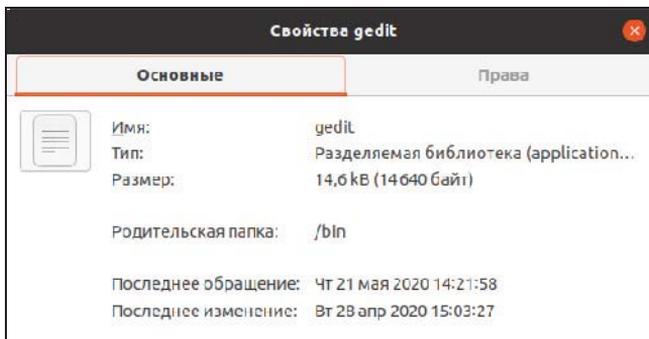


Рис. 10.9. Исполняемый файл текстового редактора Gedit

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, следует определиться с типом файла и поискать соответствующую программу для работы с файлом этого типа.

В любом случае незнакомый файл всегда можно открыть в текстовом редакторе Gedit и посмотреть его содержимое для установления его 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 свойств файла (рис. 10.10).

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. 10.11).

Если вы хотите сбросить сохраненную ассоциацию файлов, нажмите кнопку **Сбросить** в окне свойств файла, и сделанные 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 контекст-

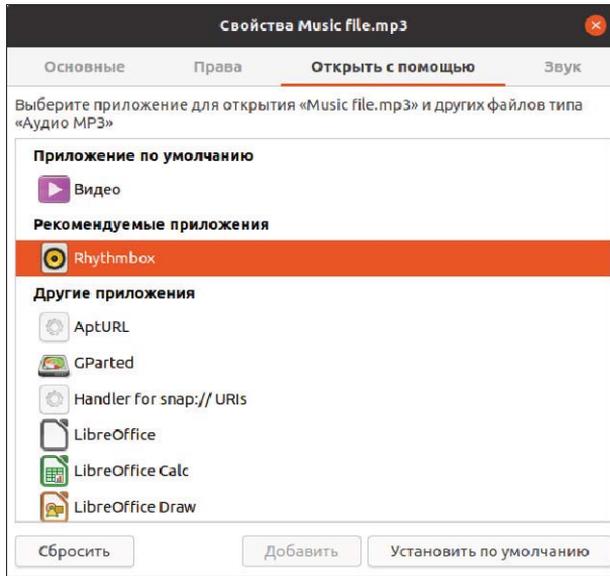


Рис. 10.10. Смена ассоциации аудиофайла

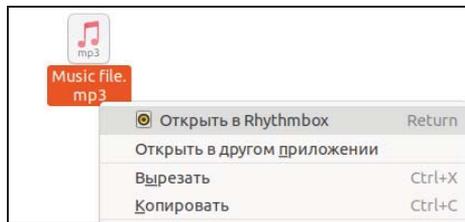


Рис. 10.11. Выбранная программа теперь ассоциируется с данным типом файла

ном меню выберите пункт **Открыть в другом приложении**, а затем выделите нужное приложение и нажмите кнопку **Выбрать**. Если для какого-либо файла нет рекомендуемых для открытия приложений, нажмите кнопку **Просмотреть все приложения** для просмотра списка всех установленных приложений в ОС. Если подходящего приложения нет в списке, придется искать подходящее 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 регистрозависимой, 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, а также научились задавать ассоциацию файлов с определенным приложением.

# 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 *суперпользователем 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 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?" Работать под учетной записью root — большой соблазн, который может окончиться 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 выполнять подобные дейст-

вия. 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

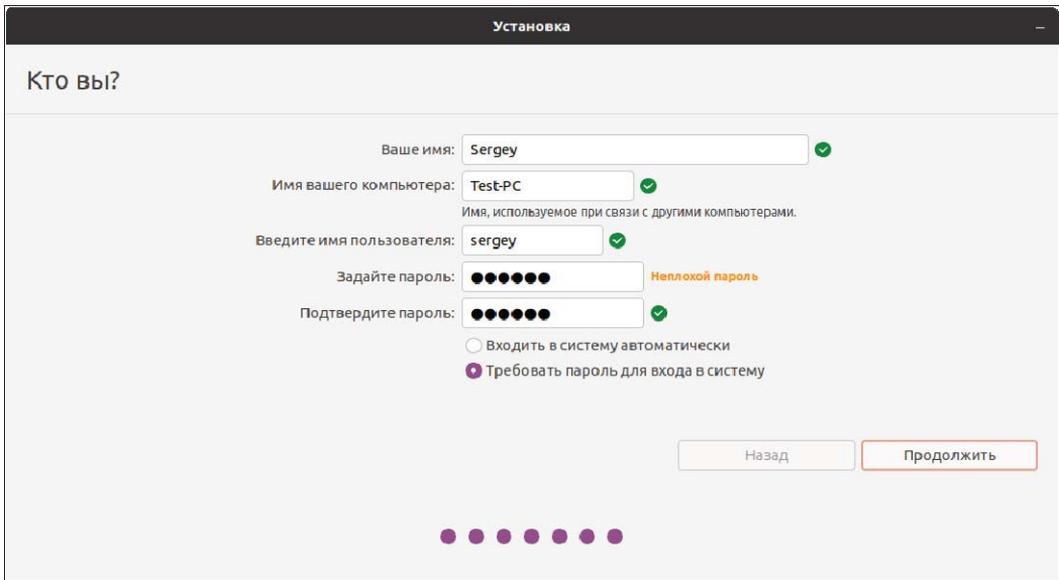
Как и ОС Windows, операционная система Ubuntu Linux позволяет создавать учетные записи двух типов:

- ◆ учетную запись администратора;
- ◆ обычную учетную запись.

Давайте сначала поговорим об *обычной учетной записи*. 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*. Она 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 имени какой учетной записи мы вводили пароль, когда ОС запрашивала нас при вводе команды `sudo` в Терминале. Правильно! Мы вводили пароль именно от имени этой учетной записи, которую создали еще на этапе установки операционной системы.



Установка

Кто вы?

Ваше имя:  ✓

Имя вашего компьютера:  ✓  
Имя, используемое при связи с другими компьютерами.

Введите имя пользователя:  ✓

Задайте пароль:  Непохожий пароль

Подтвердите пароль:  ✓

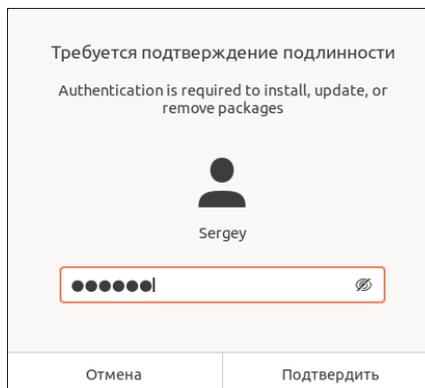
Входить в систему автоматически  
 Требовать пароль для входа в систему

••••••

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 потребуется 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).



Требуется подтверждение подлинности  
Authentication is required to install, update, or remove packages

  
Sergey

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.

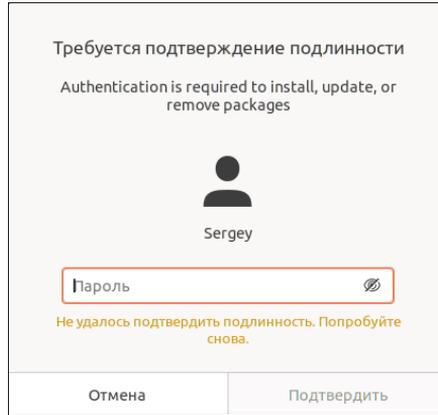


Рис. 11.3. Неправильный ввод пароля

Пугаться этого не стоит, потому что вы можете повторить попытку ввода пароля.

Теперь все стало 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 щелкните правой кнопкой мыши по рабочему столу и выберите пункт меню **Параметры**. Откроется окно параметров системы, в котором нужно найти вкладку **Пользователи**, и щелкните по ней. Откроется окно управления учетными записями (рис. 11.4).

В этом окне мы видим, что у нас создана одна учетная запись, которая уже наделена правами администратора, т. к. она была создана еще на этапе установки ОС.

Обратите внимание, что некоторые кнопки неактивны (заблокированы). Для того чтобы сделать их активными, нажмите кнопку **Разблокировать...** в верхней части окна 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

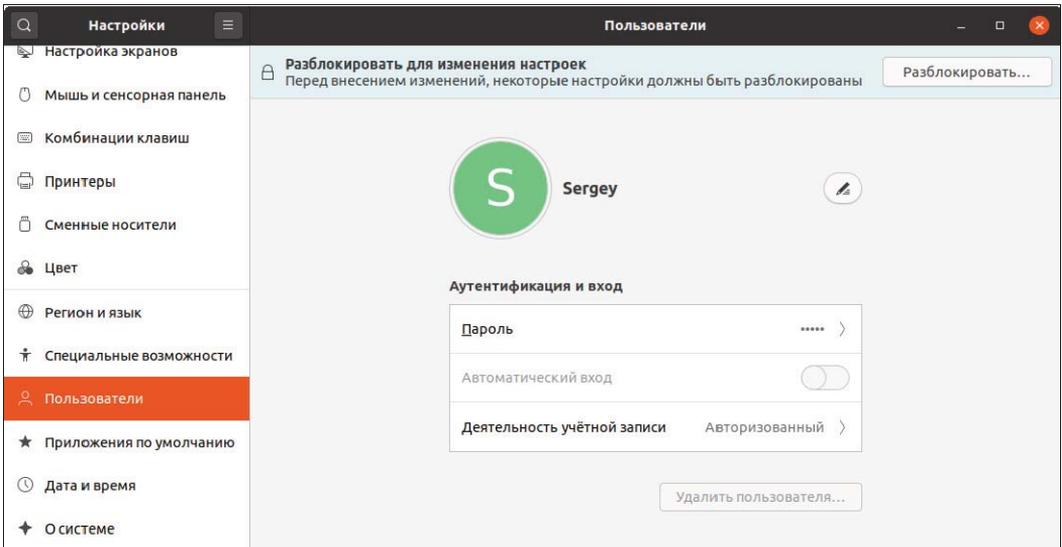


Рис. 11.4. Управление учетными записями

аутентификации пользователя, 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 верхней части окна нажмите кнопку с **Добавить пользователя...** и в открывшемся окне добавления учетной записи укажите нужный тип учетной записи: **Обычный** или **Администратор** (рис. 11.5).

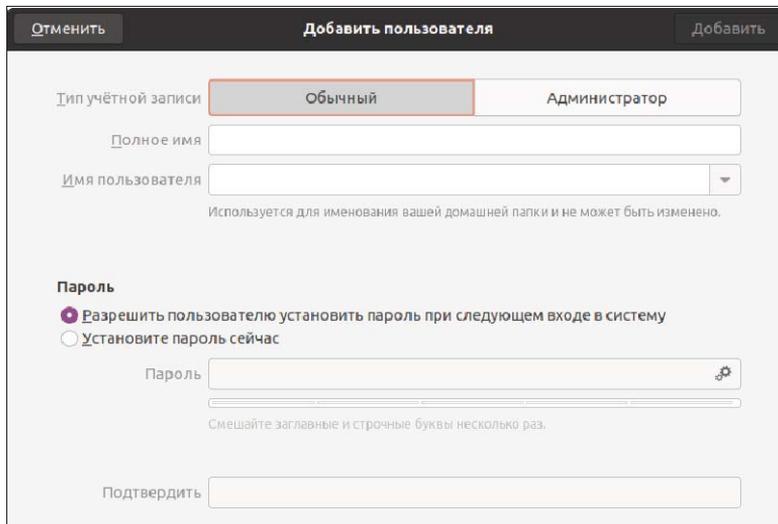


Рис. 11.5. Окно добавления учетной записи

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 пароль от учетной записи администратора, 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.

Пользователю, выполнившему команду `sudo`, 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 выяснили, что кроме административных учетных записей ОС Ubuntu Linux позволяет создавать и обычные, гостевые учетные записи, благодаря которым пользователь не сможет нарушить работоспособность операционной системы.

## ГЛАВА 12



# Знакомство с Консолью и Терминалом

## Что такое Терминал и Консоль

До этого момента нам нередко приходилось открывать Терминал и выполнять какие-либо команды.

В данной главе речь пойдет именно о работе с Терминалом и Консолью, и мы наконец-то поближе познакомимся с этими мощными инструментами, с которыми 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 — с Терминалом.

*Терминал* — это название программы в ОС Ubuntu Linux, которая позволяет обрабатывать направляемые этой программе команды и передавать 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, Консоль является устройством ввода-вывода.

Так, при запуске ОС Ubuntu Linux запускается сразу 5 виртуальных консолей, одна из которых стартует в графическом режиме. По умолчанию загружается сразу первая консоль, т. е. то, что мы привыкли называть графическим режимом. Остальные 4 виртуальные консоли 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`.

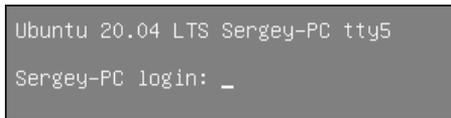
Пользователь вправе изменить стандартную командную оболочку на любую другую, но при этом стоит учитывать особенности новой оболочки, такие как различное написание типичных команд и т. п.

## Запуск Консоли

Консоль можно запустить прямо из графического режима, нажав одно из сочетаний клавиш:

- ◆ `<Ctrl>+<Alt>+<F1>` — для перезапуска первой виртуальной консоли и входа в графический режим;
- ◆ `<Ctrl>+<Alt>+<F2>` — для возврата в первую виртуальную консоль в графическом режиме;
- ◆ `<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.

По нажатию этих клавиш вызовется соответствующая консоль (рис. 12.1).



```
Ubuntu 20.04 LTS Sergey-PC tty5
Sergey-PC login: _
```

Рис. 12.1. Запущена первая виртуальная консоль `tty1`

В нашем случае была вызвана пятая виртуальная консоль `tty5` (сочетание клавиш `<Ctrl>+<Alt>+<F5>`). При запуске виртуальной консоли от вас требуется ввести свой логин и пароль. В нашем случае логин — 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 соответствующий 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>+<F2>`, но не забудьте перед этим выйти из Консоли, выполнив команду `exit`.

```
Ubuntu 20.04 LTS Sergey-PC tty5

Sergey-PC login: sergey
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

1 обновление может быть установлено прямо сейчас.
0 из этих обновлений, являются обновлениями безопасности.
Чтобы просмотреть дополнительные обновления выполните: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
*** Требуется перезагрузка системы ***

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

sergey@Sergey-PC:~$
```

Рис. 12.2. Приветствие Консоли

Если вам нужно перезагрузить рабочий стол, нажмите `<Ctrl>+<Alt>+<F1>`. Система завершит сеанс, перезагрузит рабочий стол и попросит заново пройти аутентификацию, как при обычном включении компьютера.

### 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 **Показать приложения** на панели задач, в строке поиска ввести слово *Терминал* и запустить требуемую программу;

- ◆ на рабочем столе нажать правую клавишу мыши и выбрать пункт меню **Открыть в терминале**.

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):

```
sergey@Test-PC:~$
```

Давайте разберем эту строку на составляющие (табл. 12.1).

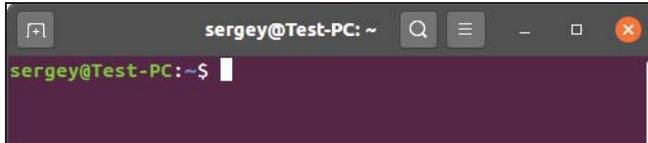


Рис. 12.3. Открыто окно Терминала

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, которыми вам придется пользоваться довольно часто (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 введенных команд вверх
<↓> либо <Ctrl>+<N>	Прокрутка недавно введенных команд вниз
<Ctrl>+<A>	Перемещение курсора в начало строки
<Ctrl>+<E>	Перемещение курсора в конец строки
<Ctrl>+<U>	Удаление текущей строки

Таблица 12.2 (окончание)

"Горячие" клавиши	Описание
<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>	Сигнал завершения режима ввода
<Ctrl>+<D>	Закрытие текущего Терминала

## 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:

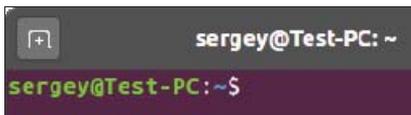
```
uname -r
```

Данная команда выведет номер выпуска операционной системы. Если запустить команду `uname` с ключом `-a`, будет выведена вся доступная информация об используемой операционной системе.

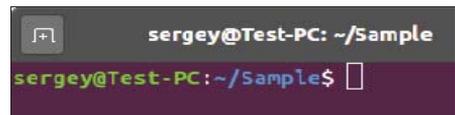
## Установка текущего каталога

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. Текущий каталог для разных сеансов Терминала:

а — домашний каталог — `~ (/home/sergey/)`;

б — домашний каталог — `~/Sample/ (/home/sergey/Sample/)`

Как видим, на рис. 12.4 показаны 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 изменения текущего каталога на другой служит команда `cd` (от англ. *change directory* — изменить каталог), например:

```
cd /home/sergey/myfolder/
```

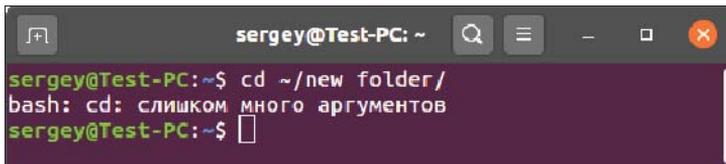
**ПРИМЕЧАНИЕ**

Для того чтобы не указывать 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*.



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ cd ~/new folder/  
bash: cd: слишком много аргументов  
sergey@Test-PC:~$
```

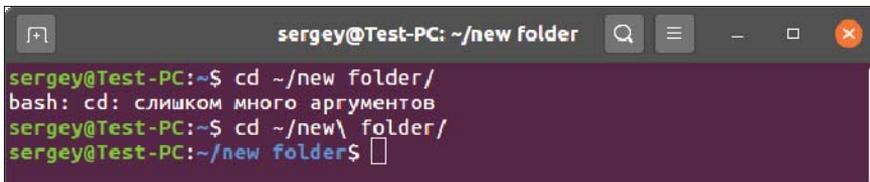
Рис. 12.5. Терминал "ругается" на символы пробела в пути к каталогу

*Экранирование* — это постановка символа обратного слеша `\` в пути к файлу или каталогу.

Например, чтобы Терминал правильно отреагировал на специальные символы, выполним экранирование пробелов в имени каталога:

```
cd ~/new\ folder/
```

Теперь Терминал правильно обработал специальные символы, т. е. символ пробела, и сменил на тот, который мы указали (рис. 12.6).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ cd ~/new folder/  
bash: cd: слишком много аргументов  
sergey@Test-PC:~$ cd ~/new\ folder/  
sergey@Test-PC:~/new folder$
```

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 -
```

Для того 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, будет выведен список содержимого текущего каталога (рис. 12.7).

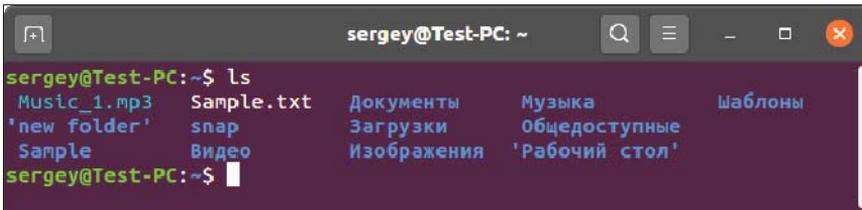


Рис. 12.7. Список содержимого текущего каталога

Однако если передать команде параметр `-l`, 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).

По умолчанию результат выводится в девять колонок. Сейчас разберем, что обозначают эти непонятные надписи на примере первой строки (табл. 12.3).

Таблица 12.3. Описание содержимого вывода команды `ls -la`

Номер столбца	Содержимое столбца	Описание
1	<code>drwxr-xr-x</code>	Права доступа на файл. В нашем случае права доступа на каталог, т. к. первый символ <code>d</code> обозначает, что перед нами каталог

Таблица 12.3 (окончание)

Номер столбца	Содержимое столбца	Описание
2	20	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	мая	Месяц последнего изменения файла
7	22	День последнего изменения файла
8	10:34	Время последнего изменения файла
9	.	Имя файла. В нашем случае точка означает ссылку на текущий каталог

```

sergey@Test-PC: ~
sergey@Test-PC:~$ ls -la
итого 104
drwxr-xr-x 20 sergey sergey 4096 мая 22 10:34 .
drwxr-xr-x  4 root   root   4096 апр 29 10:46 ..
-rw-r----- 1 sergey sergey 1402 мая 22 10:46 .bash_history
-rw-r--r--  1 sergey sergey  220 апр 29 10:46 .bash_logout
-rw-r--r--  1 sergey sergey 3771 апр 29 10:46 .bashrc
drwx----- 20 sergey sergey 4096 мая 20 10:12 .cache
drwxr-xr-x 19 sergey sergey 4096 мая 22 09:23 .config
drwx-----  3 sergey sergey 4096 мая 21 11:54 .gnupg
drwxr-xr-x  3 sergey sergey 4096 апр 29 11:11 .local
drwx-----  5 sergey sergey 4096 апр 29 11:26 .mozilla
-rw-rw-r--  1 sergey sergey    0 мая 21 13:23 Music_1.mp3
drwxrwxr-x  2 sergey sergey 4096 мая 22 10:34 'new folder'

```

Рис. 12.8. Подробный вывод содержимого каталога, включая скрытые и системные файлы

Для того чтобы просмотреть содержимое какого-то определенного каталога, выполните следующую команду:

```
ls -al /etc/
```

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 познакомились с Терминалом, стоит сказать несколько слов о справочной системе команд этой программы, так называемым мануалом. Им часто любят пугать неопытных пользователей в русскоязычном сегменте Интерне-

та. Сам мануал — это всего лишь справочная система по той или иной команде Терминала. Само слово — это калька (транслитерация) английского *manual*, что в переводе означает "руководство".

В Терминале для удобства набора команд это слово было сокращено до трех букв — `man`.

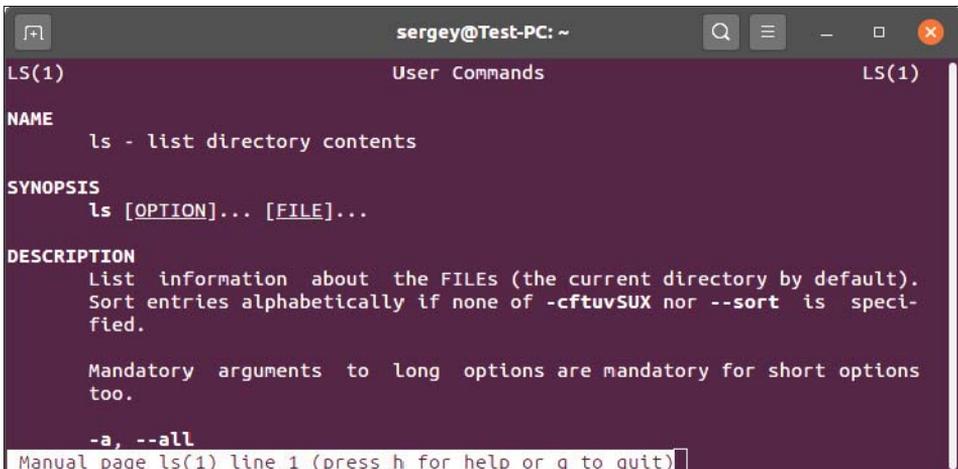
Синтаксис команды следующий:

```
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
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 этот недостаток 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.

Продемонстрируем возможности автодополнения на практике. Начните вводить в Терминале символы `apt in` и нажмите клавишу `<Tab>`, и Терминал автоматически дополнит набранное слово, точнее, команду.

Это одна ситуация. А что, если вы забыли 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).

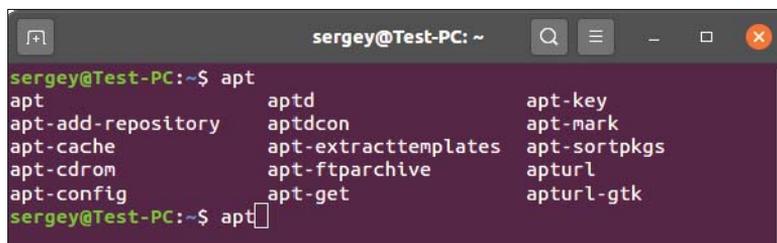


Рис. 12.10. Терминал вывел список всех команд, начинающихся с искомого слова

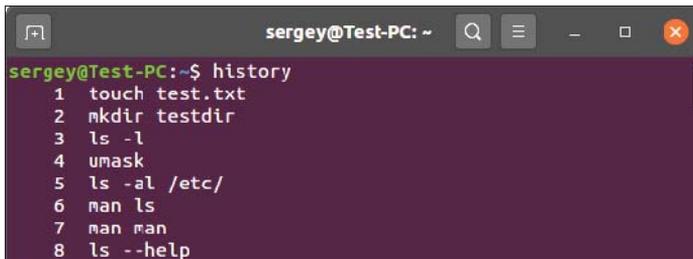
Как видите, Терминал вместо автодополнения вывел список всех команд, которые начинаются с искомого слова. Просмотрев этот список, вы без труда найдете именно ту команду, которая вам требуется. Очень полезная функциональность Терминала, не так ли?

## 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.

Для просмотра ранее введенных команд служит команда `history`, которая выведет их список (рис. 12.11).



```
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

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, 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 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 баналь-

ности. 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. Учтите, 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>`.

Тех же правил придерживайтесь и при работе в графическом режиме: отошли от компьютера — заблокируйте перед этим рабочий стол, нажав `<Super>+<L>`.

## Выводы

Мы познакомились с Консолью и Терминалом, навыки использования которых просто необходимы для полноценного и более профессионального 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 на самом деле неограниченны. Все упирается только в ваши практические навыки работы в этой программе.

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 выполняются 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 из командной строки, 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, которые 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]
```

Необязательный параметр *ключ* принимает значения, перечисленные в 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 они 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	--msg	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*]

Описание опций формата вывода приведено в табл. 13.2.

Таблица 13.2. Опции 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, будет использоваться текущее время.

Синтаксис команды следующий:

```
touch [ключи]... файл...
```

Необязательный параметр *ключи* принимает значения, перечисленные в табл. 13.3.

Таблица 13.3. Ключи команды `touch`

Сокращенный вариант	Полный вариант	Описание
-a		Изменяет только время доступа к файлу
-c	--no-create	Не создает файл, если он существует
-d	--date= <i>строка</i>	Анализирует переданную в качестве аргумента строку и использует 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	Изменяет символьные ссылки, а не сами файлы, на которые они ссылаются
-m		Устанавливает только время последнего изменения файла
-r	--reference= <i>файл</i>	Использует время файла, переданного в ключе <code>--reference=<i>файл</i></code>
-t <i>время</i>		<p>Устанавливает 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>[ [ВВ] ГГ ] ММ ДД чч мм [ . сс ]</pre> <p>где ВВ — 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; сс — значение секунд даты.</p> <p>Если параметр ВВ не задан и значение ГГ находится в пределах 69 и 99, тогда ВВ устанавливается равным 19, иначе будет использоваться значение 20</p>

Таблица 13.3 (окончание)

Сокращенный вариант	Полный вариант	Описание
	<code>--time=слово</code>	Задаёт время в соответствии с переданным форматом времени в аргументе <i>слово</i> . Формат времени принимает 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
```

В результате в домашнем каталоге вашей учетной записи будет создан соответствующий файл, если до этого он отсутствовал. В случае существования заданного файла команда только обновит время последнего изменения и доступа к нему, 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).

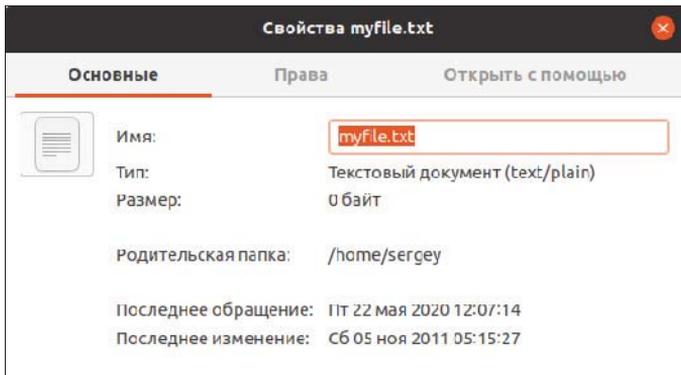


Рис. 13.1. Файл с модифицированной датой последнего изменения

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 [ключи]... каталог...
```

Необязательный параметр *ключи* принимает значения, перечисленные в табл. 13.4.

Таблица 13.4. Ключи команды `mkdir`

Сокращенный вариант	Полный вариант	Описание
<code>-m</code>	<code>--mode=режим</code>	Устанавливает права доступа к файлу, как в команде <code>chmod</code> . При этом не учитывается пользовательская маска, т. е. права устанавливаются без учета <code>umask</code>
<code>-p</code>	<code>--parents</code>	Не выводит ошибку, если каталог существует, а также создает недостающие родительские каталоги для каждого указанного каталога
<code>-v</code>	<code>--verbose</code>	Указывает на необходимость вывода сообщения о каждом созданном каталоге, а также выводит сообщение в случае, если создаваемый каталог уже существует
<code>-Z</code>		Устанавливает контекст безопасности SELinux для каждого создаваемого каталога равным типу по умолчанию
	<code>--context [=CTX]</code>	Аналогичен параметру <code>-Z</code> , но если указан <code>CTX</code> , то устанавливает контекст безопасности SELinux или SMACK равным <code>CTX</code>
	<code>--help</code>	Выводит справочную информацию
	<code>--version</code>	Выводит информацию о версии

Если не передавать команде `mkdir` никаких параметров, а указывать 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 подкаталога: `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 он уже существует.

## Удаление пустых каталогов

Создавать каталоги с помощью Терминала мы уже научились, но нужно еще и уметь их удалять. Для удаления *пустых каталогов* предназначена команда `rmdir` (от англ. *remove directory* — удалить каталог), синтаксис таков:

```
rmdir [ключи]... каталог...
```

Необязательный параметр *ключи* принимает значения, перечисленные в табл. 13.5.

Таблица 13.5. Ключи команды `rmdir`

Сокращенный вариант	Полный вариант	Описание
	<code>--ignore-fail-on-non-empty</code>	Игнорирует все ошибки, которые будут возникать при удалении непустых каталогов
<code>-p</code>	<code>--parents</code>	Удаляет каталог и его потомков, т. е. все вложенные каталоги в этом каталоге
<code>-v</code>	<code>--verbose</code>	Выводит 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 удаляемых каталогов не пусты.

Для удаления *непустых каталогов* предназначена команда `rm`, о которой сейчас и пойдет речь.

## Удаление файлов и непустых каталогов

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 [ключи]... [файл]...
```

Необязательный параметр *ключи* принимает значения, перечисленные в табл. 13.6.

Таблица 13.6. Ключи команды `rm`

Сокращенный вариант	Полный вариант	Описание
<code>-f</code>	<code>--force</code>	Игнорирует ошибки выполнения и не запрашивает подтверждения выполнения операций
<code>-i</code>		Выводит запрос на подтверждение перед каждым удалением
<code>-I</code>		Выдает запрос на подтверждение в том случае, когда удаляются сразу более 3 файлов, или перед рекурсивным удалением
	<code>--interactive [=когда]</code>	Выводит запрос на подтверждение в соответствии с переданным аргументом ключом <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>	При рекурсивном удалении иерархии пропускает все те каталоги, которые находятся в другой файловой системе, что и соответствующий аргумент командной строки
	<code>--no-preserve-root</code>	Не обрабатывает корневую файловую систему особым образом
	<code>--preserve-root</code>	Предотвращает удаление корневой файловой системы
<code>-r, -R</code>	<code>--recursive</code>	Рекурсивно удаляет каталоги и их содержимое
<code>-d</code>	<code>--dir</code>	Удаляет пустые каталоги

Таблица 13.6 (окончание)

Сокращенный вариант	Полный вариант	Описание
-v	--verbose	Сопровождает все 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 корневого файловой системы*. При выполнении этой команды не запрашиваются никакие подтверждения на удаление благодаря ключу `-f`.

Очень часто данную команду предлагают выполнить тем неопытным пользователям, которые обращаются за советом на какой-либо форум. Причем делается это не из злого умысла (не всегда), 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 предлагать выполнить "команду самоуничтожения" операционной системы.

Примечательно то, что на эту удочку 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 не просто понять, и многие сразу же попытались запустить его, чем и поплатились по доброту потерей личных файлов.

После большого ажиотажа вокруг команды `rm -rf /` многие разработчики операционных систем 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 наличии 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 она 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 [ключи]... источник каталог_(назначение)
```

Необязательный параметр *ключи* принимает значения, перечисленные в табл. 13.7.

Таблица 13.7. Ключи команды `cp`

Сокращенный вариант	Полный вариант	Описание
-a	--archive	Идентичен ключам <code>-dPR</code> и <code>--preserve=all</code>
	--attributes-only	Копирует только атрибуты файла, при этом данные файла не копируются

Таблица 13.7 (продолжение)

Сокращенный вариант	Полный вариант	Описание
	<code>--backup [=когда]</code>	Создает 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 можно установить 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, а затем повторная попытка его копирования. Параметр игнорируется, если также указан ключ <code>-n</code>
<code>-i</code>	<code>--interactive</code>	Выводит запрос перед тем, как переписывать данные. Параметр отменяет ранее указанный ключ <code>-n</code>
<code>-H</code>		Следует по символьным ссылкам, указанным в <i>источнике</i>
<code>-l</code>	<code>--link</code>	Вместо копирования создает жесткие ссылки
<code>-L</code>	<code>--dereference</code>	Всегда следует символьным ссылкам в <i>источнике</i>
<code>-n</code>	<code>--no-clobber</code>	Не перезаписывает существующий файл. Параметр отменяет ранее указанный ключ <code>-i</code>
<code>-P</code>	<code>--no-dereference</code>	Не следует по символьным ссылкам, указанным в <i>источнике</i>
<code>-p</code>		Равнозначен ключу <code>--preserve=mode,ownership,timestamps</code>
	<code>--preserve [=список_атрибутов]</code>	Сохраняет указанные атрибуты. По умолчанию атрибуты следующие: <code>mode</code> , <code>ownership</code> , <code>timestamps</code> . Если возможно, производится попытка сохранить дополнительные атрибуты: <code>context</code> , <code>links</code> , <code>xattr</code> , <code>all</code>
	<code>--parents</code>	Позволяет добавить исходный путь к каталогу_ (назначения)
<code>-R, -r</code>	<code>--recursive</code>	Рекурсивное копирование каталогов

Таблица 13.7 (окончание)

Сокращенный вариант	Полный вариант	Описание
	<code>--reflink[=<i>когда</i>]</code>	Контролирует создаваемые копии. Если указан параметр <code>--reflink=always</code> , выполняется облегченное копирование, при котором блоки данных копируются только при изменении. Если это невозможно или указан параметр <code>--reflink=auto</code> , то используется обычное копирование
	<code>--remove-destination</code>	Удаляет каждый файл назначения перед попыткой его открыть
	<code>--sparse[=<i>когда</i>]</code>	Управляет созданием разреженных файлов. С ключом <code>--sparse=always</code> всегда создается разреженный файл <i>каталога_ (назначения)</i> вне зависимости от того, содержит ли <i>источник</i> достаточно длинные последовательности нулевых байтов. Ключ <code>--sparse=never</code> служит для запрета создания разреженных файлов. По умолчанию используется значение <code>--sparse=auto</code> , что означает автоматическое создание разреженных файлов <i>каталога_ (назначения)</i>
	<code>--strip-trailing-slashes</code>	Удаляет все конечные косые черты из каждого аргумента <i>источник</i>
<code>-s</code>	<code>--symbolic-link</code>	Создает символичные ссылки вместо копирования
<code>-S</code>	<code>--suffix=<i>суффикс</i></code>	Использует для запасных копий заданный <i>суффикс</i>
	<code>--target-directory=<i>каталог</i></code>	Копирует все <i>источники</i> в <i>каталог_ (назначение)</i>
<code>-T</code>	<code>--no-target-directory</code>	Считает <i>каталог_ (назначение)</i> обычным файлом
<code>-u</code>	<code>--update</code>	Копирует только тогда, когда исходный файл новее, чем файл назначения, или когда файл назначения отсутствует
<code>-v</code>	<code>--verbose</code>	Сопровождает все производимые действия соответствующими пояснениями
<code>-x</code>	<code>--one-file-system</code>	Предписывает оставаться только в пределах одной файловой системы
<code>-Z</code>		Устанавливает контекст безопасности SELinux
	<code>--context[=<i>CTX</i>]</code>	Аналогичен параметру <code>-Z</code> . Если указан <i>CTX</i> , то устанавливает контекст безопасности SELinux или SMACK равным <i>CTX</i>
	<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`.

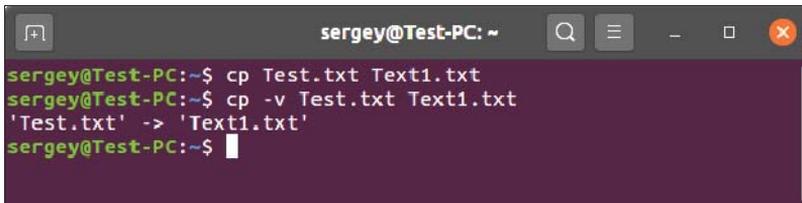
Итак, выполните следующую команду:

```
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
```

A screenshot of a terminal window titled "sergey@Test-PC: ~". The terminal shows the following commands and output:

```
sergey@Test-PC:~$ cp Test.txt Text1.txt
sergey@Test-PC:~$ cp -v Test.txt Text1.txt
'Test.txt' -> 'Text1.txt'
sergey@Test-PC:~$
```

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).

Иногда бывают задачи, когда требуется скопировать сразу несколько файлов в другой каталог. Как быть в этой ситуации: копировать каждый файла 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/
```

```

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

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 переименовывать файлы и каталоги.

Общий синтаксис команды следующий:

```
mv [ключи]... источник... каталог
```

Необязательный параметр *ключи* принимает значения, перечисленные в табл. 13.8.

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.

Таблица 13.8. Ключи команды `mv`

Сокращенный вариант	Полный вариант	Описание
	<code>--backup[=<i>контроль</i>]</code>	Создает резервные копии целевых файлов
<code>-b</code>		Идентичен параметру <code>--backup</code> , но без аргумента
<code>-f</code>	<code>--force</code>	Переписывает существующие файлы, не спрашивая при этом подтверждения со стороны пользователя
<code>-i</code>	<code>--interactive</code>	Спрашивает подтверждение со стороны пользователя перед тем, как переписывать файл или каталог
<code>-n</code>	<code>--no-clobber</code>	Не переписывает 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>	Удаляет все конечные косые черты из каждого аргумента <i>источника</i>
<code>-S</code>	<code>--suffix=<i>суффикс</i></code>	Перекрывает обычный суффикс для резервных копий
<code>-t</code>	<code>--target-directory=<i>каталог</i></code>	Перемещает все <i>источники</i> в <i>каталог</i>
<code>-T</code>	<code>--no-target-directory</code>	Считает параметр <i>каталог</i> обычным файлом
<code>-u</code>	<code>--update</code>	Перемещает, только если <i>источник</i> новее, чем файл назначения, или когда файл назначения отсутствует
<code>-v</code>	<code>--verbose</code>	Сопровождает все действия соответствующими пояснениями
<code>-Z</code>	<code>--context</code>	Задаёт контекст безопасности SELinux файлу назначения как у типа по умолчанию
	<code>--help</code>	Выводит справочную информацию
	<code>--version</code>	Выводит информацию о версии

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). Она позволяет отправить содержимое файла на стандартный вывод, т. е. на монитор, а также создавать файлы. Синтаксис команды:

```
cat [ключи]... [файл]...
```

Необязательный параметр *ключи* принимает значения, перечисленные в табл. 13.9.

Если команде `cat` не передавать никаких параметров или просто передать символ `-`, она будет копировать стандартный ввод на стандартный вывод. Это означает, что все, что будет вводиться в Терминале или Консоли, будет напоминать обычный

Таблица 13.9. Ключи команды `cat`

Сокращенный вариант	Полный вариант	Описание
-A	--show-all	Аналогичен ключу <code>-vET</code>
-b	--number-nonblank	Нумерует непустые строки при выводе
-e		Подавляет указанный ключ <code>-n</code> . Аналогичен ключу <code>-vE</code>
-E	--show-ends	Показывает символ <code>\$</code> в конце каждой строки
-n	--number	Нумерует все строки при выводе
-s	--squeeze-blank	Выводит не более одной пустой строки при повторе
-t		Аналогичен ключу <code>-vT</code>
-T	--show-tabs	Показывает символы табуляции в виде символов <code>^I</code>
-u		В настоящее время данный параметр игнорируется
-v	--show-nonprinting	Использует запись с символами <code>^</code> и <code>M-</code> , за исключением символов перевода строки и табуляции
	--help	Выводит справочную информацию
	--version	Выводит информацию о версии

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.

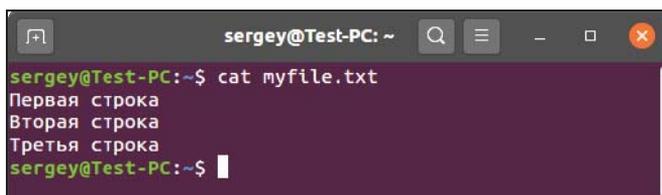
В качестве примера рассмотрим уже знакомый файл `myfile.txt`, находящийся в домашнем каталоге пользователя. Предположим, что файл имеет некоторый текст и этот текст нам нужно вывести в Терминал. Для этого выполним такую команду:

```
cat myfile.txt
```

Терминал отобразит содержимое файла `myfile.txt` (рис. 13.5).

Если нам требуется получить копию содержимого файла `myfile.txt`, то придется выполнить следующую команду:

```
cat myfile.txt > copy.txt
```



```

sergey@Test-PC: ~
sergey@Test-PC:~$ cat myfile.txt
Первая строка
Вторая строка
Третья строка
sergey@Test-PC:~$

```

Рис. 13.5. Вывод содержимого файла в Терминал

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:

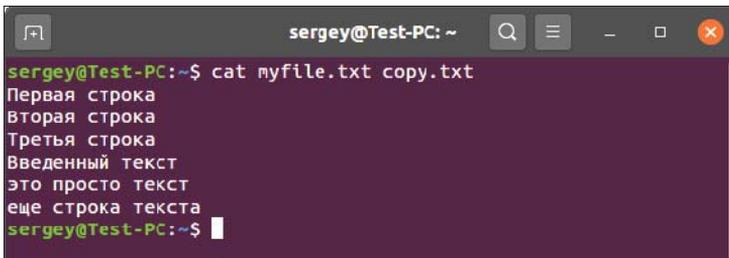
```
cat > copy.txt
```

Теперь попробуйте вводить различный тест 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 — *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
sergey@Test-PC:~$
```

Рис. 13.6. Терминал отобразил содержимое файлов `myfile.txt` и `copy.txt`

Для того 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 `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 [ключи]... [файл]...
```

Необязательный параметр *ключи* принимает значения, перечисленные в табл. 13.10.

Таблица 13.10. Ключи команды `tac`

Сокращенный вариант	Полный вариант	Описание
<code>-b</code>	<code>--before</code>	Присоединяет разделитель к началу, а не к концу строки
<code>-r</code>	<code>--regex</code>	Воспринимает разделитель как регулярное выражение
<code>-s</code>	<code>--separator=строка</code>	Использует в качестве разделителя <i>строку</i> , а не символ новой строки

В качестве примера выведем содержимое файла `myfile.txt` в обратном порядке:

```
tac myfile.txt
```

```

sergey@Test-PC: ~$ tac myfile.txt
Третья строка
Вторая строка
Первая строка
sergey@Test-PC: ~$ █

```

Рис. 13.7. Команда `tac` вывела строки из файла в обратной последовательности

После выполнения команды строки из файла `myfile.txt` будут выведены в обратной последовательности (рис. 13.7).

В данном примере наш файл содержит всего три строки. А если строк 30 или даже 300? Как раз для таких больших файлов существуют специальные команды постраничного 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 [ключи] <файл>...
```

При передаче команде `more` имени файла она постранично выводит содержимое файла в окно Терминала. Для того чтобы увидеть следующую страницу, нужно нажать клавишу <Пробел>, а для прокрутки текста на одну строку служит клавиша <Enter>. Для окончания просмотра 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 своей предшественницы (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 [ключи] [файл...]`

Необязательный параметр *ключ* принимает значения, перечисленные в табл. 13.11.

**Таблица 13.11.** Ключи команды `less`

Ключи (короткий и/или полный вариант)	Описание
<code>-a</code> <code>--search-skip-screen</code>	При поиске пропускает текущий экран (страницу)
<code>-A</code> <code>--SEARCH-SKIP-SCREEN</code>	Начинает поиск с текущей позиции, т. е. с начала текущей страницы
<code>-b [N]</code> <code>--buffers=[N]</code>	Устанавливает размер буфера для чтения. По умолчанию размер буфера равен 64 Кбайт
<code>-B</code> <code>--auto-buffers</code>	Отменяет автоматическое выделение размера буфера для блоков стандартного вывода
<code>-c</code> <code>--clear-screen</code>	Очищает экран вместо прокрутки данных
<code>-d</code> <code>--dumb</code>	Создает дамп Терминала
<code>-D [xn.n]</code> <code>--color=xn.n</code>	Сменяет цвет стандартного устройства вывода. Параметр поддерживается, только если в качестве стандартного устройства вывода используется клиентская система MS-DOS
<code>-e, -E</code> <code>--quit-at-eof</code> <code>--QUIT-AT-EOF</code>	Выходит из программы при достижении конца файла
<code>-f</code> <code>--force</code>	Принудительно открывает нерегулярные файлы, т. е. файлы с нетекстовой информацией
<code>-F</code> <code>--quit-if-one-screen</code>	Выходит из программы в том случае, если содержимое файла умещается на первом экране
<code>-g</code> <code>--hilite-search</code>	Выделяет только последний результат при поиске определенного значения
<code>-G</code> <code>--HILITE-SEARCH</code>	Не выделяет совпадения, даже если такие были найдены
<code>-h [N]</code> <code>--max-back-scroll=[N]</code>	Устанавливает ограничения на обратную прокрутку, т. е. ограничивает объем тех данных, которые будут доступны при обратной прокрутке
<code>-i</code> <code>--ignore-case</code>	Игнорирует регистр при поиске, даже если искомое значение не в верхнем регистре
<code>-I</code> <code>--IGNORE-CASE</code>	Полностью игнорирует регистр независимо от регистра искомой фразы
<code>-j [N]</code> <code>--jump-target=[N]</code>	Отображает текущую позицию в окне вывода

Таблица 13.11 (продолжение)

Ключи (короткий и/или полный вариант)	Описание
-J --status-column	Отображает столбец статуса слева от экрана вывода
-k [файл] --lesskey-file=[файл]	Использует специальный файл lesskey, описывающий привязку клавиш к командам, которые будут выполняться при их нажатии
-K --quit-on-intr	Позволяет использовать сочетание клавиш <Ctrl>+<C> для выхода из режима вывода
-L --no-lessopen	Игнорирует переменную окружения LESSOPEN
-m, -M --long-prompt --LONG-PROMPT	Отображает информацию об объеме просмотренного в процентах
-n, -N --line-numbers --LINE-NUMBERS	Включает либо отключает нумерацию строк при выводе на экран
-o [файл] --log-file=[файл]	Копирует вывод в log-файл при стандартном способе ввода
-O [файл] --LOG-FILE=[файл]	Копирует вывод в log-файл, при этом прежнее содержимое файла будет перезаписано
-p [шаблон] --pattern=[шаблон]	Выполняет шаблон из командной строки
-P [prompt] --prompt=[prompt]	Устанавливает новый текст строки приветствия, которое отображается в нижней части экрана
-q, -Q --quiet --QUIET --silent --SILENT	Устанавливает "тихий режим", при котором на стандартном устройстве вывода никакие ошибки отображаться не будут
-r, -R --raw-control-chars --RAW-CONTROL-CHARS	Устанавливает так называемый RAW-режим, при котором будут отображены все управляющие символы
-s --squeeze-blank-lines	Опускает несколько пустых строк, идущих подряд
-S --chop-long-lines	Обрезает слишком длинные строки
-t [тег] --tag=[тег]	Отыскивает нужный тег
-T [tagsfile] --tag-file=[tagsfile]	Использует теги из указанного в параметре файла

Таблица 13.11 (окончание)

Ключи (короткий и/или полный вариант)	Описание
-u, -U --underline-special --UNDERLINE-SPECIAL	Добавляет вывод специальных символов в местах окончания и переноса строк
-w --hilite-unread	Выделяет первую строку на первом экране
-W --HILITE-UNREAD	Выделяет первую строку после каждого пролистывания вперед
-x [N[,...]] --tabs=[N[,...]]	Устанавливает табуляцию равной символам, указанным в передаваемом параметре
-X --no-init	При выводе не использует библиотеки и базы данных свойств терминала, а использует дисплей компьютерного терминала аппаратно-независимо
-y [N] --max-forw-scroll=[N]	Устанавливает лимит на прокрутку вперед через передаваемое в качестве параметра значение
-z [N] --window=[N]	Устанавливает размер окна через передаваемое в качестве параметра значение
-" [c[c]] --quotes=[c[c]]	Использует имена файлов, в которых содержатся символы "". Кавычки экранируются символами ""
~ --tilde	Отключает отображение тильды ~ в конце файла
-# [N] --shift=[N]	Устанавливает количество строк горизонтальной прокрутки. Значение 0 равно половине ширины экрана
-V --version	Отображает 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>	Прокрутка текста на одну строку вниз
<Пробел>	Прокрутка текста на одну страницу вниз

Таблица 13.12 (окончание)

Клавиша	Описание
<Home>	Установка курсора в начало текста
<End>	Установка курсора в конец текста
<v>	Переход в режим редактирования отображаемого текста
<q>	Выход из режима вывода

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 <Enter>, <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>, вам следует добавить ключ `-K`:

```
less -NK /var/log/syslog
```

Теперь при нажатии клавиш <Ctrl>+<C> программа будет завершать свою работу. Нажатие клавиши <q> также будет работоспособно.

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 средствами командной строки, а правильнее 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] [ключ]
```

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, необязательный параметр *ключ* принимает значения, перечисленные в табл. 13.13.

**Таблица 13.13.** Ключи команды `find`

Ключи	Описание
<code>-name</code>	Ищет объект по заданному имени, при этом учитывается регистр символов
<code>-iname</code>	Ищет объект по заданному имени, при этом регистр символов не учитывается
<code>-type</code>	Тип искомого объекта. Может принимать следующие значения: <ul style="list-style-type: none"> <li>• <code>f</code> — файл (file);</li> <li>• <code>d</code> — каталог (directory);</li> <li>• <code>l</code> — ссылка (link);</li> <li>• <code>p</code> — канал (pipe);</li> <li>• <code>s</code> — сокет (socket)</li> </ul>
<code>-size</code>	Выполняет поиск по заданному размеру искомого объекта
<code>-atime</code>	Производит поиск с учетом последнего обращения к объекту. Время указывается в часах
<code>-mtime</code>	Выполняет поиск с учетом последнего изменения объекта. Время указывается в днях
<code>-mmin</code>	Выполняет поиск с учетом последнего изменения объекта. Время указывается в минутах
<code>-print</code>	Отображает на экране результат поиска
<code>-empty</code>	Выполняет поиск пустых каталогов

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
find: '/home/lost+found': Отказано в доступе
/home/sergey/myfile.txt
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 огромным, поэтому попробуем уточнить запрос, указав команде, что размер файла больше 10 Мбайт:

```
sudo find /home/ -mtime +5 -size +10M
```

Результат будет 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", которая 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
<b>Параметры для регулярных выражений</b>	
-E --extended-regexp	Рассматривает <i>шаблон</i> как расширенное регулярное выражение
-F --fixed-regexp	Рассматривает <i>шаблон</i> как список фиксированных регулярных выражений. Каждый образец ищется как строка, а не как регулярное выражение
-G --basic-regexp	Рассматривает <i>шаблон</i> как базовое регулярное выражение. Это поведение характерно для команды <code>grep</code> по умолчанию
-P --perl-regexp	Рассматривает <i>шаблон</i> как регулярное выражение языка Perl
-e --regexp=шаблон	Использует <i>шаблон</i> для поиска
-f --file=файл	Использует <i>шаблон</i> для поиска из <i>файла</i>
-i --ignore-case	Не учитывает регистр при поиске
-w --word-regexp	Указывает поиску, что <i>шаблон</i> должен подходить ко всем словам
-x --line-regexp	Указывает поиску, что <i>шаблон</i> должен подходить ко всем строкам в поиске
-z --null-data	При поиске разделяет строки байтом с нулевым значением, а не символом конца строки
<b>Управление выводом результата</b>	
-m --max-count=число	Останавливает поиск после достижения указанного <i>числа</i> совпадений
-b --byte-offset	Выводит вместе с выходными строками смещение в байтах
-n --line-number	Печатает номер строки вместе с выводимыми данными
--line-buffered	Сбрасывает буфер после каждой строки
-H --with-filename	Печатает имя файла для каждого совпадения

Таблица 13.14 (продолжение)

Опции	Описание
-h --no-filename	Не начинает вывод с имени файла
--label= <i>метка</i>	Использует <i>метку</i> в качестве имени файла для стандартного ввода
-o --only-matching	Показывает только часть строки, совпадающей с заданным шаблоном
-q --quiet --silent	Выводит в выходной поток только сопоставившиеся строки. Так, когда одна из входных строк соответствует образцу, возвращается статус выхода "0"
--binary-files= <i>тип</i>	Тип двоичного файла. Параметр <i>тип</i> файла может быть следующим: <code>binary</code> — бинарный; <code>text</code> — текстовый; <code>without-match</code> — не подходящий ни под один из перечисленных типов
-a --text	Аналогичен параметру <code>--binary-files=text</code>
-I	Аналогичен параметру <code>--binary-files=without-match</code>
-d --directories= <i>действие</i>	Способ обработки каталогов в соответствии с переданным значением <i>действие</i> . Параметр <i>действие</i> может принимать следующие значения: <code>read</code> — читать содержимое каталога; <code>recurse</code> — рекурсивно обрабатывать каталог; <code>skip</code> — пропускать содержимое каталогов
-r --recursive	Аналогичен ключу <code>--directories=recurse</code>
-R --dereference-recursive	Рекурсивно обрабатывает содержимое каталогов, переходя по всем символьным ссылкам
--include= <i>ф_шаблон</i>	Обрабатывает только файлы, подпадающие под <i>ф_шаблон</i>
--exclude= <i>ф_шаблон</i>	Пропускает файлы и каталоги, подпадающие под <i>ф_шаблон</i>
--exclude-from= <i>файл</i>	Пропускает файлы, подпадающие под шаблон файлов из <i>файла</i>
--exclude-dir= <i>шаблон</i>	Пропускает каталоги, подпадающие под <i>шаблон</i>
-L --files-without-match	Печатает только имена <i>файлов</i> без совпадений
-l --files-with-matches	Печатает только имена <i>файлов</i> с совпадениями
-c --count	Печатает только количество совпадающих строк из <i>файла</i>
-T --initial-tab	При необходимости выравнивает результат при помощи табуляции
-Z --null	Печатает байт "0" после имени <i>файла</i>

Таблица 13.14 (окончание)

Опции	Описание
<b>Управление контекстом</b>	
-B --before- context= <i>число</i>	Выводит <i>число</i> строк предшествующего контекста
-A --after-context= <i>число</i>	Выводит <i>число</i> строк последующего контекста
-C --context [=число]	Выводит <i>число</i> строк контекста
--color [=когда] --colour [=когда]	Использует маркеры для различия совпадающих строк. Параметр <i>когда</i> может принимать значения: <i>always</i> — всегда; <i>never</i> — никогда; <i>auto</i> — автоматически
-U --binary	Не удаляет 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	Выдает смещения без учета символов CR (символ возврата каретки)
<b>Прочие параметры</b>	
-s --no-messages	Подавляет вывод сообщений об ошибках
-v --invert-match	Выбирает 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

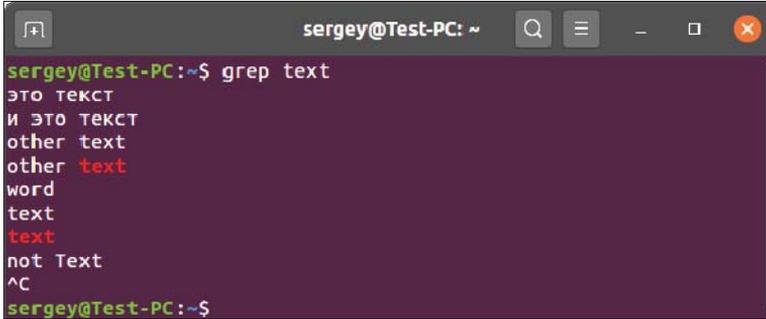
Символ	Описание
.	Любой произвольный символ
*	Множитель. Предыдущий символ может повторяться любое число раз
.*	Любая последовательность символов
^ <i>шаблон</i>	Привязка шаблона к началу строки
<i>шаблон</i> \$	Привязка шаблона к концу строки
[ <i>СИМВОЛЫ</i> ]	Указывает на любой из символов, которые 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 выполните следующую команду:

```
grep text
```

Теперь попробуйте вводить различный текст, а также слово *text* и понаблюдайте за результатом (рис. 13.9).



```
sergey@Test-PC: ~$ grep text
это текст
и это текст
other text
other text
word
text
text
not Text
^C
sergey@Test-PC: ~$
```

Рис. 13.9. Результат работы команды `grep` без параметров

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 цветом. 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:

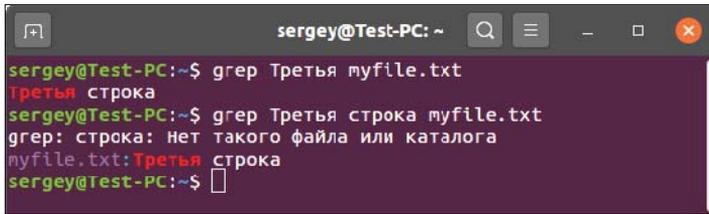
```
grep Третья myfile.txt
```

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 два слова: "Третья строка"?

```
grep Третья строка myfile.txt
```

И вот что получится (рис. 13.10).



```
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, состоящей из двух слов

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. 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. слова "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. Таким образом, правильная команда будет такой:

```
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.

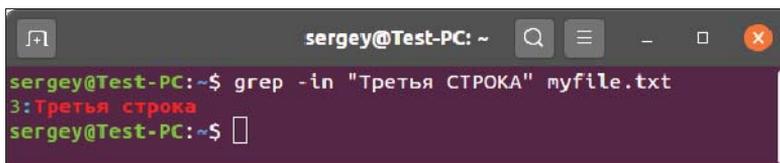
Давайте еще раз посмотрим на составленный нами запрос и попробуем его модернизировать так, чтобы получить больше информации об искомой фразе. Ничего не приходит на ум? Тогда выполните следующую конструкцию команды `grep`:

```
grep -in "Третья СТРОКА" myfile.txt
```

В этом примере мы добавили ключи `-i` и `-n`, которые позволяют игнорировать регистр при поиске (ключ `-i`) и выводить номер строки (ключ `-n`). Саму искомую фразу мы напечатали в разных регистрах: 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).

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 *m*, а за-



```
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

канчивается 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. Итак, у нас есть шаблон `^т.*ка`, обрамленный кавычками. Как мы помним, кавычки нужны для того, чтобы у нас была возможность искать не одно слово, а несколько. К тому же использование кавычек делает шаблон нагляднее. Однако сейчас нас интересует то, что находится между кавычками.

Первый символ `^` (каре) 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 т. к. 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.

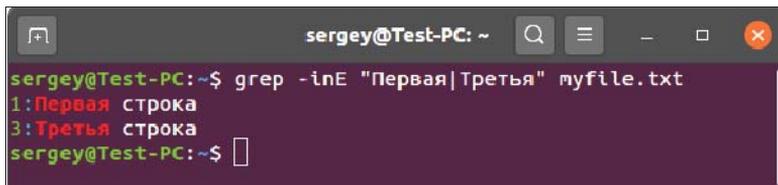
Такая с виду сложная конструкция шаблона поиска позволяет найти строку, состоящую из двух слов.

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 (`|`), которая нужна для перечисления искомых слов.

Результатом выполнения данной конструкции будет список, содержащий строки из двух слов, которые находятся на разных строчках (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 строчках

Для того чтобы просмотреть количество вхождений искомого шаблона в файле `myfile.txt`, выполните следующую команду:

```
grep -ic "строка" myfile.txt
```

Команда выведет количество вхождений искомой строки, т. е. количество повторений искомого слова. Обратите внимание, что мы использовали ключ `-c` и убрали ключ `-n`, выводящий номер строки, которая содержит искомый шаблон.

До сих пор мы работали с содержимым файлов и могли заметить, что команда `grep` просматривает только их содержимое, но не имена файлов. Однако это лишь 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: нам нужно получить список всех файлов в домашнем каталоге, имена которых заканчиваются на `".txt"`. 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 (`|`). 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'  
cat_file.txt  
copy.txt  
myfile.txt  
Sample.txt  
Test.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 эта программа 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 ОС 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 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 питание компьютера. Однако она 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. Именно поэтому 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 рассмотрим синтаксис команды `shutdown`:

```
shutdown [опции...] [время] [сообщение...]
```

Необязательный параметр *опции* принимает значения, перечисленные в табл. 13.16.

Таблица 13.16. Опции команды `shutdown`

Опции	Описание
-H --halt	Завершение работы компьютера
-P --poweroff	Завершение работы операционной системы
-r --reboot	Перезагрузка компьютера

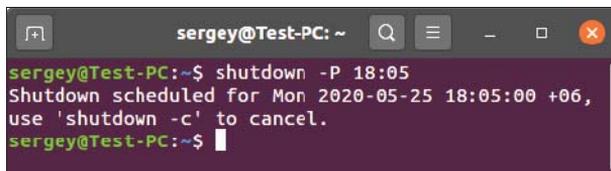
Таблица 13.16 (окончание)

Опции	Описание
-h	Аналогичен параметру <code>--poweroff</code>
-k	Отправляет предупреждение о том, что компьютер будет выключен или перезагружен
<code>--no-wall</code>	Не отправляет сообщения о перезагрузке или выключении компьютера
-c	Отменяет запланированное отключение или перезагрузку компьютера
<code>--help</code>	Выводит справочную информацию

Давайте на примерах убедимся в возможностях команды `shutdown`. Для начала запланируем выключение компьютера в 18:05. Для этого выполним команду:

```
shutdown -P 18:05
```

В ответ на это мы получим сообщение со статусом запланированной операции (рис. 13.14).



```
sergey@Test-PC: ~
sergey@Test-PC:~$ shutdown -P 18:05
Shutdown scheduled for Mon 2020-05-25 18:05:00 +06,
use 'shutdown -c' to cancel.
sergey@Test-PC:~$
```

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.

Для того чтобы перезагрузить компьютер прямо сейчас, выполните 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 и выключить питание.

**ПРИМЕЧАНИЕ**

Для перезагрузки существует еще 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 наибольшие права. Пользователю этой учетной записи доступны практически все полномочия в системе, в разумных пределах конечно, 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) равный 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 требуются 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 ее (записью) управлением. Эта привилегия только на первый взгляд приносила удобства в работе с ОС. На деле же пользователи получили большое количество проблем, связанных с безопасностью их системы.

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 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 о том, какую же команду использовать. Давайте разберемся в этом спорном вопросе и расставим все точки над `i`.

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 должна была заменить 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. 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 подвергался изменению), в то время как пользователь, выполнявший команду su, должен был волшебным образом 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 перечислены в табл. 14.1.

**Таблица 14.1.** Параметры команды sudo

Параметр	Описание
-A --askpass	Использует вспомогательную программу для ввода пароля
-b --background	Выполняет указанную команду в фоновом режиме

Таблица 14.1 (продолжение)

Параметр	Описание
-C --close-from= <i>num</i>	Закрывает все дескрипторы файлов, которые больше или равны значению, переданному в параметре <i>num</i>
-E --preserve-env	Сохраняет пользовательское окружение при выполнении команды
-e --edit	Редактирует файлы вместо выполнения команды
-g --group= <i>группа</i>	Выполняет команду от имени или ID указанной в параметре <i>группа</i>
-H --set-home	Устанавливает для переменной HOME домашний каталог
-h --host= <i>хост</i>	Выполняет команду на узле при условии, если таковая поддерживается модулем ядра
-i --login	Запускает оболочку входа в систему от имени указанного пользователя, а также задает команду, которая будет выполнена при входе в систему
-K --remove-timestamp	Полностью удаляет файл <i>timestamp</i>
-k --reset-timestamp	Позволяет объявить недействительным файл <i>timestamp</i>
-l --list	Показывает список прав пользователя или проверяет заданную команду
-n --non-interactive	Использует автономный режим без вывода запросов пользователю
-P --preserve-groups	Сохраняет вектор группы вместо установки целевой группы
-p --prompt= <i>prompt</i>	Использует указанный запрос пароля
-r --role= <i>роль</i>	Создает контекст безопасности SELinux с указанной ролью, переданной в параметре <i>роль</i>
-S --stdin	Читает пароль из стандартного ввода
-s --shell	Запускает оболочку от имени указанного пользователя, а также задает команду, которая будет выполнена при запуске оболочки
-t --type= <i>тип</i>	Создает контекст безопасности SELinux указанного <i>типа</i>
-T --command-timeout= <i>timeout</i>	Прекращает выполнение команды по истечении времени, указанного в параметре
-U --other-user= <i>пользователь</i>	В режиме списка показывает права пользователя

Таблица 14.1 (окончание)

Параметр	Описание
-u --user=ПОЛЬЗОВАТЕЛЬ	Выполняет команду (или редактирует файл) от имени или ID указанного пользователя
-v --validate	Обновляет временную метку пользователя без выполнения команды
--	Прекращает обработку аргументов командной строки

Как вы видите, параметров у команды `sudo` предостаточно.

В качестве примера 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 увидеть длину пароля.

Просто введите свой пароль и нажмите 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
```

Здесь первый параметр `-l` дает возможность вывести список прав пользователя. Второй параметр `-U` позволяет задать имя конкретного пользователя, которое мы

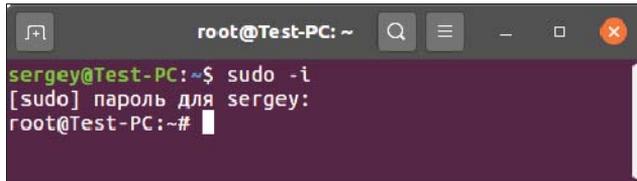
передаем последним параметром. В нашем случае мы узнаем права пользователя sergey. Этим же методом мы можем узнать право пользователя root, передав команде `sudo` в качестве последнего параметра имя пользователя `root`.

"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).



```

root@Test-PC: ~
sergey@Test-PC:~$ sudo -i
[sudo] пароль для sergey:
root@Test-PC:~#

```

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
<code>root</code>	Имя учетной записи пользователя
<code>@</code>	Символ разделителя
<code>Test-PC</code>	Имя компьютера
<code>:</code>	Символ разделителя
<code>~</code>	Каталог выполнения команды. В нашем случае команды выполняются в домашнем каталоге пользователя <code>root</code>
<code>#</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).

```
sergey@Test-PC:~$
```

а

```
root@Test-PC:~#
```

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, поменялись 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**

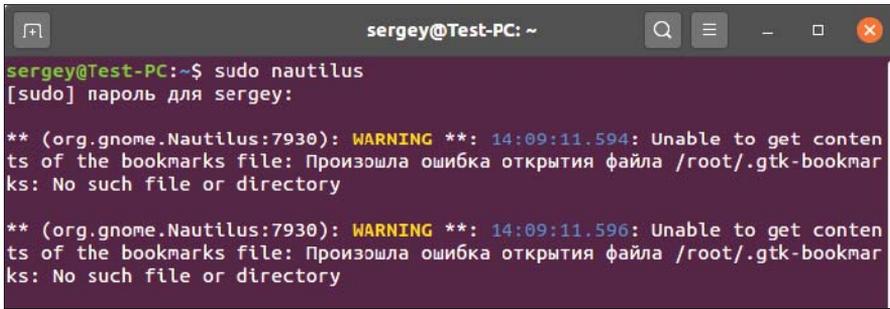
Теперь, когда мы умеем запускать программы с административными привилегиями, стоит сказать несколько слов о запуске графических программ в административном режиме.

К примеру, если вы запустите файловый менеджер Nautilus следующей командой:

```
sudo nautilus
```

в Терминале появятся ошибки, как на 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 нажатием сочетания клавиш `<Ctrl>+<C>`, то будет закрыт файловый менеджер Nautilus.

"Почему это происходит?" — спросите вы. Дело в том, что изначально все команды, запускаемые в ОС Ubuntu Linux, предназначались для работы в консольном



```

sergey@Test-PC: ~
sergey@Test-PC:~$ sudo nautilus
[sudo] пароль для sergey:

** (org.gnome.Nautilus:7930): WARNING **: 14:09:11.594: Unable to get contents of the bookmarks file: Произошла ошибка открытия файла /root/.gtk-bookmarks: No such file or directory

** (org.gnome.Nautilus:7930): WARNING **: 14:09:11.596: Unable to get contents of the bookmarks file: Произошла ошибка открытия файла /root/.gtk-bookmarks: No such file or directory

```

Рис. 14.4. Запуск Nautilus через Терминал

режиме, т. е. без графической оболочки. К тому же, когда мы пытаемся запустить Nautilus с правами администратора, то этот файловый менеджер по умолчанию ищет в каталоге `/root/` системные файлы, например `/root/.gtk-bookmarks`. Разумеется, данного файла не существует, и поэтому выводится соответствующее уведомление. Именно поэтому Терминал таким образом "отреагировал" на запуск Nautilus с правами администратора.

Если же запустить Nautilus с правами вашей учетной записи, то никаких сообщений об ошибке вы не увидите.

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` — команда `pkexec`, которая пришла на смену устаревшей команде `gksu`. `pkexec` не требует установки, и запуск Nautilus производится следующей командой:

```
pkexec env DISPLAY=$DISPLAY XAUTHORITY=$XAUTHORITY nautilus
```

В ответ на это `pkexec` запросит ваш пароль уже не в Терминале, а в графическом окне (рис. 14.5).

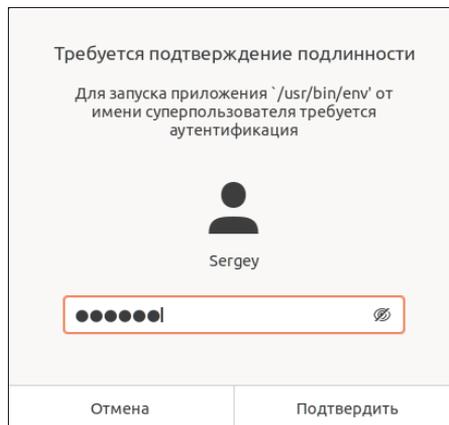
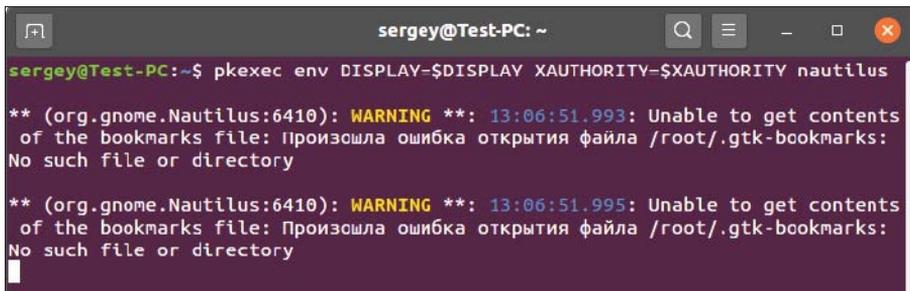


Рис. 14.5. Запрос пароля пользователя

При этом файловый менеджер Nautilus будет также искать свои конфигурационные файлы в каталоге суперпользователя root, ведь именно с правами этой учетной записи мы запустили Nautilus (рис. 14.6).

A terminal window titled 'sergey@Test-PC: ~' with a search icon, menu icon, and window control buttons. The terminal shows the command 'pkexec env DISPLAY=\$DISPLAY XAUTHORITY=\$XAUTHORITY nautilus' being executed. Below the command, two identical warning messages are displayed: '\*\* (org.gnome.Nautilus:6410): WARNING \*\*: 13:06:51.993: Unable to get contents of the bookmarks file: Произошла ошибка открытия файла /root/.gtk-bookmarks: No such file or directory'.

```
sergey@Test-PC:~$ pkexec env DISPLAY=$DISPLAY XAUTHORITY=$XAUTHORITY nautilus
** (org.gnome.Nautilus:6410): WARNING **: 13:06:51.993: Unable to get contents
of the bookmarks file: Произошла ошибка открытия файла /root/.gtk-bookmarks:
No such file or directory
** (org.gnome.Nautilus:6410): WARNING **: 13:06:51.993: Unable to get contents
of the bookmarks file: Произошла ошибка открытия файла /root/.gtk-bookmarks:
No such file or directory
```

Рис. 14.6. Предупреждение о том, что в домашнем каталоге пользователя root не существует конфигурационного файла

Не стоит пугаться этого предупреждения, потому что оно никак не повлияет на работу файлового менеджера Nautilus. Для того чтобы закрыть файловый менеджер, достаточно либо закрыть само окно Nautilus, либо нажать сочетание клавиш <Ctrl>+<C> в терминале.

Сейчас у вас возникнет резонный вопрос: а нельзя ли как-то упростить запуск программ с графическим интерфейсом с помощью команды `pkexec`? Да, можно. Выполните для этого следующую команду:

```
alias pkexec='pkexec env DISPLAY=$DISPLAY XAUTHORITY=$XAUTHORITY'
```

Данной командой мы создали алиас (псевдоним) для переменной окружения команды `pkexec`, и теперь команда запуска графического приложения сократится до следующего удобного варианта:

```
pkexec nautilus
```

#### ПРИМЕЧАНИЕ

Стоит отметить, что создание псевдонима для команды `pkexec` доступно только в течение жизни терминальной сессии. После того как вы закроете Терминал, вам снова нужно будет задавать переменную окружения для данной команды.

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 его 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`. При вызове `Visudo` временно блокируется файл `/etc/sudoers` и создается 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).

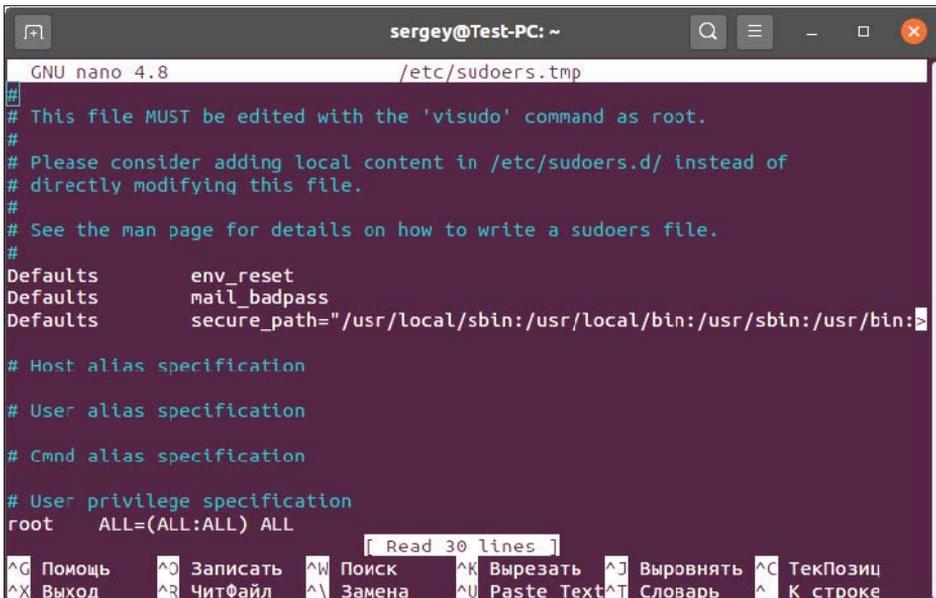


Рис. 14.7. Запущен редактор `visudo`

Перемещение курсора в редакторе visudo осуществляется с помощью клавиш-стрелок, а не посредством 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).

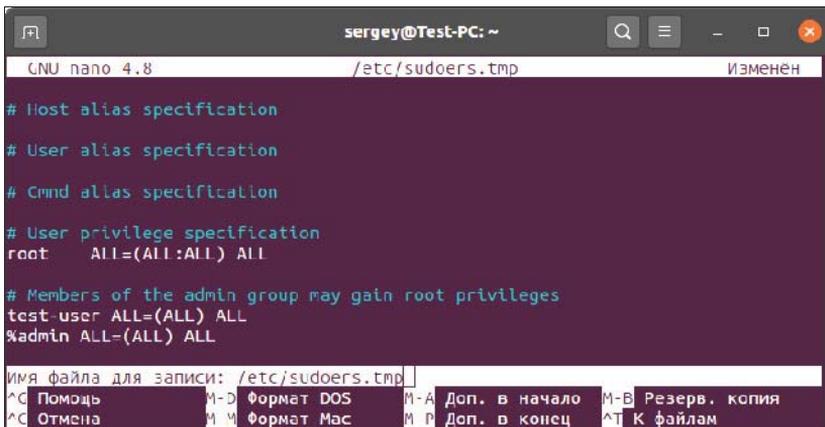


Рис. 14.8. Сохранение изменений в файле /etc/sudoers

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>. Будет выведено предупреждение о том, что файл /etc/sudoers существует, и появится вопрос, следует ли его перезаписать (рис. 14.9).

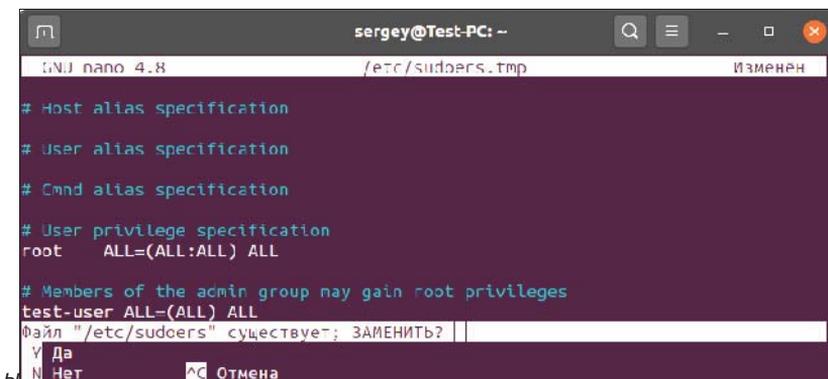


Рис. 14.9. Внесение изменений в оригинальный файл

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:

```
Cmnd_Alias MY_CMD = /usr/sbin/visudo, /usr/bin/apt
```

Теперь поясним, что означает эта запись.

`Cmnd_Alias` означает, что следующее за ним слово будет являться псевдонимом для набора команд. В нашем случае `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. Это 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 представленному на рис. 14.10.

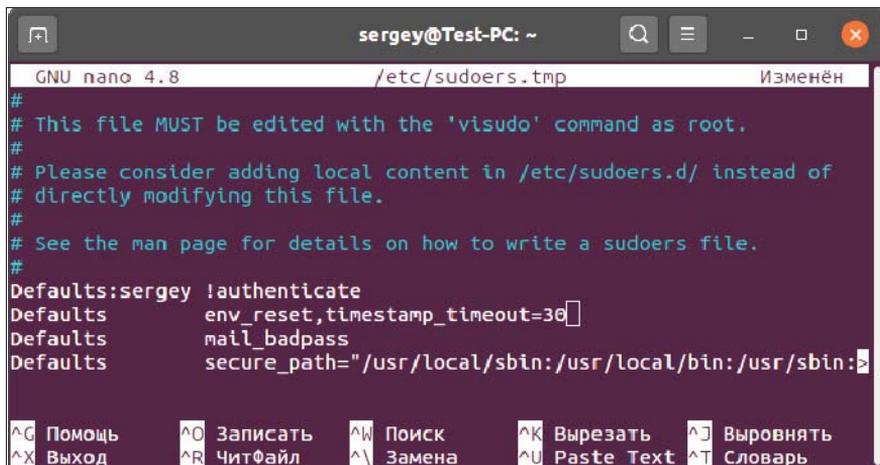


Рис. 14.10. Задание времени действия введенного пароля

В нашем случае число 30 означает время действия *sudo-сессии*, которое равно 30 минутам. Если вы хотите задать другой промежуток времени действия *sudo-сессии*, 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 особой осторожностью.

Мы также научились запускать графические приложения с административными привилегиями и редактировать конфигурацию прав на использование команды `sudo`.



# 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.

Для того чтобы просмотреть список доступных проприетарных драйверов в вашей ОС, в меню поиска приложений найдите и запустите утилиту **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 в Сети.

Если у вас пустой список дополнительных драйверов, это значит, что ваша ОС полностью поддерживает установленное оборудование в вашем компьютере и вам не требуется дополнительная установка драйверов.

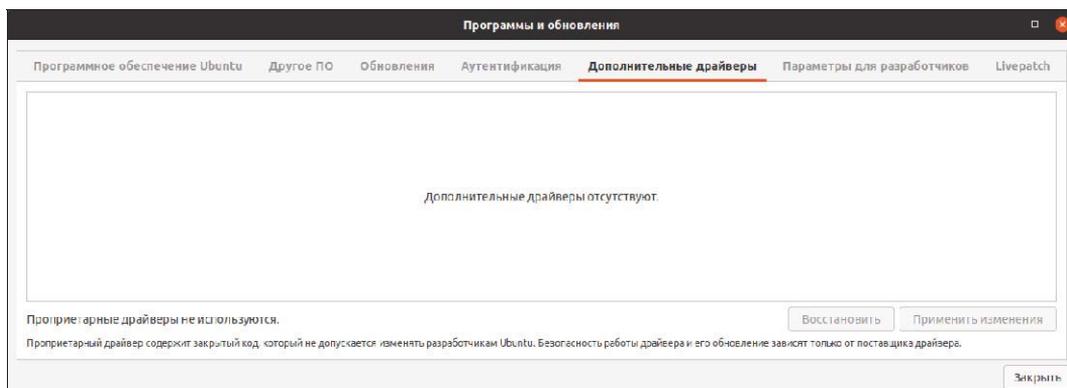


Рис. 15.1. Доступные проприетарные драйверы

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 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.

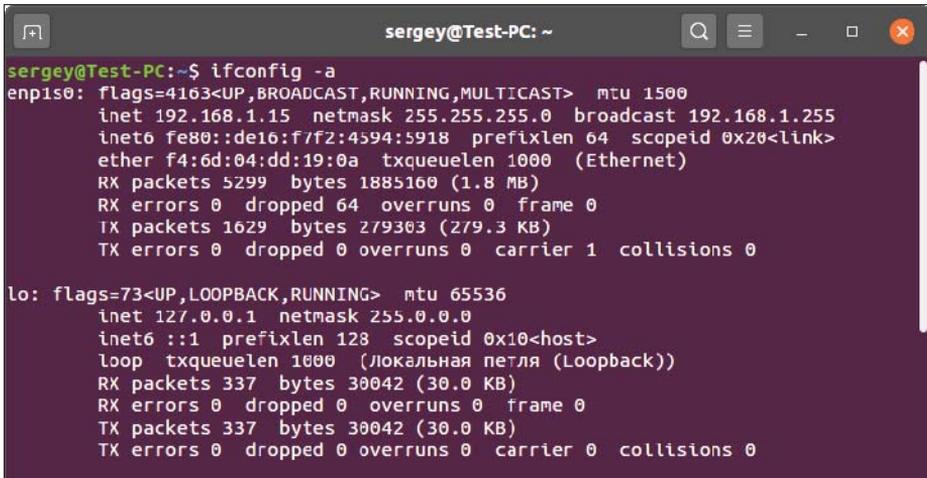
Для начала нам нужно понять, нашла ли операционная система сетевую карту. Для этого в Терминале выполните поочередно следующие команды:

```
sudo apt install net-tools
ifconfig -a
```

Первая команда устанавливает сетевые утилиты, которые позволяют работать с сетью и производить их диагностику. Вторая команда выводит параметры сетевых интерфейсов.

Если повезло, то вывод команды будет примерно таким, как на 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: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.15 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::de16:f7f2:4594:5918 prefixlen 64 scopeid 0x20<link>
    ether f4:6d:04:dd:19:0a txqueuelen 1000 (Ethernet)
    RX packets 5299 bytes 1885160 (1.8 MB)
    RX errors 0 dropped 64 overruns 0 frame 0
    TX packets 1629 bytes 279363 (279.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 1 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Локальная петля (Loopback))
    RX packets 337 bytes 30042 (30.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 337 bytes 30042 (30.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Рис. 15.2. Отображение доступных сетевых интерфейсов

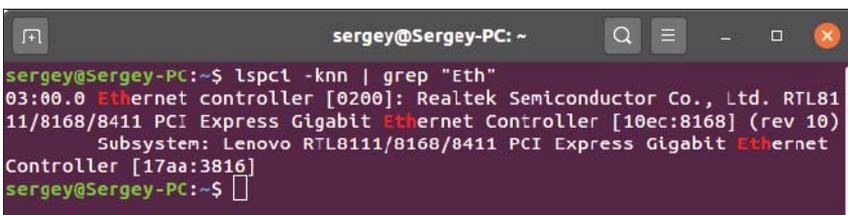
**ПРИМЕЧАНИЕ**

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 таким, как рис. 15.3.



```

sergey@Sergey-PC: ~
sergey@Sergey-PC:~$ lspci -knn | grep "Eth"
03:00.0 Ethernet controller [0200]: Realtek Semiconductor Co., Ltd. RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller [10ec:8168] (rev 10)
Subsystem: Lenovo RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller [17aa:3816]
sergey@Sergey-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 <https://www.realtek.com/en/>, и в разделе **Downloads | Communications Network ICs** найти нашу модель сетевой карты (рис. 15.4).

Как видим, в списке значатся три пункта с похожим названием. Какой из этих пунктов вам выбрать? Выбирайте любой, т. к. страница загрузки для этих ревизий чипов одинаковая, а следовательно, драйвер будет один. Теперь, перейдя по ссылке



Рис. 15.4. Поиск драйвера для сетевой карты

**Software**, выберите версию драйвера для ОС Linux и сохраните файл в домашнем каталоге.

Откройте Терминал и разархивируйте скачанный файл драйвера:

```
tar vjxf r8101-1.035.03.tar.bz2
```

#### **ПРИМЕЧАНИЕ**

В нашем случае файл архива с драйвером носит название r8101-1.035.03.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 r8101-1.035.03.

Нам нужно убедиться в том, что у нас не установлен встроенный модуль сетевой карты. Для этого выполните следующую команду:

```
lsmod | grep r8101
```

#### **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 нужно удалить. Удаление модуля производится следующей командой:

```
sudo rmmod r8101
```

После этого перейдите в каталог с распакованным исходным кодом драйвера сетевой карты:

```
cd r8101-1.035.03
```

Нам нужно запустить скрипт установки драйвера:

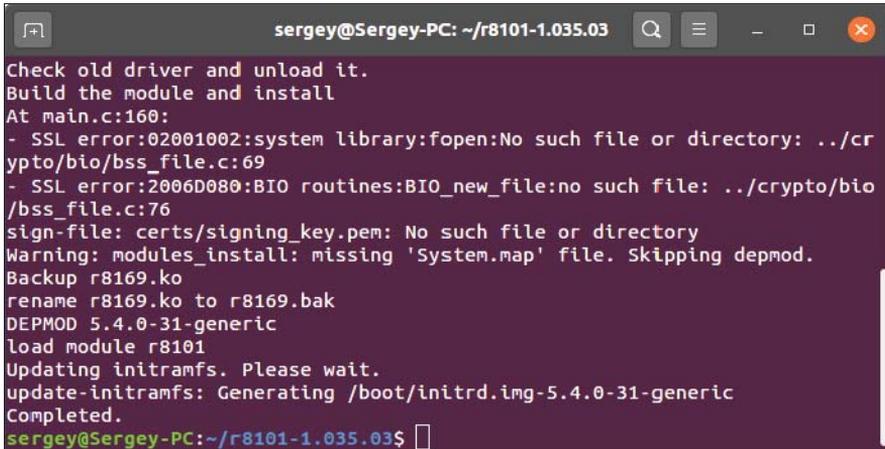
```
sudo ./autorun.sh
```

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 r8101
ifconfig -a
```

В первой команде мы проверяем, присутствует ли в списке модулей ядра модуль сетевой карты, а вторая команда отображает список доступных сетевых интерфейсов.

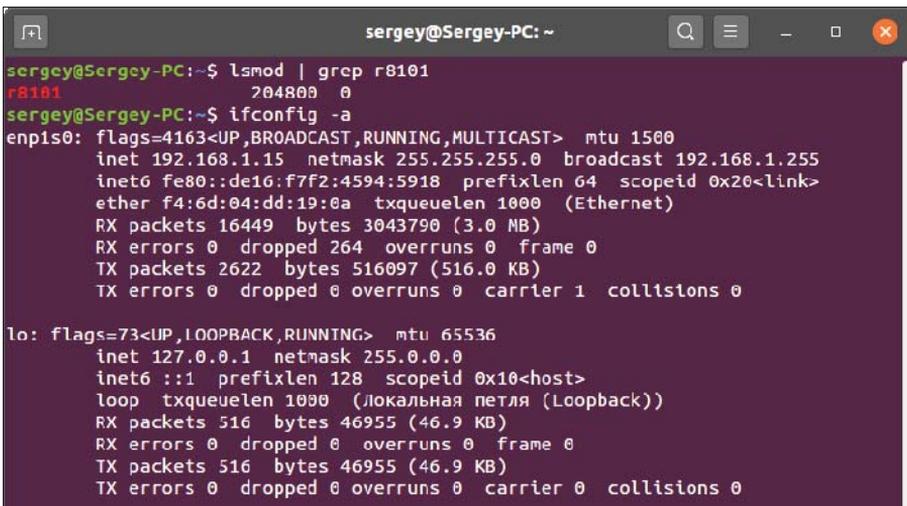


```
sergey@Sergey-PC: ~/r8101-1.035.03
Check old driver and unload it.
Build the module and install
At main.c:160:
- SSL error:02001002:system library:fopen:No such file or directory: ../cr
ypto/bio/bss_file.c:69
- SSL error:2006D080: BIO routines: BIO_new_file: no such file: ../crypto/bio
/bss_file.c:76
sign-file: certs/signing_key.pem: No such file or directory
Warning: modules_install: missing 'System.map' file. Skipping depmod.
Backup r8169.ko
rename r8169.ko to r8169.bak
DEPMOD 5.4.0-31-generic
load module r8101
Updating initramfs. Please wait.
update-initramfs: Generating /boot/initrd.img-5.4.0-31-generic
Completed.
sergey@Sergey-PC:~/r8101-1.035.03$
```

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@Sergey-PC:~$ lsmod | grep r8101
r8101                204800  0
sergey@Sergey-PC:~$ ifconfig -a
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.15 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::de16:f7f2:4594:5918 prefixlen 64 scopeid 0x20<link>
    ether f4:6d:04:dd:19:0a txqueuelen 1000 (Ethernet)
    RX packets 16449 bytes 3043790 (3.0 MB)
    RX errors 0 dropped 264 overruns 0 frame 0
    TX packets 2622 bytes 516097 (516.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 1 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (локальная петля (Loopback))
    RX packets 516 bytes 46955 (46.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 516 bytes 46955 (46.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

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.

## Общий метод установки драйвера беспроводной сети

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.

Мы рассмотрим управление драйвером беспроводной сети Wi-Fi на примере сетевой карты D-Link DWA-525.

### **ПРИМЕЧАНИЕ**

Сразу стоит сделать оговорку, что драйвер сетевой карты DWA-525 успешно устанавливается системой по умолчанию и корректно работает. Сам же Wi-Fi-адаптер D-Link DWA-525 на данный момент не имеет «свежих» драйверов под текущую операционную систему. Поэтому рассматривать установку драйверов для этого адаптера мы будем в качестве абстрактного примера, т. к. аналогичным образом выполняется установка драйверов других Wi-Fi-адаптеров.

Для чего может понадобиться управление драйвером беспроводной сети? Допустим такую ситуацию: у вас имеется установленный Wi-Fi-адаптер, который работает некорректно или вовсе не работает в ОС Linux. Для того чтобы заставить его работать, придется сначала выяснить, установлен ли драйвер, и установить его вручную, если этого не произошло.

Для начала нам нужно проверить, присутствует ли в списке интерфейсов сетевая карта для беспроводной сети:

```
ifconfig -a
```

Как мы можем убедиться, Wi-Fi-адаптер не отображается в списке интерфейсов (рис. 15.7).

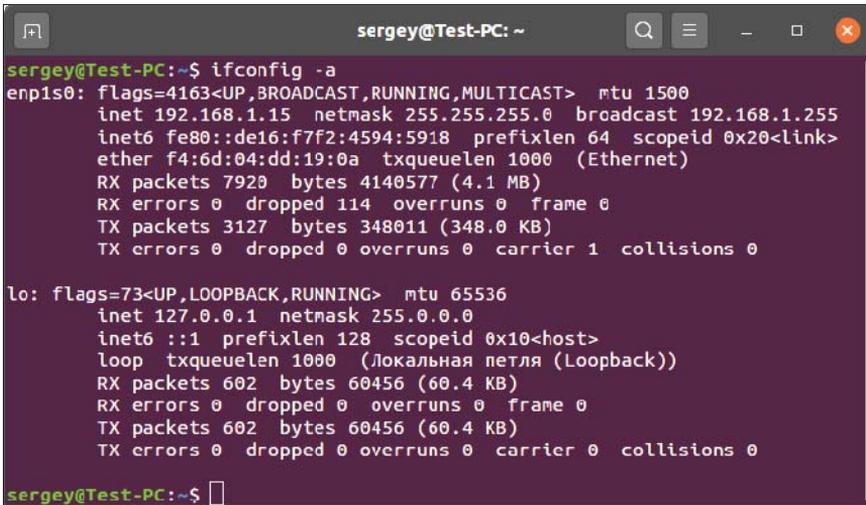
Отсутствие беспроводной сетевой карты в списке интерфейсов вовсе не означает, что ее нет физически. Далее наша задача — выяснить название и ее модель:

```
lspci -knn | grep "Network"
```

А вот и показалась сама беспроводная карточка Wi-Fi (рис. 15.8).

Теперь нам известны название и модель сетевой карты Wi-Fi — Ralink RT5360. Остается только отправиться на официальный сайт производителя <http://dlink.ru> и скачать там драйвер устройства. Для этого лучше всего воспользоваться внутренним поисковым механизмом сайта, запустив поиск по модели устройства DWA-525. После этого в результатах выдачи щелкните по найденной ссылке, а затем перейдите в соответствующий раздел загрузки драйверов — **Загрузки**. На момент написания этих строк ссылка выглядела следующим образом:

**<http://dlink.ru/ru/products/2/1290.html>**



```

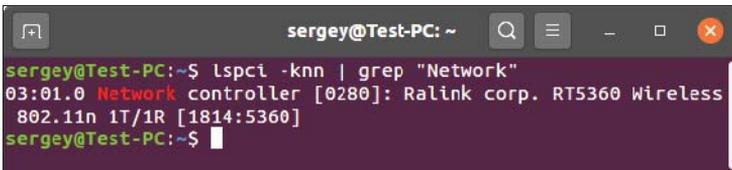
sergey@Test-PC: ~
sergey@Test-PC:~$ ifconfig -a
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.15 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::de16:f7f2:4594:5918 prefixlen 64 scopeid 0x20<link>
    ether f4:6d:04:dd:19:0a txqueuelen 1000 (Ethernet)
    RX packets 7920 bytes 4140577 (4.1 MB)
    RX errors 0 dropped 114 overruns 0 frame 0
    TX packets 3127 bytes 348011 (348.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 1 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Локальная петля (Loopback))
    RX packets 602 bytes 60456 (60.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 602 bytes 60456 (60.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

sergey@Test-PC:~$

```

Рис. 15.7. Беспроводная сетевая карта отсутствует в списке интерфейсов



```

sergey@Test-PC: ~
sergey@Test-PC:~$ lspci -knn | grep "Network"
03:01.0 Network controller [0280]: Ralink corp. RT5360 Wireless
802.11n 1T/1R [1814:5360]
sergey@Test-PC:~$

```

Рис. 15.8. Вывод названия и модели беспроводной сетевой карты

Однако пользователей ОС Linux ожидает одна маленькая неприятность: в списке загрузок перечислены драйверы только для ОС Windows, а для ОС Linux драйверы не значатся. Как быть в таком случае и неужели производитель не предоставил драйверы для нашей сетевой карты? Не стоит паниковать, т. к. пользователи ОС Linux более опытные и умеют разбивать ссылки на составляющие и находить нужное. Давайте это сейчас и сделаем.

Для начала скопируйте ссылку на драйвер для ОС Windows, например вот эту:

[http://ftp.dlink.ru/pub/Wireless/DWA-525/Drivers/1.25/20151231\\_DWA-525\\_V1.25b03.rar](http://ftp.dlink.ru/pub/Wireless/DWA-525/Drivers/1.25/20151231_DWA-525_V1.25b03.rar)

Ничего не видите интересного? А теперь:

<http://ftp.dlink.ru/pub/Wireless/DWA-525/Drivers/>

Да, мы отсекали часть ссылки и теперь можем напрямую зайти на публичный FTP-сервер производителя и, перемещаясь по каталогам, найти нужный нам драйвер.

Перейдите по ссылке <http://ftp.dlink.ru/pub/Wireless/DWA-525/Drivers/> и откройте каталог **Linux**, а затем каталог с нужной ревизией сетевой карты.

#### ПРИМЕЧАНИЕ

К сожалению, программными средствами не всегда получается установить номер ревизии, поэтому придется заглянуть внутрь системного блока и найти эту информацию

на сетевой карте либо порыться в документации к сетевой карте, которая осталась у вас при покупке данной карты. В любом случае, если вы не смогли установить ревизию, то можно поочередно экспериментировать с установкой драйверов разных ревизий.

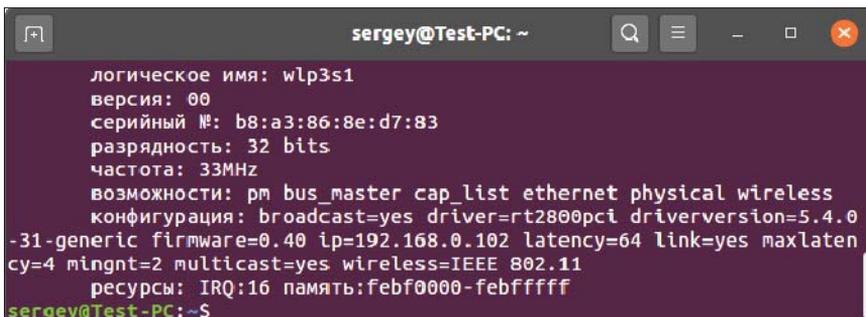
Скачайте драйвер для ОС Linux и сохраните файл в домашнем каталоге. Загруженный файл называется `2010_01_28_RT2860_Linux_STA_v2.3.0.0_Alpha_v2.zip`.

Теперь "подготовим" систему для последующей установки драйвера.

Первым делом необходимо удалить все лишнее, т. е. все то, что могло быть установлено вами либо операционной системой в попытках заставить Wi-Fi работать. Проверим, какой драйвер у нас задействован для беспроводной сетевой карты:

```
sudo lshw -C network
```

Этой командой мы проверяем, загружен ли в данный момент времени драйвер модуля сетевой карты. В результате мы увидим примерно следующую картину (рис. 15.9).



```
сергей@Test-PC: ~  
логическое имя: wlp3s1  
версия: 00  
серийный №: b8:a3:86:8e:d7:83  
разрядность: 32 bits  
частота: 33MHz  
возможности: pm bus_master cap_list ethernet physical wireless  
конфигурация: broadcast=yes driver=rt2800pci driverversion=5.4.0-31-generic  
firmware=0.40 ip=192.168.0.102 latency=64 link=yes maxlaten  
cy=4 mingnt=2 multicast=yes wireless=IEEE 802.11  
ресурсы: IRQ:16 память:febf0000-febfffff  
сергей@Test-PC: ~$
```

Рис. 15.9. Обнаружен модуль драйвера

Нас интересует строка "конфигурация":

```
конфигурация: broadcast=yes driver=rt2800pci driverversion=5.4.0-31-generic  
firmware=0.40 ip=192.168.0.102 latency=64 link=yes maxlatency=4 mingnt=2  
multicast=yes wireless=IEEE 802.11
```

Из этой строки мы узнаем, что используется драйвер `rt2800pci`, и разумным будет его отключить и добавить его в черный список:

```
sudo rmmmod rt2800pci  
echo "blacklist rt2800pci" | sudo tee -a /etc/modprobe.d/blacklist.conf
```

Первой командой мы удаляем драйвер, но для того чтобы он снова не установился при последующей перезагрузке системы, мы добавляем его в черный список. Логично предположить: так как сетевая карта установлена в компьютере, то при последующем включении компьютера операционная система попытается установить для нее соответствующий драйвер. Для этих целей мы и добавили этот модуль в черный список.

Теперь необходимо перезагрузить операционную систему.

Подготовительные действия закончены, и теперь нужно распаковать загруженный драйвер.

Содержимое архива `2010_01_28_RT2860_Linux_STA_v2.3.0.0_Alpha_v2.zip` распакуете в домашний каталог пользователя, не прибегая к командной строке. Для этого щелкните правой кнопкой мыши по архиву и выберите команду **Извлечь в текущую папку**.

### **ПРИМЕЧАНИЕ**

О том, как работать с архивами из командной строки, мы познакомимся чуть позже.

Для удобства переименуйте каталог с содержимым архива в `wifi`.

Запустите Терминал и сделайте каталог `wifi` текущим:

```
cd wifi
```

Теперь осталось запустить компиляцию драйвера, которая занимает считанные секунды:

```
make
```

После этого необходимо запустить команду установки драйвера:

```
make install
```

На этом установка драйвера будет завершена.

Теперь поговорим о том, как задействовать драйвер, который мы добавили в черный список. Для этого выполните команду в Терминале:

```
sudo modprobe rt2800pci
```

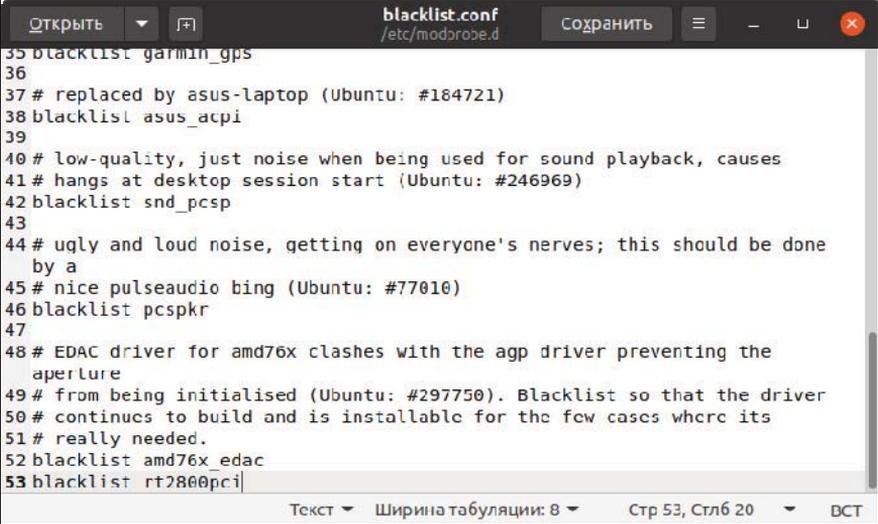
После ввода пароля вашей учетной записи модуль `rt2800pci`, который является драйвером беспроводной сети, будет запущен, но он будет загружен в память до выключения компьютера. Но как быть, если загрузка этого модуля нужна постоянно? Чуть ранее мы добавили этот модуль в черный список. Для того чтобы его удалить, откроем файл `/etc/modprobe.d/blacklist.conf` и переместимся в его конец, где находится запись об этом модуле (рис. 15.10), следующей командой:

```
sudo gedit /etc/modprobe.d/blacklist.conf
```

Удалите строку `blacklist rt2800pci` и сохраните файл. Теперь при последующем включении компьютера данный модуль будет автоматически загружаться.

Таким образом происходят выгрузка модулей ядра из памяти и их последующая блокировка от "самовольной" загрузки операционной системой.

Теперь вы знаете, как добавлять/удалять из черного списка тот или иной модуль. К данному методу иногда приходится прибегать при установке различных драйверов.



```
blacklist.conf
/etc/modprobe.d
Сохранить
Открыть
35 blacklist garmin_gps
36
37 # replaced by asus-laptop (Ubuntu: #184721)
38 blacklist asus_acpi
39
40 # low-quality, just noise when being used for sound playback, causes
41 # hangs at desktop session start (Ubuntu: #246969)
42 blacklist snd_pcsp
43
44 # ugly and loud noise, getting on everyone's nerves; this should be done
45 # by a
46 blacklist pulseaudio_bing (Ubuntu: #77010)
47
48 # EDAC driver for amd76x clashes with the agp driver preventing the
49 # from being initialised (Ubuntu: #297750). Blacklist so that the driver
50 # continues to build and is installable for the few cases where its
51 # really needed.
52 blacklist amd76x_edac
53 blacklist rt2800pci
```

Рис. 15.10. Модуль `rt2800pci` находится в черном списке

## Установка драйвера видекарты из репозитория

Если ваша ОС Ubuntu загрузилась в графическом режиме, 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 видекарты 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 видекарты 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 видекарты 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 вашего пароля введите его и для подтверждения изменения списка источников нажмите клавишу <Enter>. Будет выдан список пакетов из только что добавленного репозитория.

Убедиться в том, что новый источник был добавлен, можно на вкладке **Другое ПО** утилиты **Программы и обновления** (рис. 15.11).

Теперь нам нужно сообщить программе установки, что обновился список пакетов:

```
sudo apt update
```

Осталось обновить систему:

```
sudo apt 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 install -f` и повторить обновление списка пакетов и обновление системы.

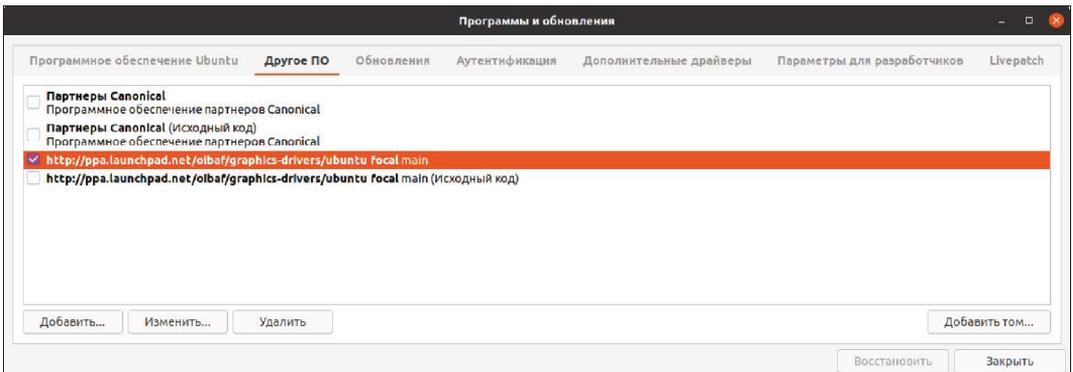


Рис. 15.11. Добавлен новый источник репозитория PPA

Это все, что касается установки более свежих драйверов, которые ставятся автоматически при обновлении системы. Помимо этого способа, существуют 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 поме-

нять концепцию для своего детища, компания Canonical решила запретить установку драйвера `fglrx` в ОС Ubuntu. Вместо этого теперь предлагается использовать драйвер `radeon` (устанавливается по умолчанию) или `amdgpu-pro`. 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-pro` 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).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ lspci -k| grep -EA2 'VGA'  
01:00.0 VGA compatible controller: Advanced Micro Devices, Inc. [AMD/ATI]  
RV515 PRO [Radeon X1300/X1550 Series]  
Subsystem: ASUSTeK Computer Inc. RV515 PRO [Radeon X1300/X1550 Ser  
ies]  
Kernel driver in use: radeon  
sergey@Test-PC:~$
```

Рис. 15.12. Получение информации о видеокарте и используемом видеодрайвере

У нас имеется видеокарта Radeon X1300 и используется драйвер `radeon`. Данный драйвер, точнее его модуль, используется ядром операционной системы и не является проприетарным драйвером.

Первым делом мы должны проверить, поддерживается ли наша видеокарта драйвером `amdgpu-pro`. Для этого на странице <https://www.amd.com/en/support/kb/release-notes/rn-prorad-lin-amdgpupro> в разделе **AMD Product Compatibility** найдите видеокарту Radeon X1300. На момент написания книги драйвером `amdgpu-pro` поддерживаются только новейшие видеокарты серии Radeon RX, Radeon R9, Radeon R7Radeon Pro WX и AMD FirePro. Нашей видеокарты в этом списке нет, значит, немного не повезло, и нам не удастся воспользоваться новой версией `amdgpu-pro`. По этой причине для видеокарты Radeon X1300 лучшим решением будет оставить стандартный драйвер `radeon`.

Но как же быть, если у вас установлена видеокарта, которая поддерживается драйвером `amdgpu-pro`? Давайте рассмотрим вариант установки `amdgpu-pro` 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-pro*.

**ПРИМЕЧАНИЕ**

На момент написания этих строк самой последней версией драйвера amdgpu была версия 167.40, которая расположена по ссылке <http://support.amd.com/en-us/kb-articles/Pages/AMDGPU-PRO-Driver-for-Linux-Release-Notes.aspx> и предназначена для версии ОС Ubuntu 16.04.

В разделе загрузки выберите ссылку **AMDGPU-Pro Driver Version 17.40 for Ubuntu 16.04.3**. Сохраните файл в домашнем каталоге. Теперь наш загруженный файл `amdgpu-pro-17.40-492261.tar.xz` нужно разархивировать:

```
tar -Jxvf amdgpu-pro-17.40-492261.tar.xz
```

Далее необходимо перейти в каталог с распакованными файлами:

```
cd amdgpu-pro-17.40-492261
```

и выполнить установку видеодрайвера:

```
./amdgpu-pro-install
```

Терминал запросит у вас пароль. Для подтверждения установки введите `y` и нажмите клавишу `<Enter>` (рис. 15.13).

```
sergey@Test-PC: ~/amdgpu-pro-17.40-492261
libstdc++6:i386 libtinfo5 libtinfo5:i386 libtinfo6:i386
libunistring2:i386 libvdpaui-amdgpu-pro:i386 libvdpaui-amdgpu-pro
libvdpaui:i386 libx11-6:i386 libx11-xcb1:i386 libxau6:i386
libxcb-dri2-0:i386 libxcb-dri3-0:i386 libxcb-glx0:i386
libxcb-present0:i386 libxcb-sync1:i386 libxcb-xfixes0:i386
libxcb1:i386 libxdamage1:i386 libxdmcp6:i386 libxext6:i386
libxfixes3:i386 libxshmfence1:i386 libxxf86vm1:i386 libzstd1:i386
mesa-vdpau-drivers:i386 opengl-amdgpu-pro-icd:i386
opengl-amdgpu-pro-icd vdpau-driver-all:i386 vulkan-amdgpu-pro:i386
vulkan-amdgpu-pro xserver-xorg-video-amdgpu-pro
xserver-xorg-video-glamoregl-amdgpu-pro zlib1g:i386
Обновлено 0 пакетов, установлено 76 новых пакетов, для удаления отмечено 0
пакетов, и 0 пакетов не обновлено.
Необходимо скачать 23,8 MB/136 MB архивов.
После данной операции объём занятого дискового пространства возрастёт на 6
83 MB.
Хотите продолжить? [Д/н]
```

Рис. 15.13. Терминал запрашивает подтверждение установки

После того как драйвер установится, выполните команду:

```
sudo usermod -a -G video $LOGNAME
```

Эта команда позволит добавить текущего пользователя в группу `video`. Осталось только перезагрузить систему:

```
shutdown -r now
```

Таким способом выполняется установка открытого драйвера `amdgpu-pro`.

В случае возникновения проблем в работе драйвера, например когда невозможно выполнить вход в графическую оболочку, нужно удалить установленный видеодрайвер `amdgpu-pro`. Делается это следующим образом: переключитесь в консоль-

ный режим (сочетание клавиш <Ctrl>+<Alt>+<F3>), затем после ввода имени и пароля выполните команду:

```
sudo amdgpu-pro-uninstall
```

Подтвердите удаление видеодрайвера `amdgpu-pro` и перезагрузите операционную систему.

## Установка аудиодрайвера

В Linux-подобных операционных системах за звук отвечает один-единственный аудиодрайвер, который носит название ALSA.

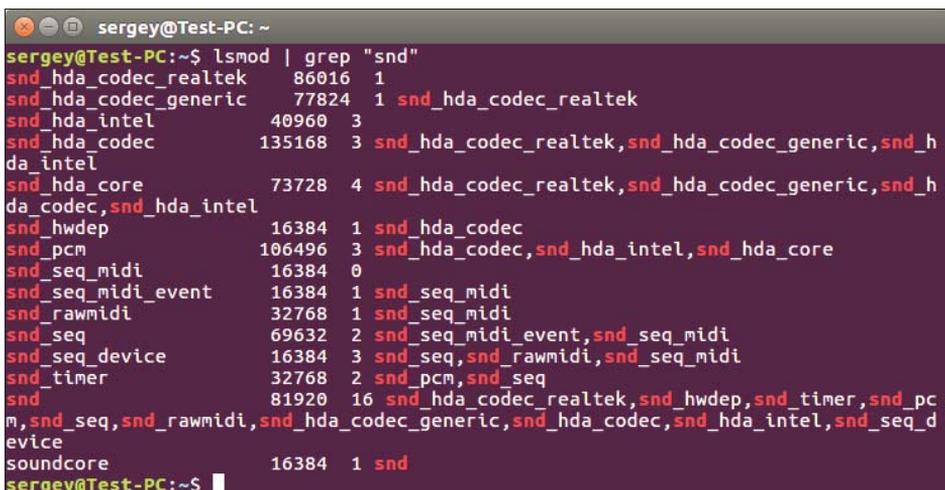
*ALSA* (Advanced Linux Sound Architecture — продвинутая звуковая архитектура Linux) — это целая архитектура звуковых драйверов, обеспечивающая поддержку огромного количества звуковых карт, начиная от старых моделей и заканчивая продвинутыми современными моделями.

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).

При выводе команды `lsmod | grep "snd"` все аудиодрайверы представлены в виде модулей ядра, названия которых начинаются с префикса `snd_`.



```
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

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 update
```

В ответ на эту команду будет загружен новый список аудиодрайверов из только что добавленного репозитория. Осталось лишь обновить систему:

```
sudo apt upgrade
```

Перезагрузите операционную систему и проверьте работоспособность звукового устройства.

## Установка ALSA из исходных текстов

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 из меню в левой части страницы выберем ссылку **Download**. На этой странице нам будут представлены следующие пакеты:

- ◆ **Drivers** — пакет содержит компоненты модуля ядра. На данный момент загружать этот 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)** — 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;
- ◆ PyALSA (pyalsa) — пакет содержит необходимые зависимости для работы некоторых библиотек звуковой архитектуры ALSA;
- ◆ OSS compat lib (alsa-oss) — пакет содержит набор библиотек, которые необходимы для обеспечения совместимости с устаревшей Open Sound System (OSS).

Нам необходимы только четыре пакета: Firmware, Library, Plugins и Utilities.

#### **ПРИМЕЧАНИЕ**

При желании, если вы являетесь истинным ценителем звука и любите более детально настраивать его качество, 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 четыре файла:

- ◆ alsa-firmware-1.2.1.tar.bz2;
- ◆ alsa-lib-1.2.2.tar.bz2;
- ◆ alsa-plugins-1.2.2.tar.bz2;
- ◆ alsa-utils-1.2.2.tar.bz2.

Распакуем эти архивы:

```
tar jxvf alsa-firmware-1.2.1.tar.bz2
tar jxvf alsa-lib-1.2.2.tar.bz2
tar jxvf alsa-plugins-1.2.2.tar.bz2
tar jxvf alsa-utils-1.2.2.tar.bz2
```

После выполнения этих команд в домашнем каталоге будут созданы четыре одноименных каталога. Как вы уже догадались, сейчас мы начнем устанавливать каждый пакет по отдельности.

Сначала установим пакет Firmware (alsa-firmware):

```
cd alsa-firmware-1.2.1
./configure
make
sudo make install
```

Первая команда делает текущим каталог alsa-firmware-1.2.1, 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.

Возвращаемся в домашний каталог, выполнив команду cd.

То же самое проделываем и с пакетом Library (alsa-lib):

```
cd alsa-lib-1.2.2
./configure
```

```
make
sudo make install
```

Теперь нужно установить плагины, т. е. пакет Plugins (alsa-plugins), предварительно вернувшись в домашний каталог:

```
cd alsa-plugins-1.2.2
./configure
make
sudo make install
```

И последним шагом будет установка утилит Utilities (alsa-utils). Для начала нужно выгрузить службу alsa:

```
sudo /sbin/alsa unload
```

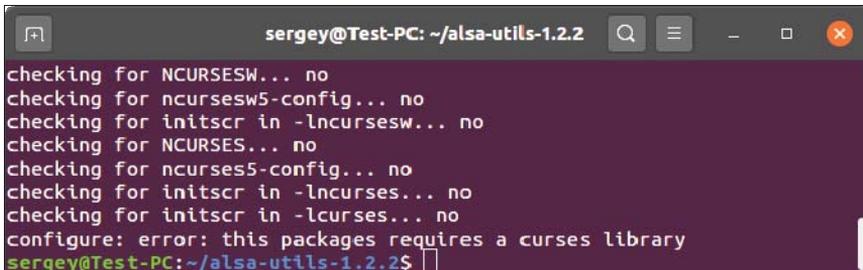
Перейдем в каталог с исходными текстами утилиты, предварительно вернувшись в домашний каталог:

```
cd alsa-utils-1.2.2
```

А теперь начинается самое интересное — попытаемся сконфигурировать наш набор утилит:

```
./configure
```

Не так тут все просто. Ошибочка вышла (рис. 15.15).



```
sergey@Test-PC: ~/alsa-utils-1.2.2
checking for NCURSESW... 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-1.2.2$
```

Рис. 15.15. Требуется установка библиотеки curses

Ошибка "this packages requires a curses library" означает, что для компиляции данного пакета в системе требуется библиотека curses. Сейчас мы ее установим:

```
sudo apt install libncurses5-dev
```

Если 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 немного измененную команду конфигурирования:

```
./configure --disable-alsaconf
```

Ключ `--disable-alsaconf` отключает сборку инструмента конфигурации `alsaconf`, который не совместим с `Udev` (управление устройствами для новых версий ядра Linux).

Как видите, теперь команда выполнена 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 (команда Терминала `alsamixer`) — 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).

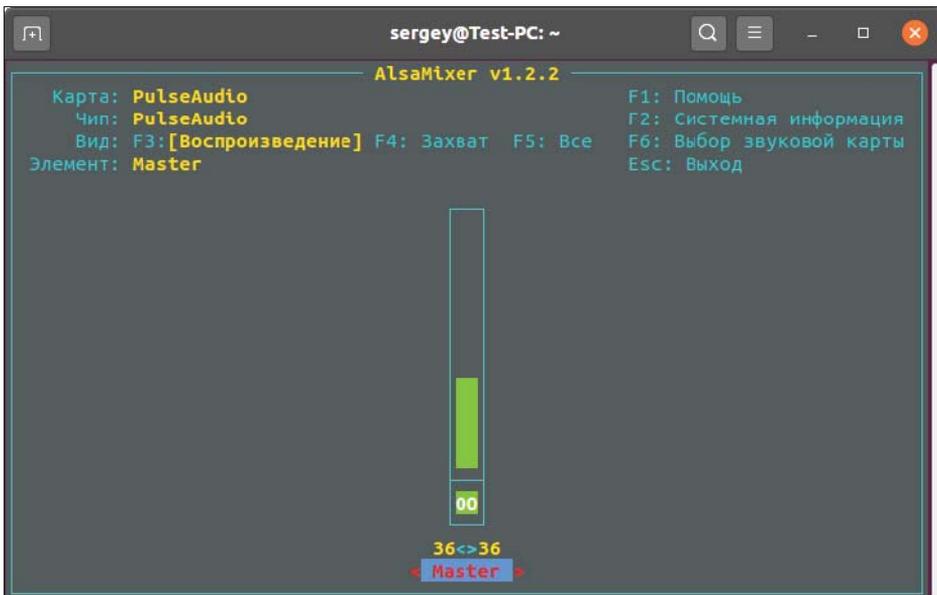


Рис. 15.16. Окно утилиты `AlsaMixer`

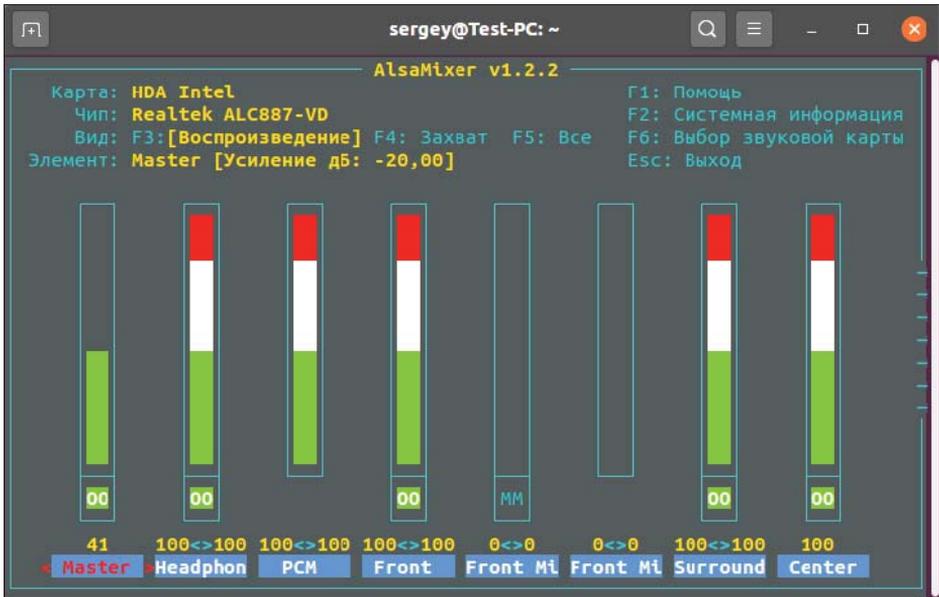


Рис. 15.17. Настройка звуковой карты в ALSaMixer

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, которые имеют название ALSA и содержат в себе все аудио-драйверы практически для любой звуковой карты.

## 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 таковых в ОС Windows. Основное различие состоит не только в самом процессе установки, а в том, что не всегда удается 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, 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: <https://global.brother.ru/>. Затем выберите раздел **Продукция/Поддержка**, 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>, а затем 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: <https://developers.hp.com>. Затем перейдите в раздел **Download** и скачайте программу HP Linux Imaging and Printing. Она 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 **Инструкция и загрузки**. 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.ru>. Затем перейдите в меню **Поддержка пользователей**, выберите ссылку **Поддержка и драйвера**. 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, даже если операционная система смогла определить ваш принтер (рис. 16.1), тем не менее без установленной службы CUPS вы просто не сможете распечатать нужный вам документ.

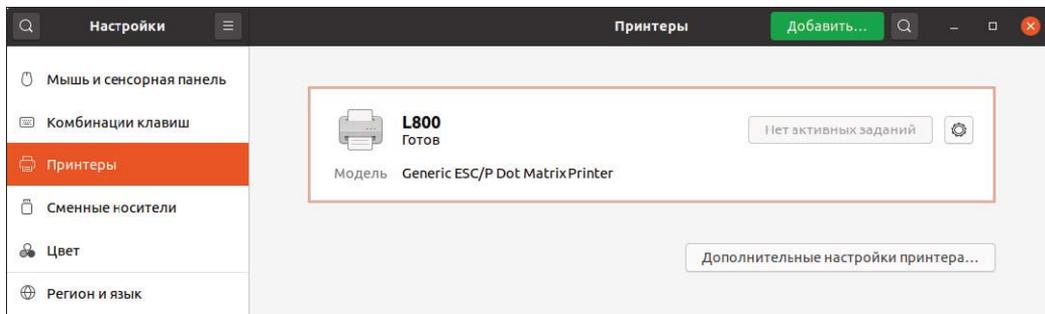


Рис. 16.1. Принтер определен системой

Изначально служба печати, т. е. служба CUPS, установлена в ОС. И даже если принтер успешно определен системой, ничего хорошего из этого не выйдет. Так, при попытке отправить документ на печать произойдет лишь видимость печати, поскольку принтер был установлен автоматически в обход службы CUPS. Операционная система при печати проигнорирует обращение к этой службе и выполнит видимость печати, т. е. документ будет оправлен на печать, но на самом деле не будет распечатан.

Поэтому для начала удалите из операционной системы принтер, который был автоматически установлен. Для этого откройте окно **Параметры** и перейдите на вкладку **Принтеры**. Затем напротив принтера нажмите кнопку с изображением

шестеренки  и выберите пункт **Удалить принтер**. Принтер будет удален. И можно будет переходить к его ручной установке.

### 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 операционной системы

Теперь, после того как вы удалили автоматически установленный принтер, можно переходить к рассмотрению варианта установки принтера Epson L800 средствами ОС Ubuntu Linux, т. е. без установки соответствующих драйверов. Процесс будет происходить с задействованием службы CUPS.

Для начала откройте окно **Параметры** и перейдите на вкладку **Принтеры**, если вы уже успели закрыть это окно. Отобразится окно управления принтерами, в котором необходимо нажать кнопку **Дополнительные настройки принтера....** Откроется окно добавления принтера в систему (рис. 16.2).

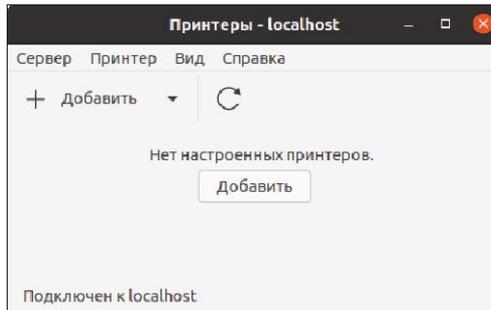


Рис. 16.2. Окно добавления принтера

В нашем случае установленные принтеры отсутствуют, т. к. мы предварительно их удалили. 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 наша модель не значится, однако даже из этой ситуации есть выход.

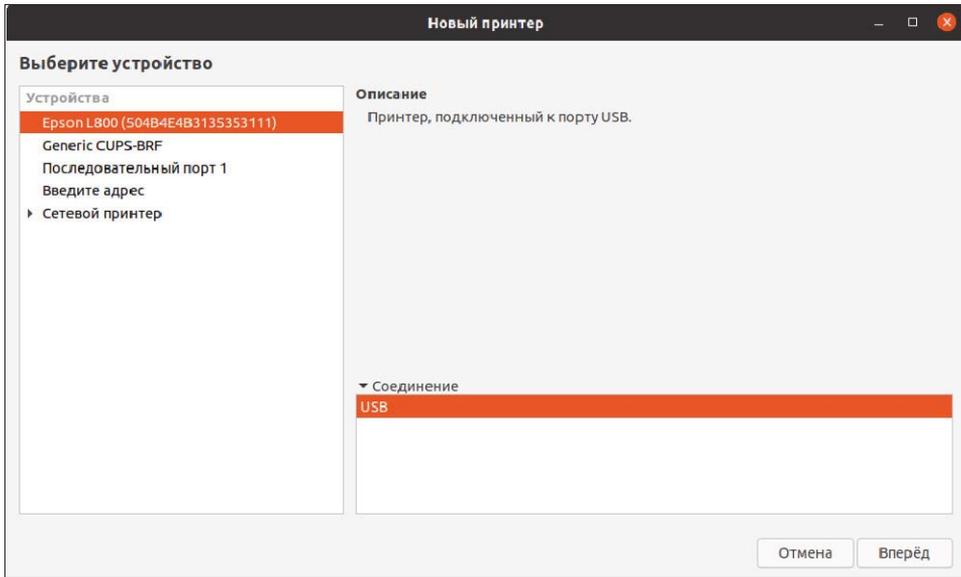


Рис. 16.3. Выбор принтера, подключенного к компьютеру

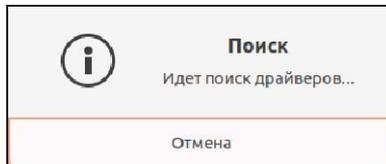


Рис. 16.4. Поиск драйверов для принтера

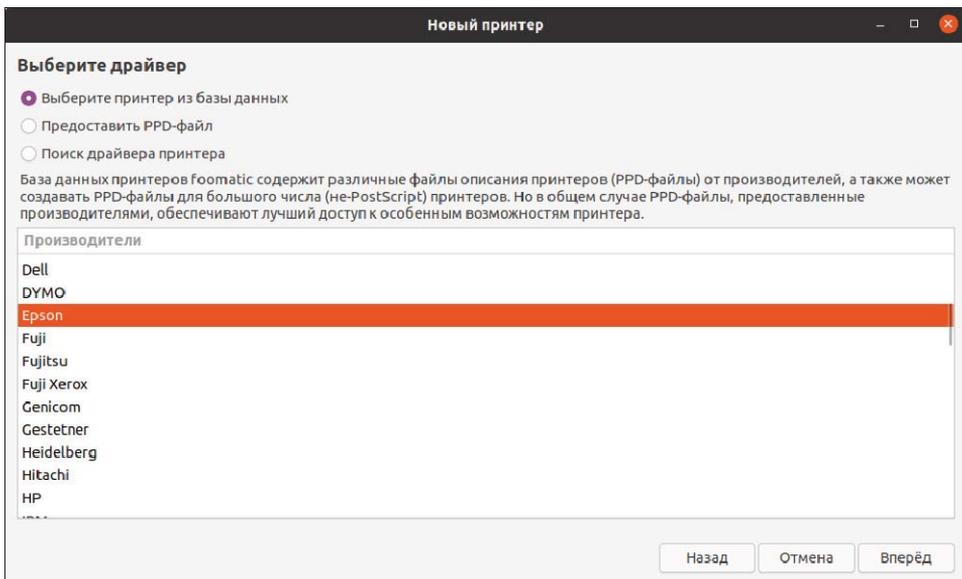


Рис. 16.5. Окно выбора драйвера для принтера

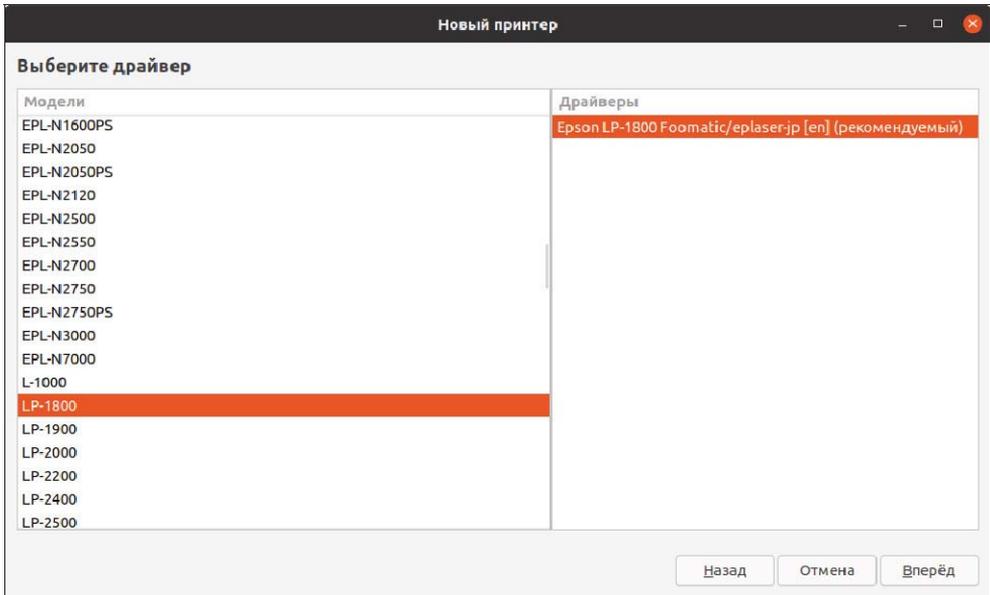


Рис. 16.6. Выбор модели принтера

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 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 **LP-1800** 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 окно **Параметры**, а затем перейти на вкладку **Принтеры**. После этого нажать кнопку

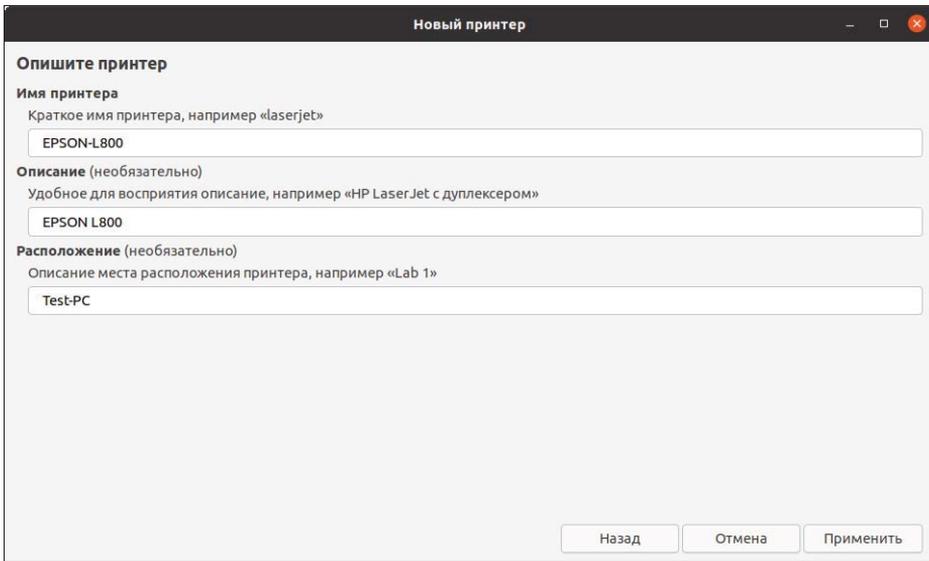


Рис. 16.7. Описание принтера

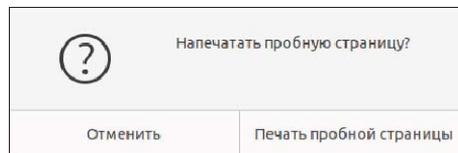


Рис. 16.8. Печать пробной страницы

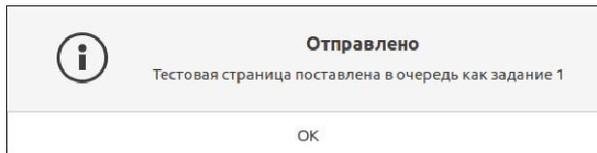


Рис. 16.9. Тестовая страница отправлена в очередь печати

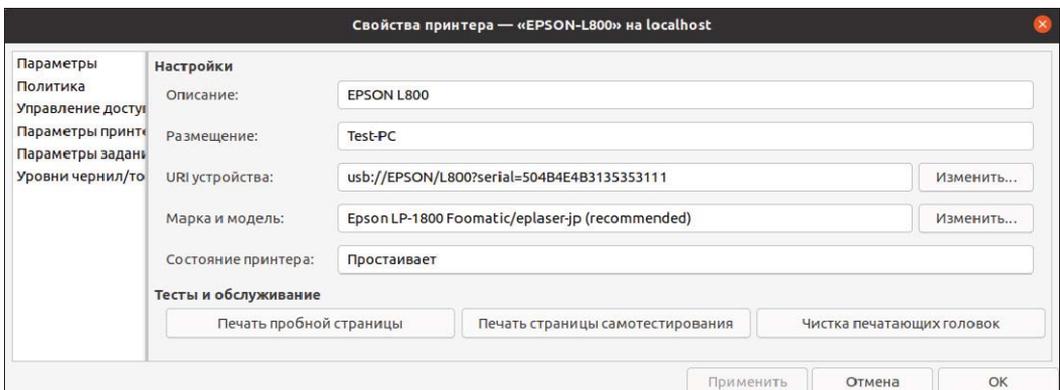


Рис. 16.10. Окно свойств локального принтера

**Дополнительные настройки принтера...**, затем в списке установленных принтеров выделить нужный принтер и в контекстном меню, вызываемом правой кнопкой мыши, выбрать пункт меню **Свойства** (рис. 16.11).

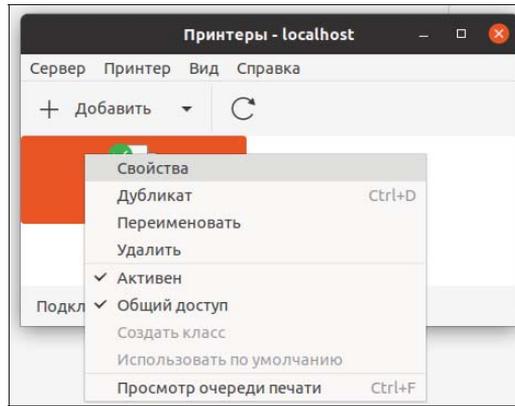


Рис. 16.11. Меню свойств установленного принтера

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 или Google Chrome. Запустите удобный для вас браузер и в адресной строке введите следующий адрес:

**http://localhost:631/**

Откроется Web-интерфейс службы CUPS. Выберите раздел **Администрирование**. Вы будете перенаправлены в раздел администрирования установленных в системе принтеров (рис. 16.12).

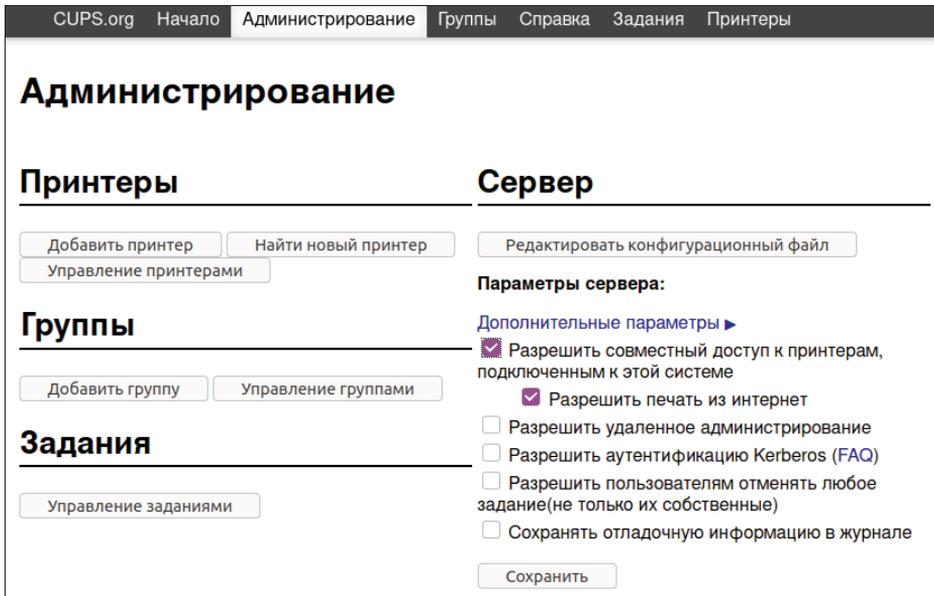


Рис. 16.12. Раздел администрирования службы CUPS

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**. Сохраните изменения, нажав кнопку **Сохранить**. Для подтверждения внесенных изменений служба CUPS потребует ввода вашего имени и пароля (рис. 16.13).

Введите имя пользователя и пароль и нажмите кнопку **ОК**. После этого служба CUPS будет перезапущена, и установленный принтер будет доступен по сети.

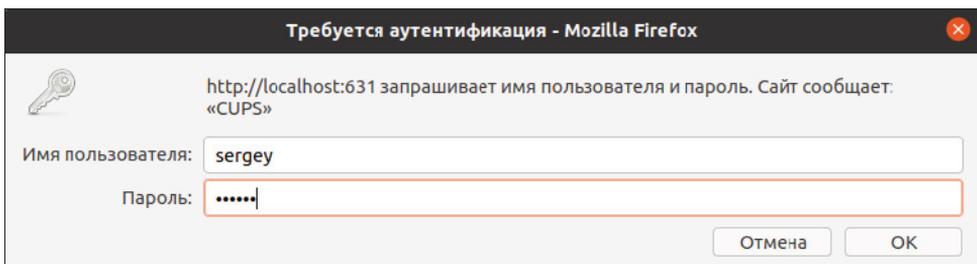


Рис. 16.13. Запрос аутентификации службы CUPS

## 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 на открывшейся странице выберите раздел **Products & Drivers**, а затем 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).

Рис. 16.14. Поиск драйверов для МФУ

Нажмите кнопку поиска, и вам будут предложены 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).

Выберите пакет `epson-inkjet-printer-escpr_1.7.7-1lsb3.2_amd64.deb` и нажмите кнопку **Загрузить**.

Название файла	Размер файла	
epson-inkjet-printer-escpr-1.7.7-1lsb3.2.i486.rpm	3.92 MB	<input type="button" value="Загрузить"/>
epson-inkjet-printer-escpr_1.7.7-1lsb3.2_i386.deb	4.41 MB	<input type="button" value="Загрузить"/>
epson-inkjet-printer-escpr-1.7.7-1lsb3.2.x86_64.rpm	3.92 MB	<input type="button" value="Загрузить"/>
<b>epson-inkjet-printer-escpr_1.7.7-1lsb3.2_amd64.deb</b>	4.41 MB	<input type="button" value="Загрузить"/>
epson-inkjet-printer-escpr-1.7.7-1lsb3.2.src.rpm	4.2 MB	<input type="button" value="Загрузить"/>
epson-inkjet-printer-escpr-1.7.7-1lsb3.2.tar.gz	4.55 MB	<input type="button" value="Загрузить"/>

Рис. 16.15. Список файлов для загрузки

**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 будут предложены 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 выберите нужную разрядность и скачайте драйвер, нажав кнопку **Download**. Будет загружен файл `iscan-bundle-2.30.4.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_1.7.7-1l1sb3.2_amd64.deb
```

**ПРИМЕЧАНИЕ**

Если при попытке установки пакета у вас появилась ошибка о том, что не удастся разрешить зависимости и отсутствует установленный пакет lsb, то выполните команду `sudo apt --fix-broken install`, которая исправит все зависимости, и после этого повторите установку пакета.

Данная команда позволит установить загруженный нами файл драйвера принтера. Теперь следует установить следующий пакет:

```
sudo dpkg -i /home/sergey/epson-printer-utility_1.1.1-1l1sb3.2_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-2.30.4.x64.deb.tar.gz
```

Так как мы разархивировали установочный пакет, нам нужно перейти в каталог, который появился после распаковки архива:

```
cd iscan-bundle-2.30.4.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** окна **Параметры** 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 окно **Параметры**, выберите пункт **Принтеры** и нажмите кнопку **Дополнительные настройки принтера...**, в котором необходимо нажать кнопку **Добавить**. Откроеется уже знакомое нам окно добавления принтера (рис. 16.16).

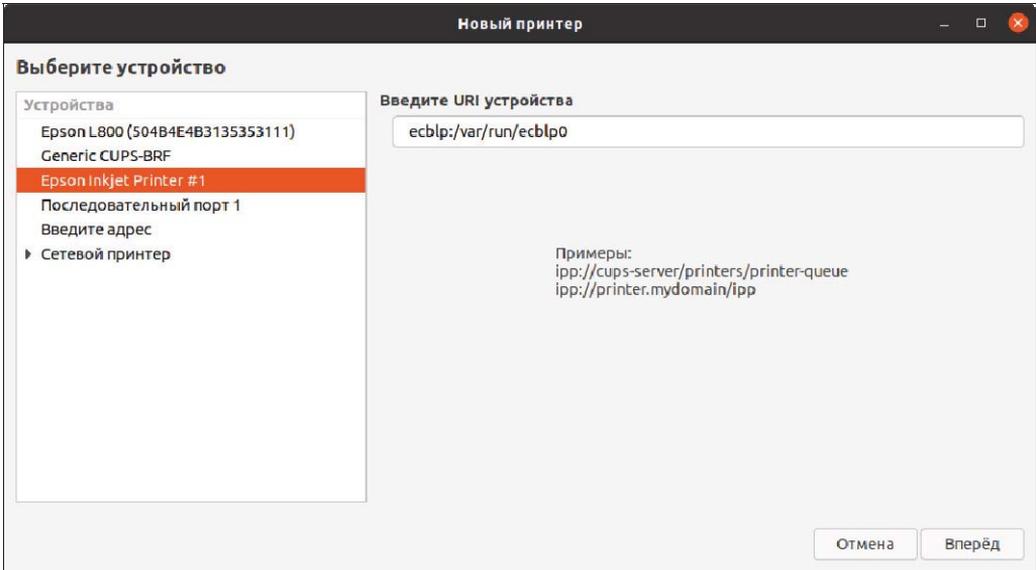


Рис. 16.16. Окно добавления принтера

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 1.7.7-11sb3.2`. 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.

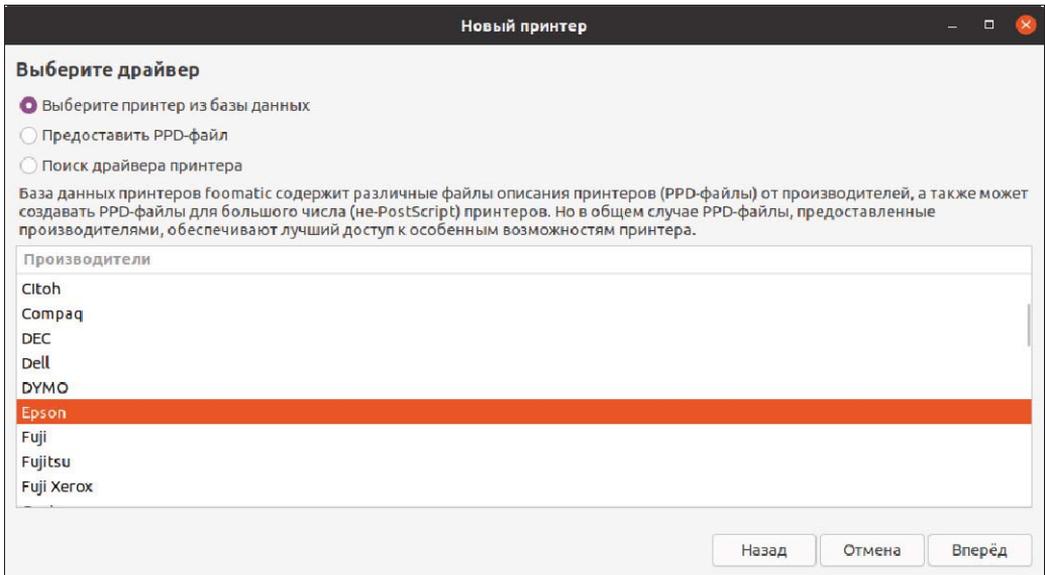


Рис. 16.17. Выбор драйвера

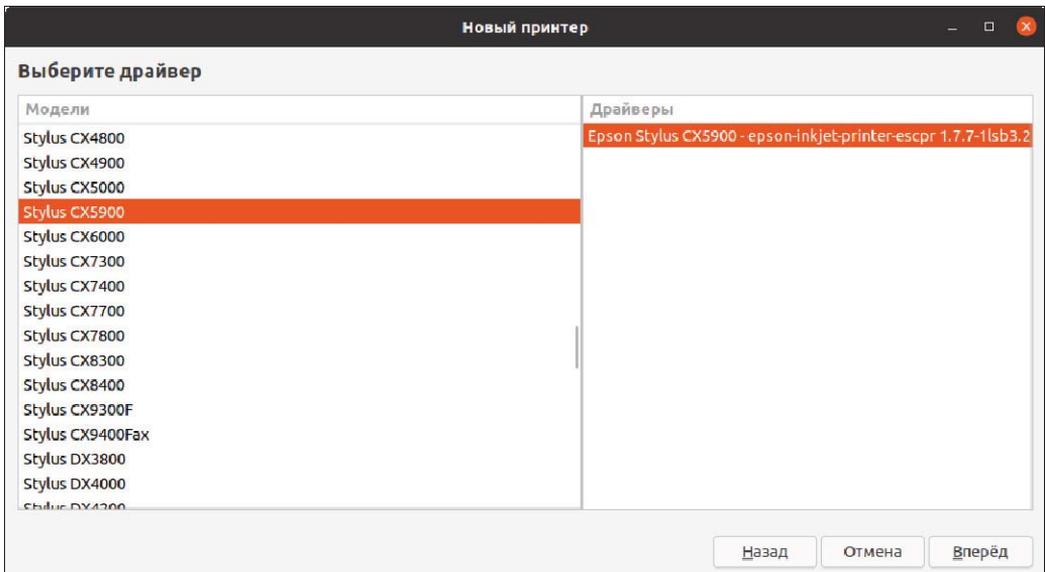


Рис. 16.18. Выбор модели МФУ

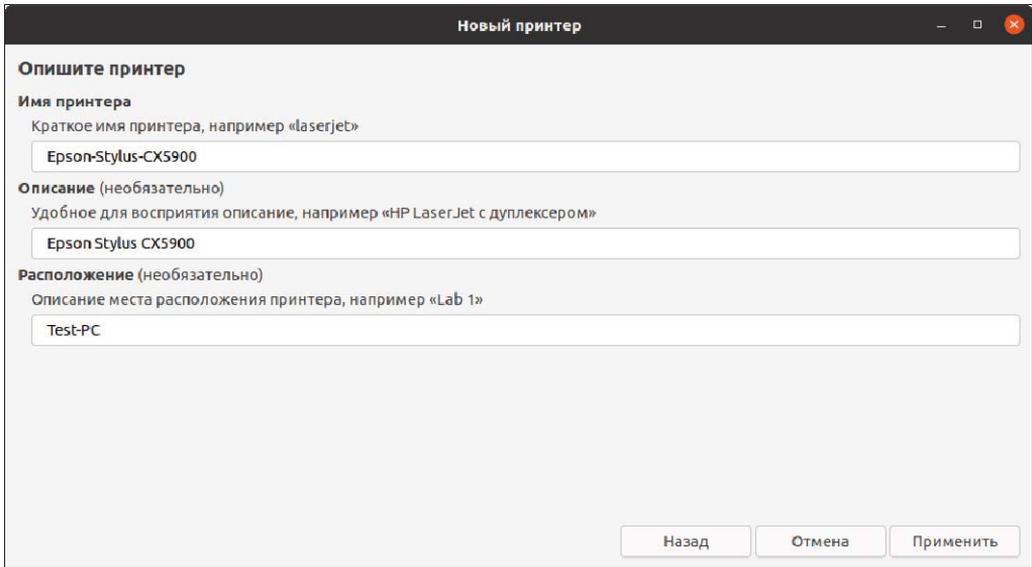


Рис. 16.19. Задание имени, расположения и описание принтера

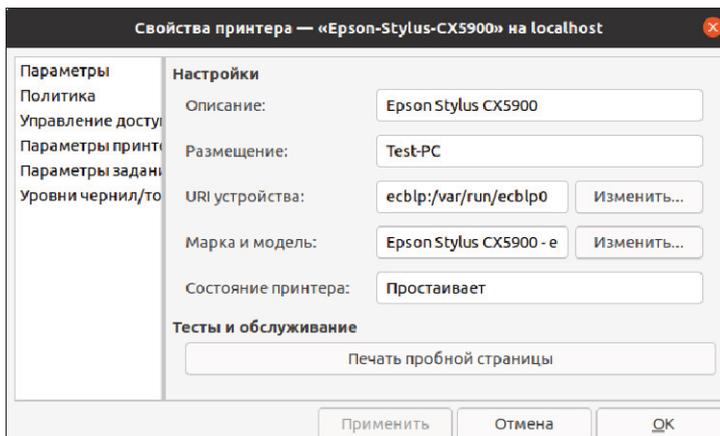


Рис. 16.20. Свойства установленного принтера

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 *csan* или *epson*.

На этом процесс установки МФУ Epson Stylus CX5900 завершен.

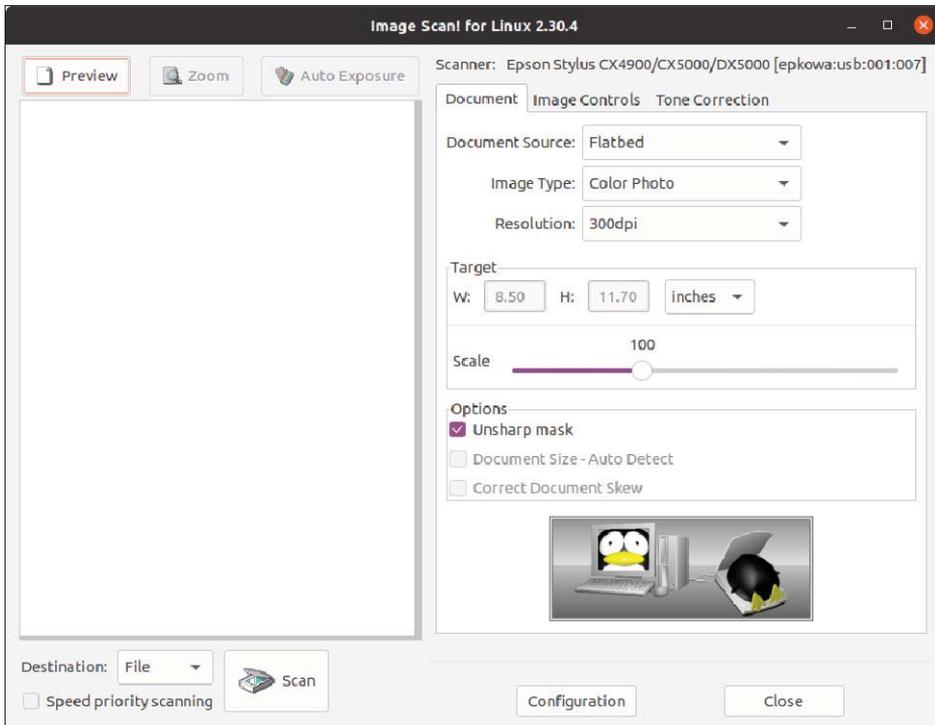


Рис. 16.21. Программа Image Scan

## Выводы

Мы научились устанавливать принтеры в ОС Ubuntu. Сам процесс установки не представляет особой сложности. 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 данной главе знаний и умений вы без труда сможете установить и настроить любой принтер в этой операционной системе.



# 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. 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 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:

- ◆ с помощью менеджера приложений Ubuntu (Ubuntu Software);
- ◆ из deb-пакетов;
- ◆ из Терминала;
- ◆ из исходных текстов.

Чуть ниже мы более подробно рассмотрим каждый вариант установки программного обеспечения в ОС Ubuntu Linux, а пока поговорим о зависимостях пакетов.

## Зависимости пакетов в ОС Ubuntu Linux

В технической документации часто упоминаются *зависимости пакетов*, однако о том, что это такое, разъяснить забывают. Попробуем разобраться.

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, а значит, разработчику приложения не нужно включать в пакет эту 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 будет отсутствовать в ОС? При установке такого пакета средствами менеджера пакетов, который умеет проверять зависимости, не случится ничего плохого — просто 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 нарушаются 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 Ubuntu Software. Далее по тексту мы будем называть эту программу *Менеджером приложений Ubuntu*.

Запустить Менеджер приложений Ubuntu можно из панели задач, щелкнув на соответствующем значке (рис. 17.1).

Откроется окно Менеджера приложений Ubuntu (рис. 17.2).



Рис. 17.1. Значок Менеджера приложений Ubuntu на панели задач

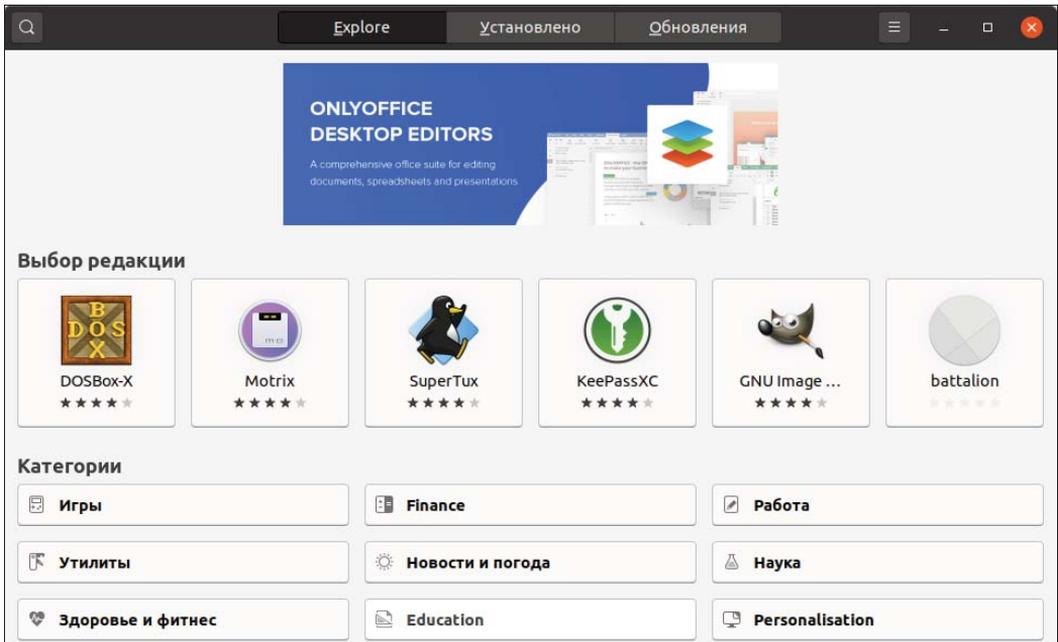


Рис. 17.2. Окно Менеджера приложений Ubuntu

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:

- ◆ **Explore** — 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. Если будут найдены обновления, то рядом с названием вкладки будет указана какая-либо цифра. Она означает, что для вашей системы доступно обновление, которое необходимо установить, нажав соответствующую кнопку на вкладке.

Менеджер приложений Ubuntu имеет простой и понятный интерфейс, и вам не составит труда найти нужное приложение.

#### 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 (кнопка которого находится в левом верхнем углу приложения) введите слово *qBittorrent*. Результат поиска отобразится в этом же окне (рис. 17.3).

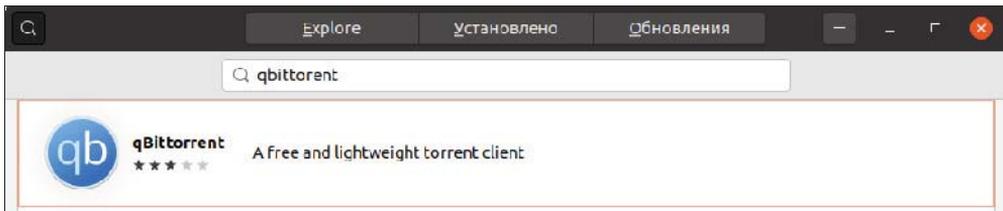


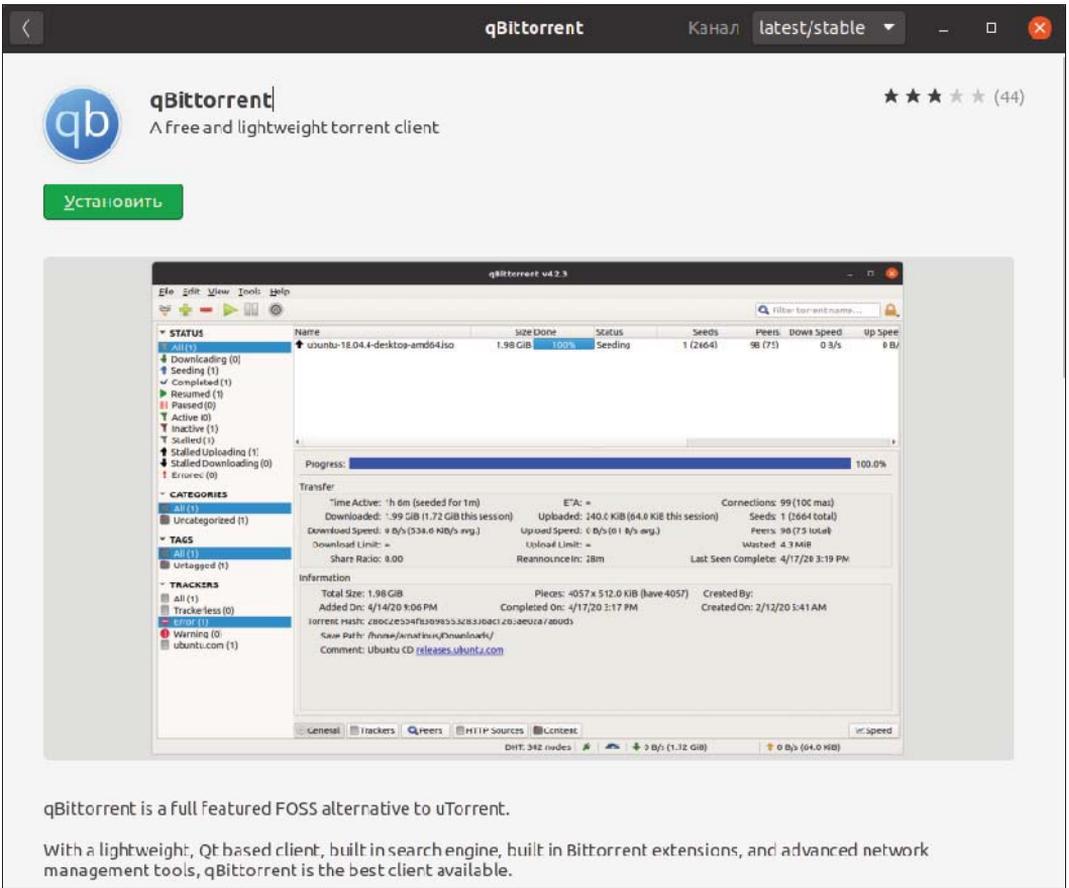
Рис. 17.3. Результат поиска программы qBittorrent

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**. Появится окно аутентификации пользователя (рис. 17.5).

Введите свой пароль и нажмите кнопку **Подтвердить**. Начнется установка программы qBittorrent, об этом процессе будет сообщать индикатор хода выполнения установки (рис. 17.6).

По окончании процесса установки программы qBittorrent индикатор процесса установки исчезнет, а вместо него в Менеджере приложений Ubuntu появятся две кнопки: **Удалить** и **Permissions**.



qBittorrent is a full featured FOSS alternative to uTorrent.

With a lightweight, Qt based client, built in search engine, built in Bittorrent extensions, and advanced network management tools, qBittorrent is the best client available.

Рис. 17.4. Страница с подробным описанием программы qBittorrent

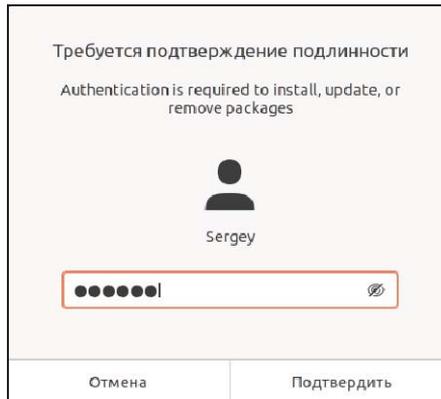


Рис. 17.5. Окно аутентификации пользователя

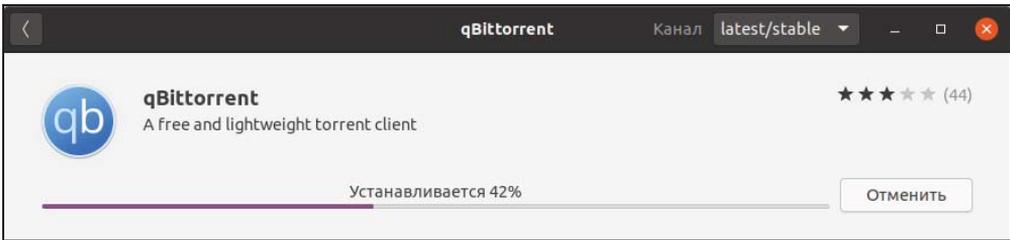


Рис. 17.6. Индикатор хода выполнения установки программы qBittorrent

Запуск установленного приложения осуществляется тем же способом, что и других приложений: из меню запуска приложений или из Терминала соответствующей командой.

## Установка приложений из deb-пакетов

Стоит иметь в виду, что не все программы включены в этот список Менеджера приложений Ubuntu. Так, популярный браузер Google Chrome отсутствует в списке, 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, что это приложение поддерживается нашей ОС. Давайте проверим, поддерживается ли браузер Google Chrome нашей операционной системой: перейдем по ссылке <https://www.google.ru/chrome/> и убедимся в наличии поддержки ОС Ubuntu Linux (рис. 17.7).

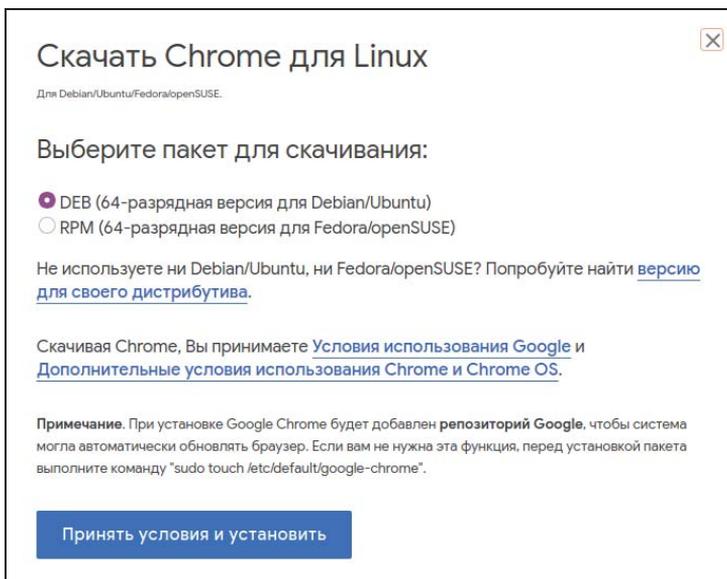


Рис. 17.7. Браузер Google Chrome доступен для ОС Ubuntu

Нам повезло, браузер поддерживается операционной системой Ubuntu Linux, поэтому мы можем смело нажимать кнопку **Принять условия и установить**, предварительно убедившись в том, что переключатель установлен напротив пункта **DEB (64-разрядная версия для Debian/Ubuntu)**. 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:

- ◆ с помощью программы установки;
- ◆ из командной строки.

Первый метод установки характеризуется своей простотой и удобством, и даже при отсутствии навыков вам не составит труда установить требуемый deb-пакет, не прибегая к запуску Терминала. Рассмотрим этот способ установки.

Откройте каталог, в который вы загрузили наш установочный пакет `google-chrome-stable_current_amd64.deb`, щелкните правой кнопкой по нему (по пакету) мыши и выберите пункт меню **Открыть в другом приложении**. Откроется окно выбора приложения для запуска данного файла, точно такое же, как мы видели при смене ассоциации файлов. В этом окне выберите пункт **Установка приложений**. Откроется окно, напоминающее окно Менеджера приложений Ubuntu (рис. 17.8).

#### ПРИМЕЧАНИЕ

В описании к программе появилось предупреждение о том, что она имеет несвободную лицензию. Именно по этой причине браузер Google Chrome отсутствует в списке приложений Менеджера приложений Ubuntu.



Рис. 17.8. Установка Google Chrome с помощью приложения Установка приложений

Для установки Google Chrome нажмите кнопку **Установить**, а затем подтвердите установку, введя свой пароль и нажав кнопку **Подтвердить**.

Процесс установки будет сопровождаться изменением индикатора прогресса в этом же окне. По окончании установки индикатор прогресса сменится кнопкой **Удалить**.

Для того чтобы запустить только что установленный браузер Google Chrome, откройте меню поиска приложений, нажав кнопку **Обзор** на панели основного меню либо клавишу <Super> (знакомая нам клавиша <Win>) и введя в строке поиска слово *chrome*. В результатах поиска отобразится установленное приложение Google Chrome. Запустите его.

Как только приложение запустится, на панели задач появится его значок. Однако есть одна особенность: как только вы закроете браузер Google Chrome, значок приложения исчезнет с панели задач. Какими будут наши следующие действия? Правильно, нужно закрепить значок приложения на панели задач. Для этого щелкните правой кнопкой мыши по значку работающего приложения и выберите пункт меню **Добавить в избранное** (рис. 17.9).

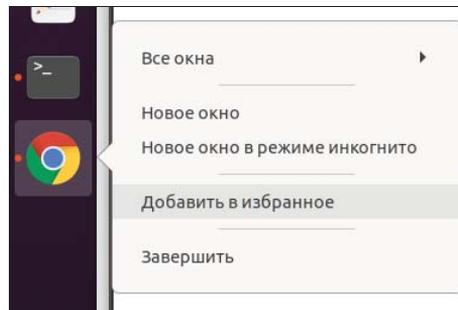


Рис. 17.9. Закрепление значка приложения Google Chrome на панели задач

Теперь, даже при неактивном приложении, значок браузера с панели задач 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 — 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`, происходит его запуск.
4. Происходит распаковка файлов устанавливаемого пакета, и выполняется 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. Стартует 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 [ключи] <команда>
```

Вкратце рассмотрим основные значения, которые принимает необязательный параметр *ключ* (табл. 17.1).

**Таблица 17.1.** Ключи команды `dpkg`

Сокращенный вариант	Полный вариант	Описание
-i	--install	Устанавливает пакет, переданный в параметре <i>команда</i>
	--instdir=<каталог>	Изменяет корневой каталог для установки приложения, при этом не меняет административный каталог
	--unpack	Распаковывает пакет без его установки и последующей настройки
	--admindir=<каталог>	Использует каталог, указанный в параметре <i>каталог</i> , вместо каталога <code>/var/lib/dpkg</code>
-A	--record-avail	Обновляет запись о доступных пакетах информацией, которая указана в устанавливаемом пакете

Таблица 17.1 (окончание)

Сокращенный вариант	Полный вариант	Описание
	<code>--configure</code>	Переконфигурирует распакованный пакет
	<code>--log=&lt;имя_файла&gt;</code>	Записывает изменения состояний и действия в файл, переданный в параметре <i>имя_файла</i>
<code>-l</code>	<code>--list</code>	Выводит список установленных в системе пакетов с указанием их версии, архитектуры и краткого описания
<code>-L</code>	<code>--listfiles &lt;имя_пакета&gt;</code>	Выводит список используемых пакетом файлов, имя которого передано в параметре <i>имя_пакета</i>
	<code>--triggers-only</code>	Запускает только послеустановочный триггер
<code>-r</code>	<code>--remove</code>	Удаляет пакет, указанный в параметре <i>команда</i>
	<code>--root=&lt;каталог&gt;</code>	Устанавливает приложение в другой корневой каталог, указанный в параметре <i>каталог</i>
<code>-p</code>	<code>--print-avail</code>	Позволяет получить информацию о доступной версии пакета, переданного в параметре <i>команда</i>
<code>-P</code>	<code>--purge</code>	Удаляет пакет вместе с его конфигурационными файлами
<code>-s</code>	<code>--status</code>	Получает статус указанного пакета, переданного в параметре <i>команда</i>
<code>-V</code>	<code>--verify</code>	Проверяет 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 команду:

```
dpkg -l
```

Следующая команда выведет список файлов, которые использует только что установленный пакет:

```
dpkg -L google-chrome-stable
```

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).

```

sergey@Test-PC: ~
sergey@Test-PC:~$ dpkg -l | grep "chrome"
ii  chrome-gnome-shell          10.1-5
    all                       GNOME Shell extensions integration for web browsers
ii  google-chrome-stable       83.0.4103.61-1
    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) — это 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`, когда занимались установкой драйверов. Теперь познакомимся с этой командой подробнее.

У вас сразу возникнет вопрос: почему мы использовали команду `apt`, а не `apt-get`, которая зачастую упоминается в технической литературе? 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 ОС. 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 при описании установки какой-либо программы в технической литературе часто используют 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. Кроме того, в технической литературе эта команда упоминается по привычке, т. к. используют ее те, кто начал свое знакомство с ОС Linux еще до появления команды `apt`. Поэтому для общего представления нужно знать о существовании обеих команд. Но имейте в виду, что вместо устаревшей команды `apt-get` все же 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 [ключи] <пакет>
```

В качестве параметра `пакет` команде `apt` передается имя устанавливаемого пакета. Необязательный параметр `ключи` может принимать значения, перечисленные в табл. 17.2.

Таблица 17.2. Ключи команды `apt`

Ключ	Описание
<code>list</code>	Выводит список пакетов из указанных имен в параметре <code>пакет</code>
<code>search</code>	Ищет информацию о пакете в описаниях пакетов

Таблица 17.2 (окончание)

Ключ	Описание
show	Отображает дополнительные данные о пакете
install	Устанавливает указанный пакет в параметре <i>пакет</i>
remove	Удаляет пакет, переданный в параметре <i>пакет</i>
purge	Удаляет не только указанный пакет, но и все его конфигурационные файлы
autoremove	Автоматически удаляет все неиспользуемые пакеты
update	Обновляет список доступных пакетов
upgrade	Обновляет операционную систему, при этом обновляются и содержащиеся в системе пакеты, но в случае возникновения конфликтов из-за зависимостей пакетов такие пакеты будут пропущены
full-upgrade	Полностью обновляет систему, при этом в случае возникновения конфликтов с зависимостью пакетов эти пакеты будут удалены, а новые версии пакетов будут установлены
edit-sources	Редактирует файл с источниками пакетов
satisfy	Выполняет установку с учетом строк зависимости

Итак, давайте применим команду `apt` и установим аудиоредактор Audacity:

```
sudo apt install audacity
```

После ввода вашего пароля начнется установка приложения. Для подтверждения установки программы Терминал запросит ваше согласие строкой "Хотите продолжить [Д/н]" и будет ожидать нажатия клавиши <Д> или <Y>. Нажмите любую из этих клавиш в зависимости от того, какая раскладка клавиатуры сейчас включена, и дождитесь окончания установки программы. Об окончании установки свидетельствует приветственная строка Терминала, готовая к вводу новых команд (рис. 17.11).

```
sergey@Test-PC: ~
Настраивается пакет libqt5svg5:amd64 (5.12.8-0ubuntu1) ...
Обрабатываются триггеры для shared-mime-info (1.15-1) ...
Обрабатываются триггеры для desktop-file-utils (0.24-1ubuntu2) ...
Обрабатываются триггеры для mime-support (3.64ubuntu1) ...
Обрабатываются триггеры для hicolor-icon-theme (0.17-2) ...
Обрабатываются триггеры для gnome-menus (3.36.0-1ubuntu1) ...
Обрабатываются триггеры для libc-bin (2.31-0ubuntu9) ...
Обрабатываются триггеры для man-db (2.9.1-1) ...
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 также потребуется подтверждение удаления программы — нажатие клавиши <Y> 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, воспользуйтесь следующей командой:

```
sudo apt purge audacity
```

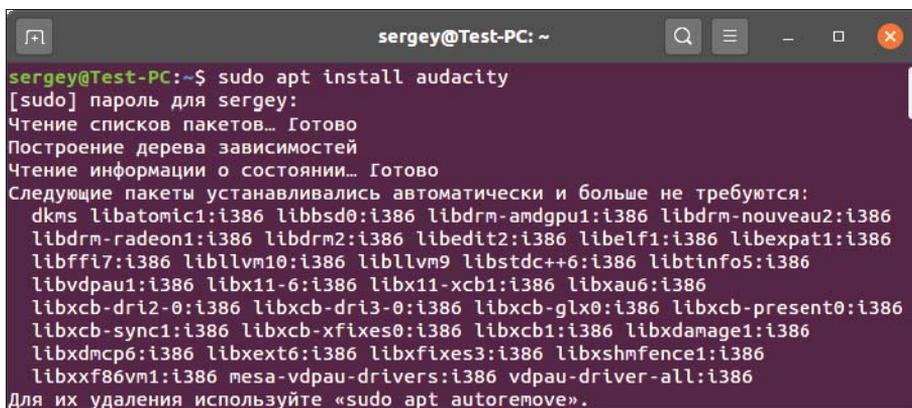
Данный способ является наиболее правильным, т. к. программа apt сама удалит все, что не нужно, и при этом сделает это как можно корректно. Если удалить файлы вручную, то велика вероятность, что вы по ошибке удалите нужные системе файлы и дальнейшая установка этого приложения будет заканчиваться ошибкой. 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).

Как видите, в нашей системе присутствует несколько неиспользуемых пакетов. Нам они ни к чему, поэтому давайте их удалим:

```
sudo apt autoremove
```



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ sudo apt install audacity  
[sudo] пароль для sergey:  
Чтение списков пакетов... Готово  
Построение дерева зависимостей  
Чтение информации о состоянии... Готово  
Следующие пакеты устанавливались автоматически и больше не требуются:  
dkms libatomic1:i386 libbsd0:i386 libdrm-amdgpu1:i386 libdrm-nouveau2:i386  
libdrm-radeon1:i386 libdrm2:i386 libedit2:i386 libelf1:i386 libexpat1:i386  
libffi7:i386 libllvm10:i386 libllvm9 libstdc++6:i386 libtinfo5:i386  
libvdpau1:i386 libx11-6:i386 libx11-xcb1:i386 libxau6:i386  
libxcb-dri2-0:i386 libxcb-dri3-0:i386 libxcb-glx0:i386 libxcb-present0:i386  
libxcb-sync1:i386 libxcb-xf86vm1:i386 libxcb1:i386 libxdamage1:i386  
libxdmcp6:i386 libxext6:i386 libxf86vm1:i386 libxshmfence1:i386  
libxxf86vm1:i386 mesa-vdpau-drivers:i386 vdpau-driver-all:i386  
Для их удаления используйте «sudo apt autoremove».
```

Рис. 17.12. Программа apt нашла неиспользуемые пакеты

После подтверждения удаления этих пакетов программа apt поочередно удалит каждый из этих неиспользуемых 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_2.7.4.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_2.7.4.tar.gz
```

В результате выполнения команды в вашем домашнем каталоге будет создан каталог `~/avidemux_2.7.4/`, 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 прочитаем этот файл. Возможно, в нем есть инструкции по установке программы (рис. 17.14).

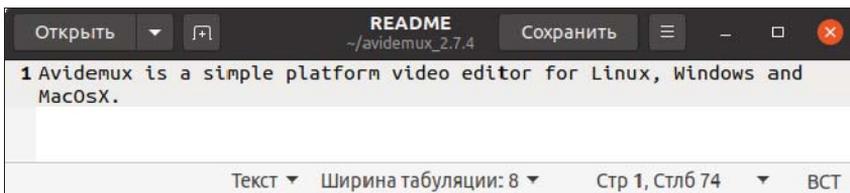


Рис. 17.14. Содержимое файла `ReadMe`

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_2.7.4
```

Теперь, когда мы находимся в каталоге `/home/sergey/avidemux_2.7.4/`, 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 install gcc g++ make cmake pkg-config libpng-dev fakeroot yasm  
libsqlite3-dev build-essential
```

#### **ПРИМЕЧАНИЕ**

Как вы могли заметить, в разделе **Requirement list (debian/ubuntu)** описывается применение команды `apt-get`, но мы-то ведь знаем, что эта команда устаревшая и вместо нее нужно использовать команду `apt`. Кроме того, пакет `libpng12-dev` был переименован в `libpng-dev`. Поэтому наши команды немного отличаются от команд, которые указаны на сайте разработчика программы.

Установка данных модулей не займет много времени. От вас только потребуется подтвердить установку и дождаться ее окончания. Если сейчас снова попытаться запустить предыдущую команду, 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 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, который занимает немного времени, но не так быстр, как установка необходимых компонентов для компиляции этого приложения. Об успешном процессе компиляции свидетельствует строка `*** ALL DONE ***` (рис. 17.15).

```

sergey@Test-PC: ~/avidemux_2.7.4
4-200529-plugins.deb
-rw-rw-r-- 1 sergey sergey 254610 мая 29 11:41 avidemux3-plugins-qt5-2.7.4-2
00529-plugins.deb
-rw-rw-r-- 1 sergey sergey 6548 мая 29 11:41 avidemux3-qt5-2.7.4-200529-de
v.deb
-rw-rw-r-- 1 sergey sergey 3970228 мая 29 11:41 avidemux3-qt5-2.7.4-200529-ru
ntime.deb
-rw-rw-r-- 1 sergey sergey 58726 мая 29 11:41 avidemux3-settings-2.7.4-2005
29-settings.deb
** ALL DONE **
** The installable packages are in the debs folder **
sergey@Test-PC:~/avidemux_2.7.4$

```

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).

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_2.7.4/install/usr/` в корневой каталог `/usr/`:

```
sudo cp -r /home/sergey/avidemux_2.7.4/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.

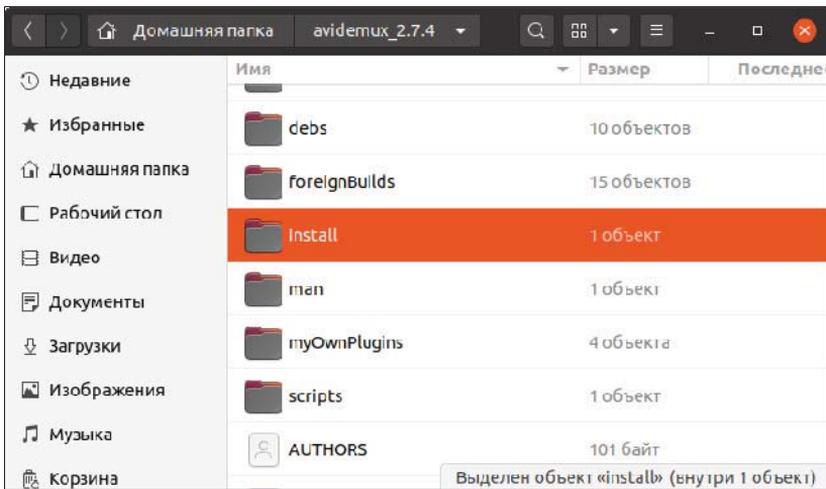


Рис. 17.16. Каталог `install` находится в каталоге с исходными текстами программы

Как мы помним, если ввести часть слова в Терминале и три раза нажать клавишу `<Tab>`, Терминал подскажет нам полное название команд, которые соответствуют заданному слову (рис. 17.17).

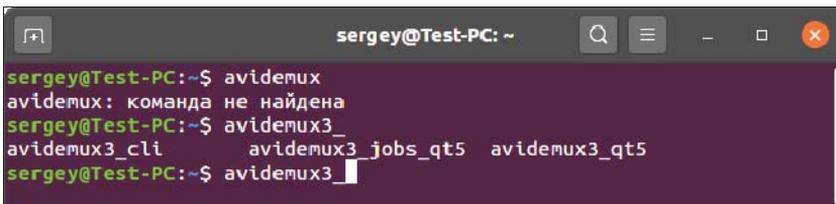


Рис. 17.17. Терминал вывел список команд, которые начинаются на слово `avidemux`

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, вы несколько раз на страницах книги встречали слово "репозиторий", но до сих пор эта тема обходилась 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 всего изложения мы несколько раз упоминали репозитории, 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 конфигурировать систему 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 меню поиска приложений по запросу "программы и обновления" (рис. 17.18).

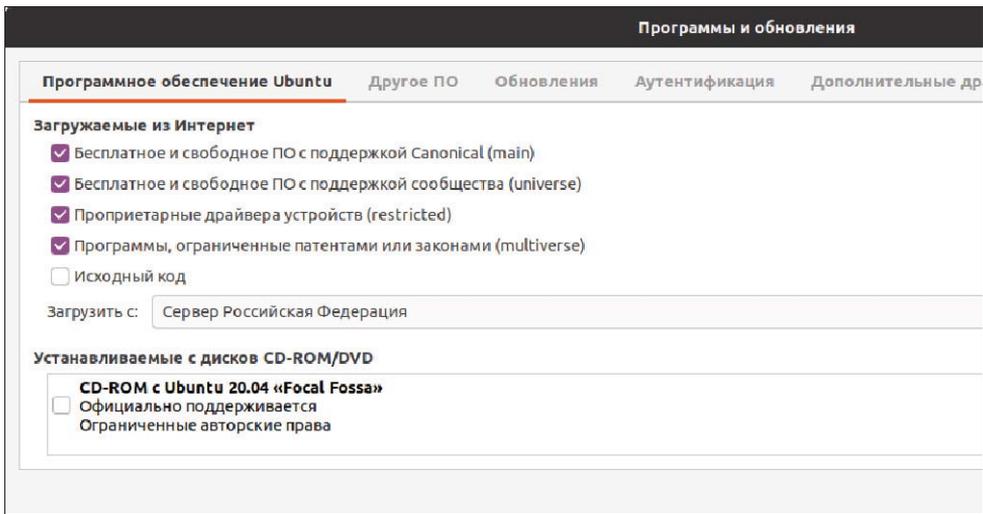


Рис. 17.18. Окно настроек репозитория

На первой вкладке окна **Программное обеспечение Ubuntu** вы можете управлять ветками официальных репозиториях (*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 ней отображается весь список репозиториях Ubuntu, которые подключены к системе.

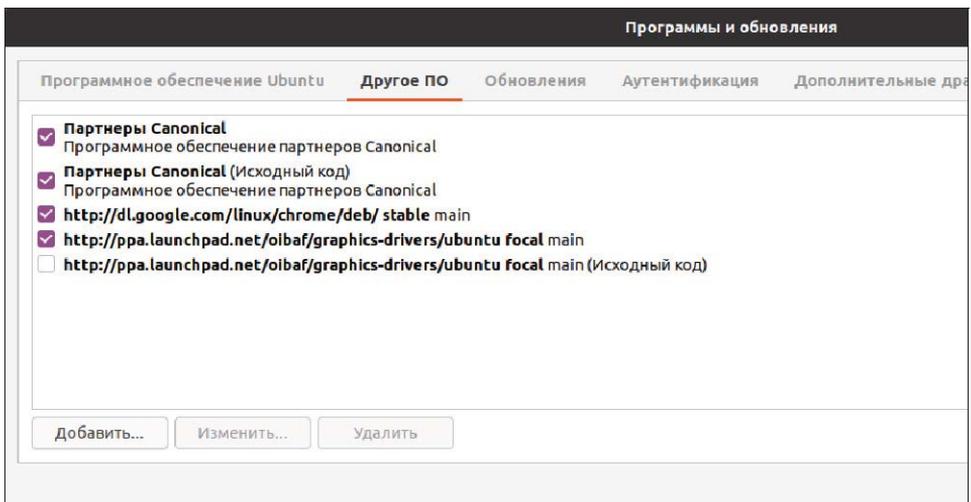


Рис. 17.19. Вкладка со списком репозиториях

Это означает, что, помимо предустановленных официальных репозиториев, 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.

### ПРИМЕЧАНИЕ

Описанный ниже способ добавления источника репозитория для большинства программ уже не является рекомендуемым по той причине, что на смену ему пришла система управления Snap-пакетами, которую мы рассмотрим чуть позже.

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 APT** введите строку `ppa:notepadqq-team/notepadqq` (рис. 17.20).

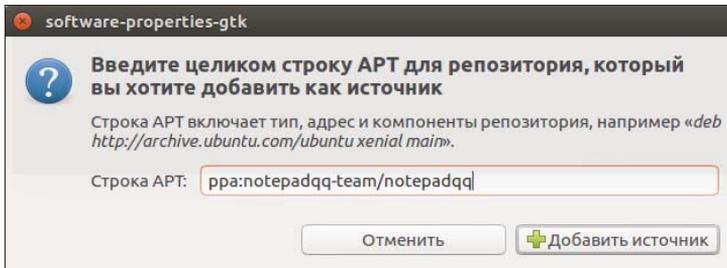


Рис. 17.20. Добавление нового репозитория

После этого нажмите кнопку **Добавить источник** и подтвердите добавление источника, введя свой пароль. Среди списка репозиториев у вас появятся два репозитория персонального архива пакетов (PPA) (рис. 17.21).

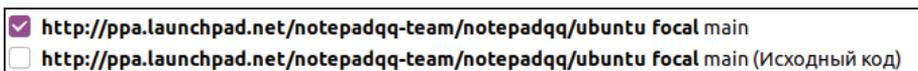


Рис. 17.21. Два новых источника репозитория

Нажмите кнопку **Закреть**. Появится информационное окно, сообщающее о том, что список источников программ и их обновлений изменился и требуется его обновить (рис. 17.22).

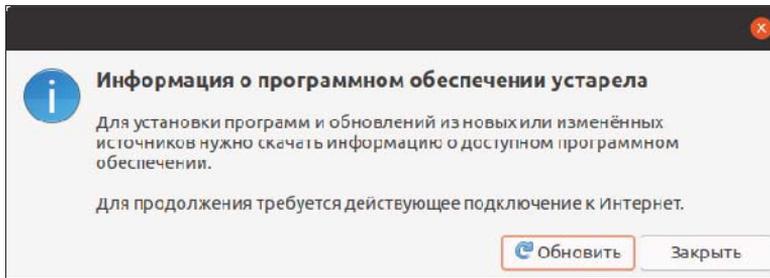


Рис. 17.22. Обновление списка PPA

Нажмите кнопку **Обновить** для того, чтобы обновился список данных программного обеспечения.

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 репозиториев. Не забывайте удалять ненужные вам репозитории, т. к. наличие в системе неиспользуемых репозиториев — плохой тон.

## Snap-пакеты в Ubuntu

Snap — это система развертывания (установки) и управления программными пакетами, применяемая в ОС на базе Debian.

Прежде всего, система управления Snap-пакетами предназначена для корректной работы программ в разных версиях операционной системы. Как она работает?

Представим ситуацию, когда вы устанавливаете программу Notepadqq и через некоторое время выходят обновления ОС, вы их загружаете и устанавливаете. Затем пытаетесь запустить Notepadqq и с удивлением обнаруживаете, что ваша любимая программа отказывается запускаться. Что же произошло с программой и по какой причине она отказывается запускаться?

Дело в том, что при установке обновлений ОС обновились также и системные библиотеки. Однако программа Notepadqq до этого работала на прежней версии системных библиотек и по-прежнему ищет для своей работы старую версию системных библиотек. По этой причине многие программы отказываются запускаться после обновления самой операционной системы.

Наверняка у вас возникнет идея установить свежую версию нужной вам программы. Да, иногда это помогает, но не всегда. Почему? Дело в том, что зачастую разработчики программ просто-напросто не успевают выпускать обновления своих программ под текущую версию операционной системы. И до появления Snap-пакетов выход был один: ждать того момента, пока разработчик программы не адаптирует ее под свежую ОС. Согласитесь, ведь это не очень продуктивное занятие и к тому же может потребовать довольно длительного времени.

С появлением системы Snap-пакетов проблема разрешилась: разработчик программы включает в пакет с программой весь необходимый набор с нужной ему версией и при установке данного пакета создается специальное программное окружение, в котором приложение хранит все свои зависимости (системные библиотеки конкретной версии).

Таким образом, если вы обновите операционную систему, то приложение будет работать, т. к. прежняя версия системной библиотеки не перезаписалась, а осталась неизменной в программной среде самого приложения. Разумеется, вы вправе установить новую версию программы тогда, когда она появится, а до тех пор работать с прежней версией и не терять время на ожидание обновлений вашей программы.

Устанавливается система управления Snap-пакетами следующей командой:

```
sudo apt install snap
```

Теперь давайте установим программу Atom с помощью системы управления Snap-пакетами:

```
sudo snap install atom --classic
```

Данная команда позволяет быстро и легко установить нужную нам программу, которая точно будет работать даже после обновления вашей операционной системы.

Для того чтобы произвести удаление Snap-пакета, выполните следующую команду:

```
sudo snap install atom --classic
```

## Удаление программ

За все время изучения ОС Ubuntu Linux мы установили большое количество программ, однако до сих пор не затрагивали тему удаления установленных программ.

Существуют два способа удаления программ:

- ◆ в графическом режиме;
- ◆ 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).

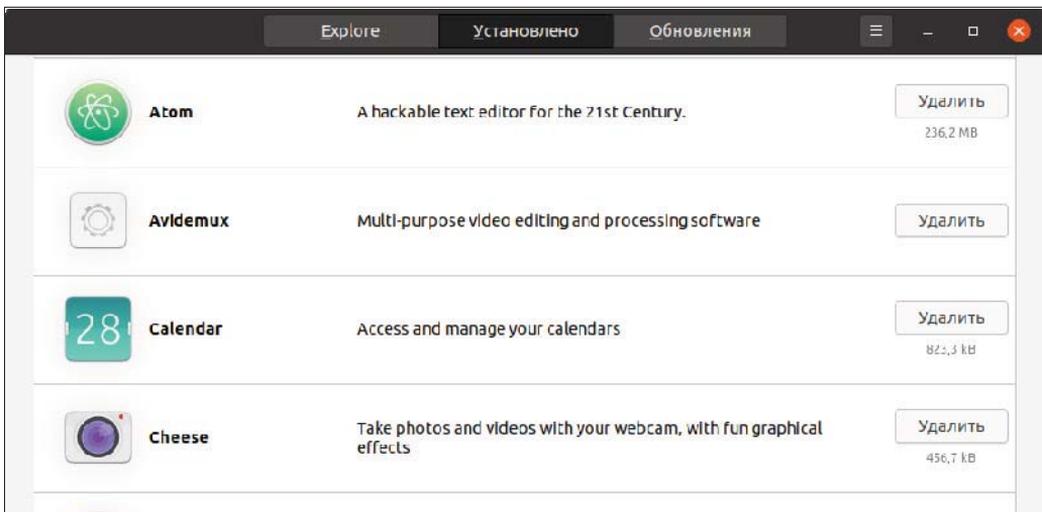


Рис. 17.23. Список установленных приложений в ОС Ubuntu

### ПРИМЕЧАНИЕ

Обратите внимание на то, что вновь устанавливаемые приложения отображаются в этом списке в том порядке, в котором мы их устанавливали.

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.

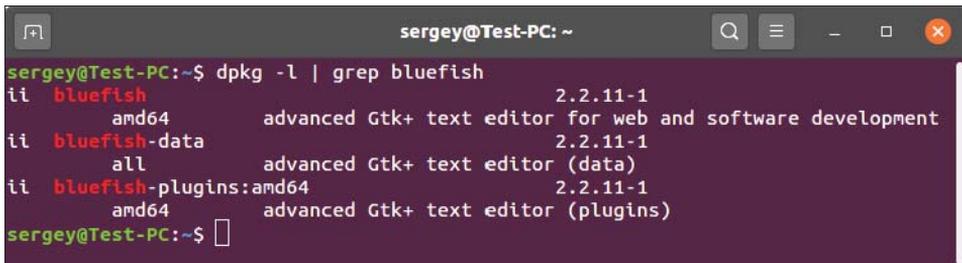
Итак, для начала нам нужно узнать точное название удаляемой программы. Для этого получим список установленных приложений:

```
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 (программа Bluefish была установлена предварительно):

```
dpkg -l | grep bluefish
```

Теперь запрос получится более конкретизированным (рис. 17.24).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ dpkg -l | grep bluefish  
ii bluefish 2.2.11-1  
    amd64 advanced Gtk+ text editor for web and software development  
ii bluefish-data 2.2.11-1  
    all advanced Gtk+ text editor (data)  
ii bluefish-plugins:amd64 2.2.11-1  
    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 были установлены три пакета: bluefish, bluefish-data и bluefish-plugins:amd64. Их нам придется удалять по очереди, поэтому гораздо удобнее будет выполнить объединенную команду удаления пакетов:

```
sudo apt remove bluefish bluefish-data bluefish-plugins:amd64
```

После выполнения данной команды все три указанных пакета будут удалены. Для того чтобы убедиться в этом, 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. Эти умения нам пригодятся на протяжении всей работы с нашей операционной системой, т. к. установка и удаление программ являются основой работы в любой ОС.

Мы познакомились с репозиториями, научились их добавлять как в графическом режиме, так и в режиме командной строки. Мы узнали, что такое зависимости при установке и удалении программ, а также научились работать со Snap-пакетами.

## ГЛАВА 18



# Установка дополнительного программного обеспечения

## Дополнительное программное обеспечение 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 ОС. 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, 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 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 следует установить. Вам остается только сделать пару щелчков мышью, и необходимый набор системных библиотек будет установлен.

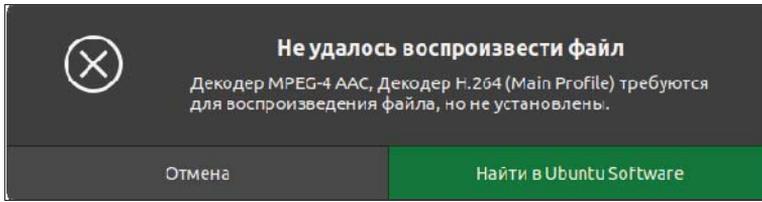


Рис. 18.1. Видеопроеигрыватель предлагает установить дополнительные медиамодули

В качестве примера запустите любой видеофайл с расширением avi или mp4, и медиаплеер предложит вам установить дополнительные медиамодули (рис. 18.1).

Нажмите кнопку **Найти в Ubuntu Software**, и вам будет предложено установить те модули, которые потребуются для корректного воспроизведения открытого вами типа файла (рис. 18.2).



Рис. 18.2. Требуется установка дополнительных модулей для воспроизведения

Нажмите кнопку **Установить** и для подтверждения установки введите пароль вашей учетной записи. Начнется процесс установки требуемых пакетов (рис. 18.3).

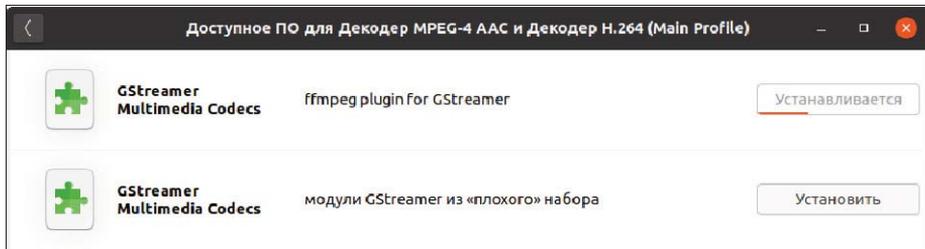


Рис. 18.3. Процесс установки отмеченных пакетов

Для просмотра более детальной информации по нужному модулю можно получить, нажав на самом модуле. Откроется окно с подробным описанием модуля и хода установки (рис. 18.4).

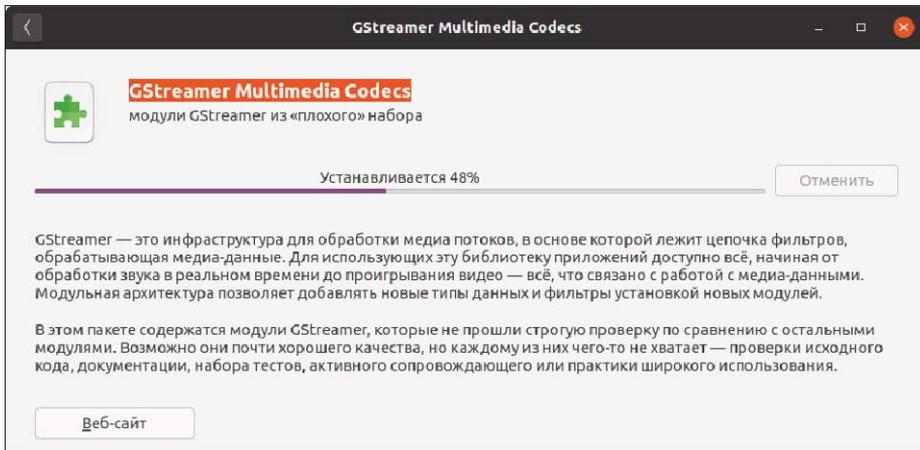


Рис. 18.4. Подробности мультимедийного модуля

По окончании процесса установки необходимых модулей 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 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. Аудиопроеигрыватель 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).

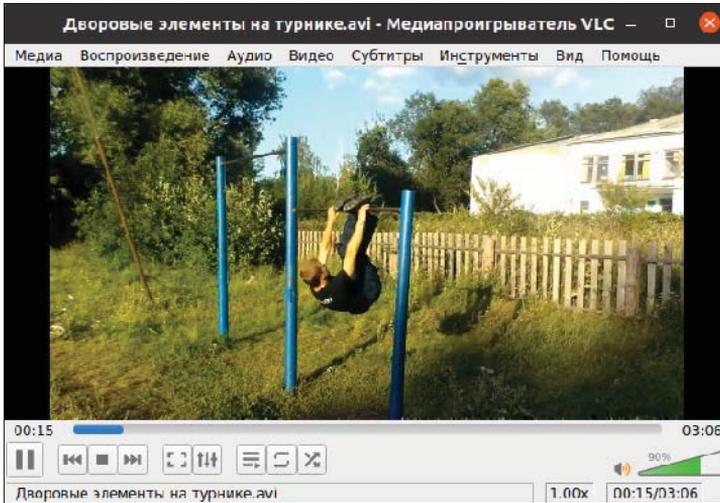


Рис. 18.6. Видеоплеер VLC

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 две программы, 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 в предыдущей главе. Второй вариант больше подойдет тем, кто стремится поль-

зваться 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. Это умеет делать BitTorrent-клиент под названием qBittorrent (рис. 18.7).

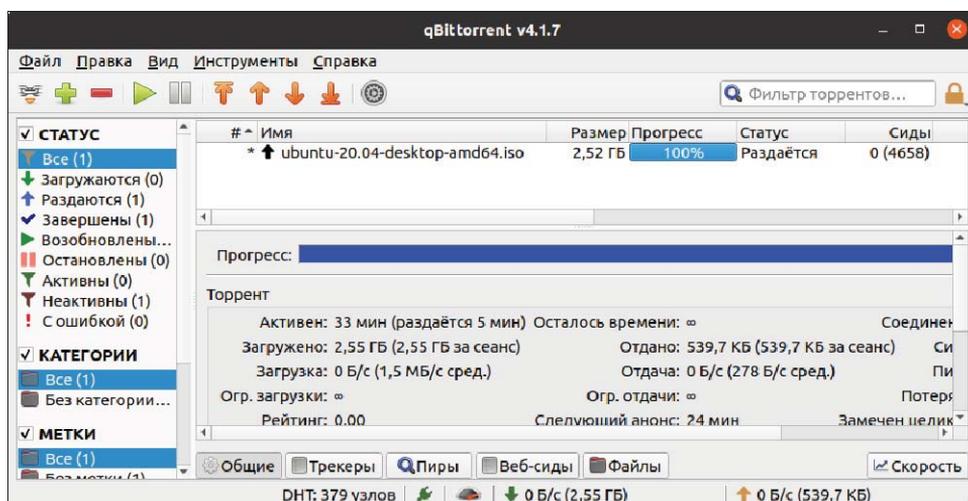


Рис. 18.7. BitTorrent-клиент qBittorrent

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 появились 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.

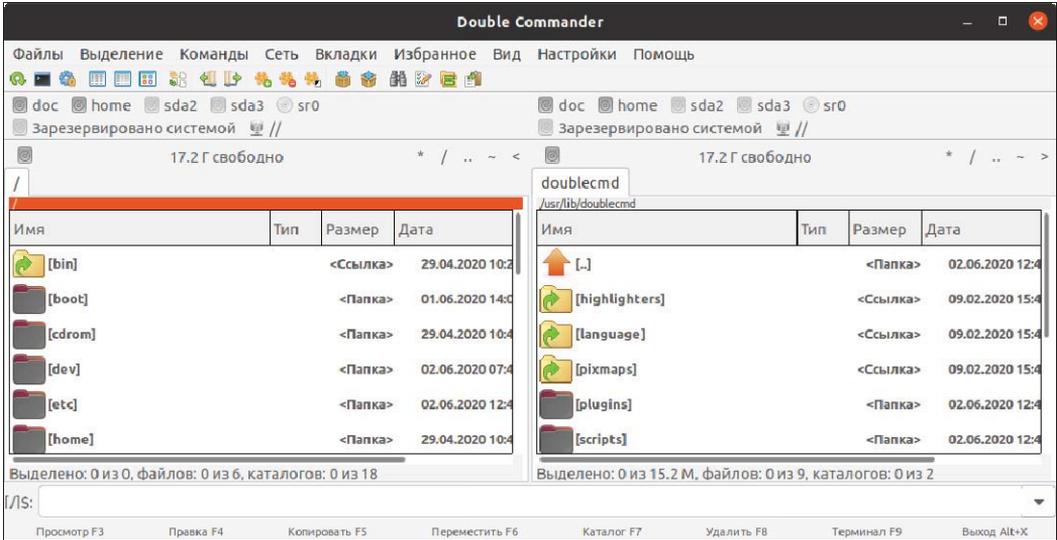


Рис. 18.8. Файловый менеджер Double Commander

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 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 функ-

циональной 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 приложениями, 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, тем не менее это не гарантирует работу того или иного Windows-приложения под ОС Ubuntu Linux. Все проблемы совместимости связаны с закрытостью 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 функционирование большинства популярных Windows-приложений, и разработчики Wine активно работают в этом направлении.

## Установка Wine

В стандартных репозиториях ОС Ubuntu уже имеется доступная для вашего дистрибутива версия Wine. В ранних дистрибутивах ОС Ubuntu для установки Wine из стандартного репозитория предназначалась следующая команда:

```
sudo apt install wine
```

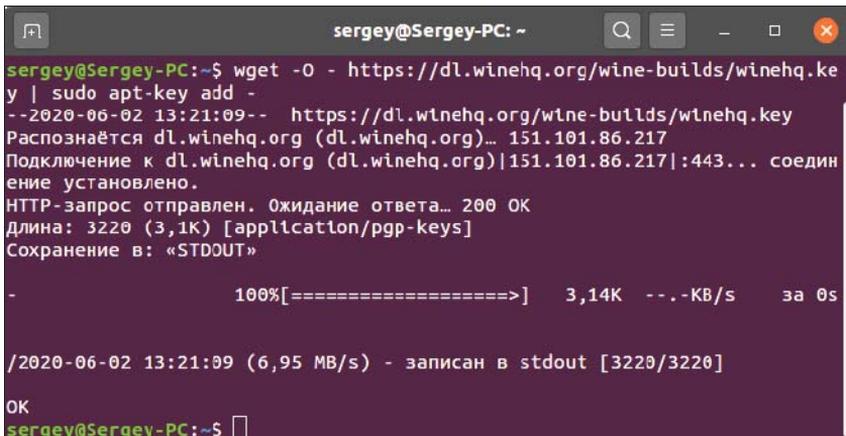
Однако для установки актуальной версии Wine лучше всего следовать инструкции разработчиков, расположенной на странице <https://wiki.winehq.org/ubuntu>.

Для начала рекомендуется включить поддержку 32-битной архитектуры:

```
sudo dpkg --add-architecture i386
```

После этого необходимо скачать и добавить официальный ключ репозитория Wine (рис. 19.1):

```
wget -O - https://dl.winehq.org/wine-builds/winehq.key | sudo apt-key add -
```

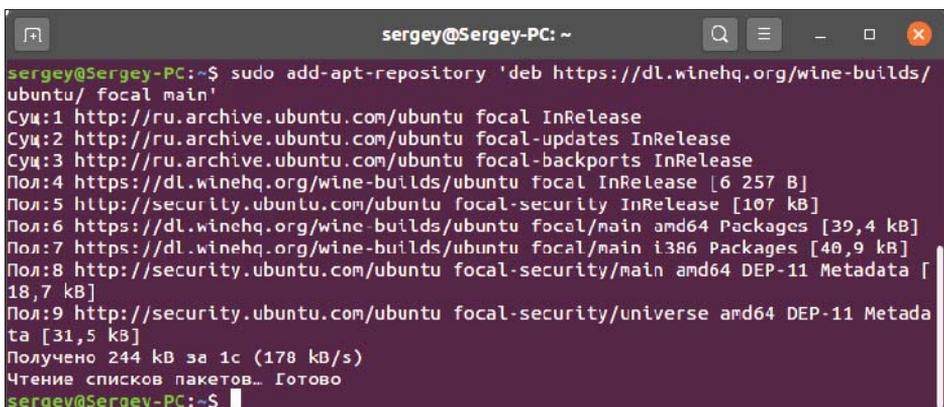


```
sergey@Sergey-PC: ~
sergey@Sergey-PC:~$ wget -O - https://dl.winehq.org/wine-builds/winehq.ke
y | sudo apt-key add -
--2020-06-02 13:21:09-- https://dl.winehq.org/wine-builds/winehq.key
Распознаётся dl.winehq.org (dl.winehq.org)... 151.101.86.217
Подключение к dl.winehq.org (dl.winehq.org)[151.101.86.217]:443... соедин
ение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
длина: 3220 (3,1К) [application/pgp-keys]
Сохранение в: «STDOUT»
-          100%[=====] 3,14К --.-KB/s за 0с
/2020-06-02 13:21:09 (6,95 MB/s) - записан в stdout [3220/3220]
OK
sergey@Sergey-PC:~$
```

Рис. 19.1. Добавления ключа репозитория Wine

Теперь мы можем добавить репозиторий для нашей версии Ubuntu (рис. 19.2):

```
sudo add-apt-repository 'deb https://dl.winehq.org/wine-builds/ubuntu/
focal main'
```



```
sergey@Sergey-PC: ~
sergey@Sergey-PC:~$ sudo add-apt-repository 'deb https://dl.winehq.org/wine-builds/
ubuntu/ focal main'
Суц:1 http://ru.archive.ubuntu.com/ubuntu focal InRelease
Суц:2 http://ru.archive.ubuntu.com/ubuntu focal-updates InRelease
Суц:3 http://ru.archive.ubuntu.com/ubuntu focal-backports InRelease
Пол:4 https://dl.winehq.org/wine-builds/ubuntu focal InRelease [6 257 B]
Пол:5 http://security.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Пол:6 https://dl.winehq.org/wine-builds/ubuntu focal/main amd64 Packages [39,4 kB]
Пол:7 https://dl.winehq.org/wine-builds/ubuntu focal/main i386 Packages [40,9 kB]
Пол:8 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [
18,7 kB]
Пол:9 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metada
ta [31,5 kB]
Получено 244 kB за 1с (178 kB/s)
Чтение списков пакетов... Готово
sergey@Sergey-PC:~$
```

Рис. 19.2. Репозиторий Wine был успешно добавлен

После добавления нового репозитория нам необходимо обновить список пакетов:

```
sudo apt update
```

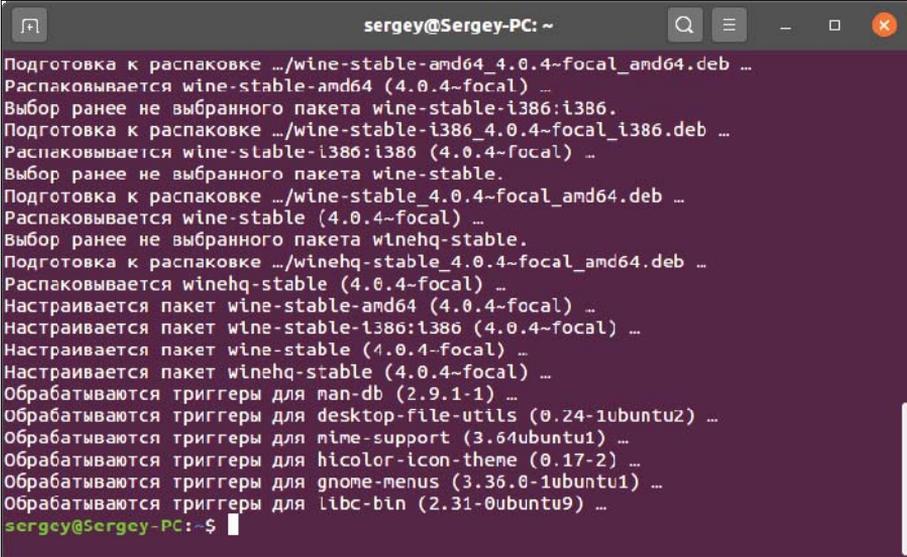
Эта команда обновит список пакетов и перестроит список зависимостей, которые необходимы для установки актуальной версии Wine.

Подготовительные действия закончены, осталось только установить Wine, выполнив следующую команду:

```
sudo apt install --install-recommends winehq-stable
```

После ввода пароля пользователя и подтверждения установки запустится процесс установки. Это займет некоторое время, все зависит от пропускной способности вашего интернет-соединения.

О завершении процедуры установки свидетельствует окно приветствия, готовое к вводу команд (рис. 19.3).



```
sergey@Sergey-PC: ~
Подготовка к распаковке .../wine-stable-amd64_4.0.4~focal_amd64.deb ...
Распаковывается wine-stable-amd64 (4.0.4~focal) ...
Выбор ранее не выбранного пакета wine-stable-i386:i386.
Подготовка к распаковке .../wine-stable-i386_4.0.4~focal_i386.deb ...
Распаковывается wine-stable-i386:i386 (4.0.4~focal) ...
Выбор ранее не выбранного пакета wine-stable.
Подготовка к распаковке .../wine-stable_4.0.4~focal_amd64.deb ...
Распаковывается wine-stable (4.0.4~focal) ...
Выбор ранее не выбранного пакета winehq-stable.
Подготовка к распаковке .../winehq-stable_4.0.4~focal_amd64.deb ...
Распаковывается winehq-stable (4.0.4~focal) ...
Настраивается пакет wine-stable-amd64 (4.0.4~focal) ...
Настраивается пакет wine-stable-i386:i386 (4.0.4~focal) ...
Настраивается пакет wine-stable (4.0.4~focal) ...
Настраивается пакет winehq-stable (4.0.4~focal) ...
Обрабатываются триггеры для man-db (2.9.1-1) ...
Обрабатываются триггеры для desktop-file-utils (0.24-1ubuntu2) ...
Обрабатываются триггеры для mime-support (3.64ubuntu1) ...
Обрабатываются триггеры для hicolor-icon-theme (0.17-2) ...
Обрабатываются триггеры для gnome-menus (3.36.0-1ubuntu1) ...
Обрабатываются триггеры для libc-bin (2.31-0ubuntu9) ...
sergey@Sergey-PC:~$
```

Рис. 19.3. Установка Wine завершена

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**.

При первом запуске Wine программа запросит установку Wine Mono, а затем пакета Wine Gecko. Установка этих пакетов потребуется для нормального функционирования программ, использующих фреймворк .NET, а также для корректной работы браузеров. Нажмите кнопку **Установить** для установки этих дополнений. 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.

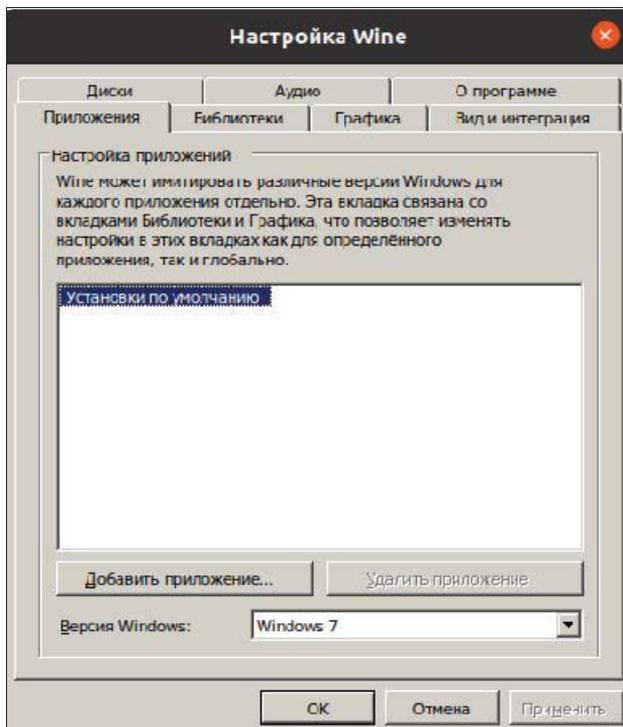


Рис. 19.4. Окно настроек Wine

В окне настроек Wine можно дополнительно сконфигурировать 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.

## Установка и запуск Windows-приложений

Вас, вероятно, 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.

**2ГИС — всегда под рукой**

Приложения и компоненты

Версии 2ГИС для компьютеров, смартфонов и планшетов.

[Версия для ПК](#) [Мобильная версия](#)

[Windows](#) [Linux](#) [Модули](#)

↓ [Оболочка 2ГИС](#)  
Версия 3.16.3.0, .orig.zip, 22.60 MB

↓ [База данных Омска](#)  
Май 2020, .orig.zip, 61.45 MB

**i** Для работы 2ГИС под Linux потребуются установленный Wine версии не ниже 1.5. Скачайте архивы оболочки и базы данных справочника, распакуйте загруженные пакеты в домашнюю директорию. В случае некорректного отображения шрифтов в программе необходимо будет добавить стандартные Windows-шрифты в каталог Wine: «~/wine/drive\_c/windows/fonts».

Рис. 19.5. Раздел загрузки карт 2ГИС

**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.

Теперь необходимо распаковать загруженный файл. Для того 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.

```
sergey@Test-PC: ~
sergey@Test-PC:~$ ls
2GISShell-3.16.3.0.orig.zip
alsa-firmware-1.2.1
alsa-firmware-1.2.1.tar.bz2
alsa-lib-1.2.2
alsa-lib-1.2.2.tar.bz2
alsa-plugins-1.2.2
alsa-plugins-1.2.2.tar.bz2
alsa-utils-1.2.2
```

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/gym.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

Начнется загрузка, а затем установка выбранных вами пакетов (рис. 19.8).

Дождавшись окончания процесса, перезапустите программу 2ГИС.

На этом установка 2ГИС будет завершена.

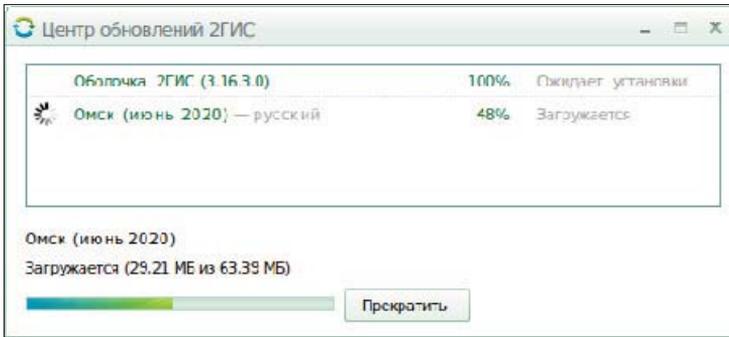


Рис. 19.8. Загрузка выбранных пакетов

Для запуска приложения 2ГИС из Терминала используйте следующую команду:

```
wine ~/2gis/3.0/gym.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 `gym.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 проводить установку Wine, которая является некой программной платформой совместимости с Windows API, что позволяет запускать приложения, предназначенные для работы в ОС Windows.

Мы также научились запускать установленные Windows-приложения 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. Они 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).

Данная программа позволяет работать с различными типами архивов в ОС Ubuntu Linux. С помощью Менеджера архивов вы сможете создавать новые архивы, про-

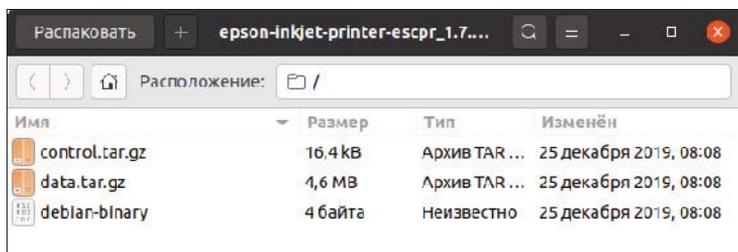


Рис. 20.1. Менеджер архивов

смаатривать и изменять содержимое уже существующих архивов, извлекать файлы из архива.

Менеджер архивов позволяет работать с двумя типами архивов:

- ◆ несжатыми архивами: ar, iso и tar;
- ◆ сжатыми архивами: 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 архиватора 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 **Сжать....** Откроется диалоговое окно создания нового архива (рис. 20.2).

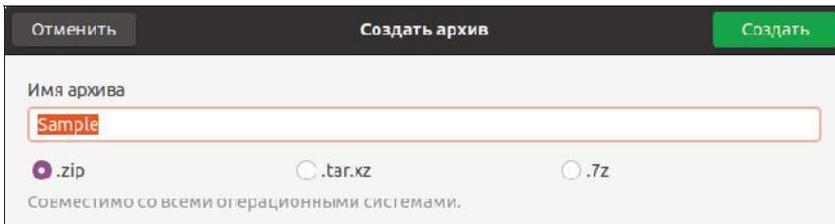


Рис. 20.2. Создание архива

В поле ввода **Имя архива** вы можете задать имя создаваемого архива, а чуть ниже установить переключатель нужного вам формата сжатия. Архив будет создан в текущем каталоге.

Сейчас мы рассмотрели простейший вариант создания архива в текущем каталоге, однако при архивации вы также можете задать и другие параметры, такие как пароль для архива, шифрование списка файлов или разделение создаваемого архива строго по определенному размеру. Все эти возможности доступны при создании архива из окна самой программы Менеджер архивов.

Запуск Менеджера архивов можно произвести из меню приложений либо командой Терминала:

```
file-roller
```

В меню Менеджера архивов выберите пункт **Создать архив....** Откроется окно создания архива с дополнительными параметрами, которые не были доступны нам из меню файлового менеджера Nautilus (рис. 20.3).

Дополнительные опции доступны в раскрывающемся списке **Другие параметры.**

#### **ПРИМЕЧАНИЕ**

Обратите внимание, что дополнительные опции доступны не для всех типов создаваемых архивов.

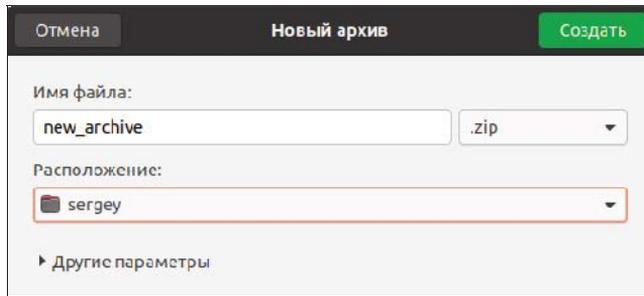


Рис. 20.3. Расширенное меню создания архива

После того как вы задали необходимые параметры, нажмите кнопку **Создать**. Откроеется окно добавления файлов в создаваемый архив, который будет создан в каталоге, указанном в параметре **Расположение** (рис. 20.4).

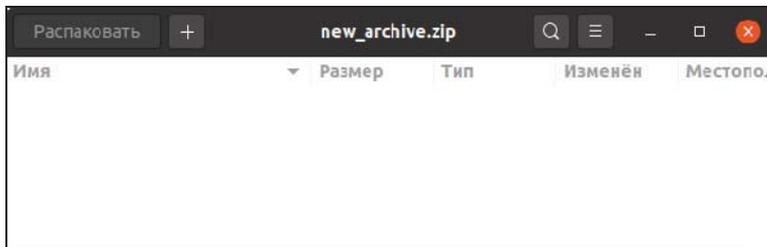


Рис. 20.4. Окно создания архива

Для того чтобы в создаваемый архив добавить файл или каталог, нажмите кнопку с изображением знака "плюс" и в окне добавления файлов отметьте флажком необходимые файлы, после чего нажмите кнопку **Добавить**. Выбранные вами файлы и каталоги будут помещены в создаваемый архив (рис. 20.5).

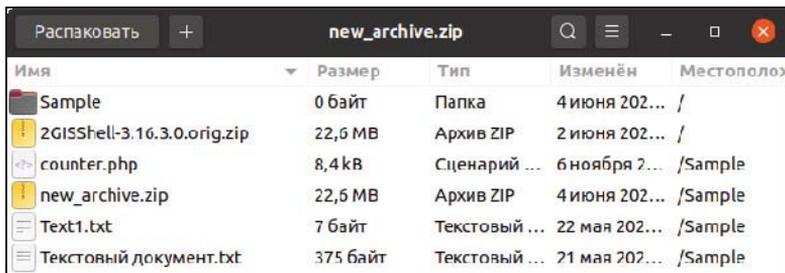


Рис. 20.5. Окно архива с добавленными файлами

### ПРИМЕЧАНИЕ

Обратите внимание, что архив создается после того, как вы добавите в него файл. До этого момента архив еще не существует.

Таким незамысловатым способом создаются архивы в графическом режиме с помощью Менеджера архивов.

Для просмотра содержимого архива дважды щелкните левой кнопкой мыши по нужному архиву, и Менеджер архивов отобразит его содержимое.

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 в текущую папку** либо **Извлечь в...** с целью указания требуемого каталога для распаковки. Архив будет распакован. То же самое можно проделать и из открытого окна Менеджера архивов, если на панели инструментов нажать кнопку **Распаковать**.

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 зараз 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 [ключи] [файл]
```

Необязательный 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 необязательный параметр *ключи* (табл. 20.1).

Таблица 20.1. Ключи команды tar

Сокращенный вариант	Полный вариант	Описание
-A	--catenate, --concatenate	Присоединяет tar-файлы к уже существующему архиву
-b	--blocking-factor= <i>размер_блока</i>	Разбивает создаваемый архив на блоки, равные указанному значению <i>размер_блока</i>

Таблица 20.1 (окончание)

Сокращенный вариант	Полный вариант	Описание
-c	--create	Создает новый архив
-C	--directory= <i>каталог</i>	Извлекает содержимое в указанный каталог
	--delete	Удаляет ненужный объект из архива
-f	--file= <i>архив</i>	Создает архив в виде файла на диске. Задает имя архива, который нужно создать или извлечь
-j	--bzip2	Пропускает созданный архив через архиватор bzip2. К имени файла добавляется расширение .bz2
-J	--xz	Пропускает созданный архив через архиватор xz
-k	--keep-old-files	Обновляет архив без перезаписи уже существующих файлов
	--overwrite	Перезаписывает существующие файлы при извлечении их из архива
-p	--preserve-permissions, --same-permissions	Извлекает информацию о правах доступа к файлу
-r	--append	Добавляет файлы в конец архива
-t	--list	Выводит список содержимого архива
-u	--update	Обновляет в архиве файлы на более новые
-v	--verbose	Выводит подробную информацию об обрабатываемых файлах
-x	--extract, --get	Извлекает файлы из архива
-z	--gzip, --gunzip, --ungzip	Пропускает созданный архив через архиватор gzip. К имени файла добавляется расширение .gz
-Z	--compress, --uncompress	Пропускает созданный архив через архиватор compress

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/
```

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 таком 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 успешности 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 формировать в два этапа. Для того 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.

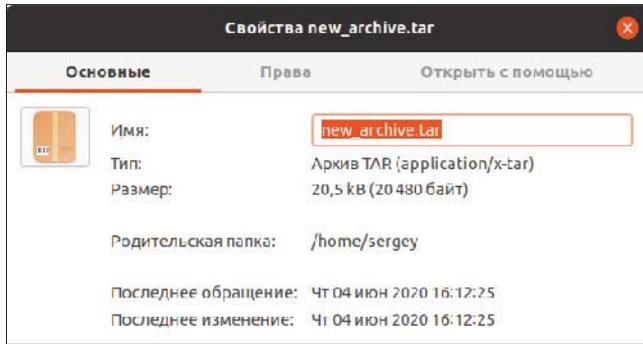
Давайте продемонстрируем создание сжатого `gzip`-архива одной командой:

```
tar -cvzf new_archive.tar.gz sample/
```

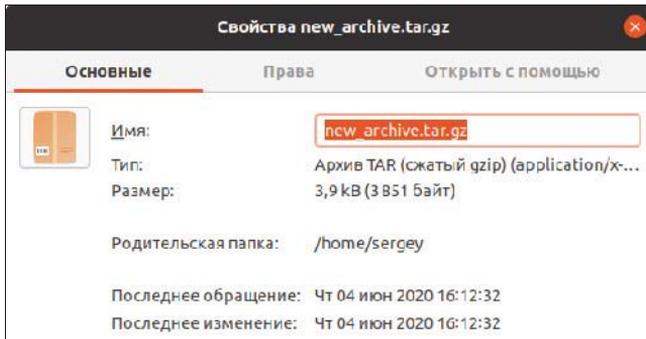
После выполнения данной команды вы получите сжатый архив. Для сравнения взгляните на рис. 20.6.

Обратите внимание на размер получившихся файлов.

Теперь проясним один момент: почему в имени файла при создании архива мы указываем два расширения — `tar.gz`. Разумеется, можно не указывать расширение, и архив все равно будет создан, но это будет неправильным решением. Лучше следовать тем правилам, которые были выработаны годами, и, согласитесь, 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.



а



б

Рис. 20.6. Свойства несжатого (а) и сжатого (б) архивов

Извлечение файлов из сжатого архива ничем не отличается от его создания:

```
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, 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
```

После выполнения 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 распаковки 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 нужно выложить его 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 [ключи] [файл]
```

Необязательный параметр *файл* принимает имя файла, который нужно сжать.

Вкратце рассмотрим основные значения, которые принимает необязательный параметр *ключи* (табл. 20.2).

**Таблица 20.2.** Ключи 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	Выводит полученный архив в стандартный поток вывода, например в окно Терминала
-d	--decompress	Распаковывает архив

Таблица 20.2 (окончание)

Сокращенный вариант	Полный вариант	Описание
-f	--force	Сжимает или распаковывает архив 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	Не удаляет обработанные (сжатые) файлы
-l	--list	Выводит информацию об обрабатываемом файле, такую как размер сжатого файла, размер оригинального файла, коэффициент сжатия и имя оригинального файла
-n	--no-name	При сжатии не сохраняет исходное имя и время создания файла
-N	--name	Сохраняет или восстанавливает оригинальное имя и время создания файла
-q	--quiet	Подавляет выдачу предупреждающих сообщений
-r	--recursive	Рекурсивно обрабатывает каталоги, когда задан шаблон имени файла
-S	--suffix= <i>суффикс</i>	Задает требуемый суффикс, отличный от .gz
-t	--test	Тестирует целостность сжатого архива
-v	--verbose	Задает подробный вывод информации об обрабатываемых файлах
	--rsyncable	Выполняет архивацию, используя протокол rsync
-1	--fast	Задает самую высокую скорость сжатия. В таком режиме сжатие данных незначительное. Если параметр не указан, уровень сжатия равен 6
-9	--best	Задает самый высокий коэффициент сжатия. При таком режиме сжатие данных занимает значительное время. Если параметр не указан, уровень сжатия равен 6

Давайте научимся работать с этой командой. Итак, у нас есть несжатый архив `new_archive.tar`, который мы хотим сжать. Для этого выполним 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.

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`. И 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 [ключи] [файл]
```

Необязательный параметр *файл* принимает имя файла, который должен подвергнуться сжатию.

Команда `bzip2` имеет практически те же самые ключи, 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 самыми распространенными форматами архивов 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 извлекает 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 все ее возможности вам вряд ли когда-нибудь понадобятся, 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
```

Эта 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 архивным форматом является `7Z`, а архив имеет суффикс `7z`. Для извлечения файлов из таких архивов вам потребуется установка дополнительного архиватора, т. к. данный формат поддерживается программой `7za`, которая не предустановлена в ОС Ubuntu (устанавливается с помощью команды `sudo apt install p7zip-full`). 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.

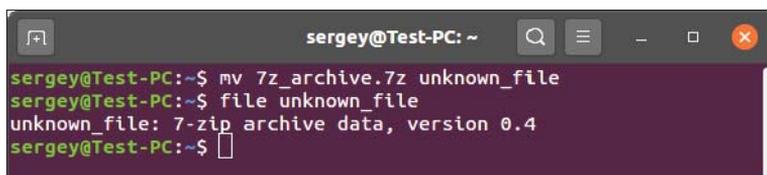
## Неизвестный тип файла

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. 20.7).



```
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 0.4
sergey@Test-PC:~$
```

Рис. 20.7. 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.	Установка и использование ОС Ubuntu в VirtualBox
Глава 31.	Создание и установка сертификатов Let's Encrypt
Глава 32.	Настройка файлового сервера Samba



## 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.

### **ПРИМЕЧАНИЕ**

Современные ОС позволяют создать несколько учетных записей с разными правами и осуществлять одновременную работу сразу нескольких учетных записей. Подключение и вход осуществляется удаленно. При этом другие пользователи, уже выполнившие вход в учетную запись, могут дальше продолжать работать с ОС.

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 различных групп 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 наделение 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?

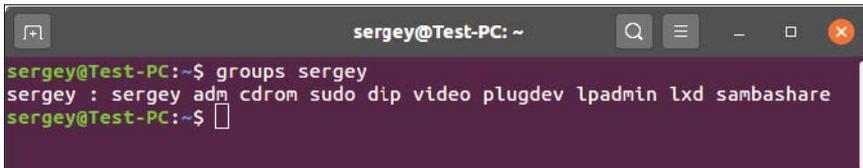
## Права доступа в Linux

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. Напри-

мер, 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`, `lxd` и `sambashare`.

Таким образом, можно узнать, к какой группе принадлежит учетная запись.



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ groups sergey  
sergey : sergey adm cdrom sudo dip video plugdev lpadmin lxd sambashare  
sergey@Test-PC:~$
```

Рис. 21.1. Просмотр групп учетной записи

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).

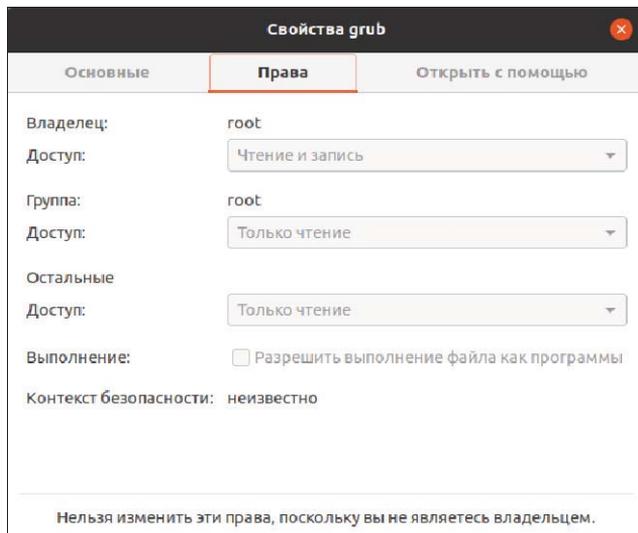


Рис. 21.2. Просмотр сведений о владельце файла

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. Они 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** окна свойств файла. Это только самая малая часть того, что предусмотрено разработчиками операционной системы на базе Linux. На деле ОС Ubuntu Linux позволяет выполнять более гибкую настройку прав доступа, нежели это реализовано в свойствах файла на вкладке **Права**.

В ОС Ubuntu Linux каждому файлу соответствует определенный набор прав доступа, предоставленный в виде 9 битов. Каждые 3 бита из этого набора определяют права для владельца файла (**user**), группы (**group**) и всех остальных (**other**) (рис. 21.3).

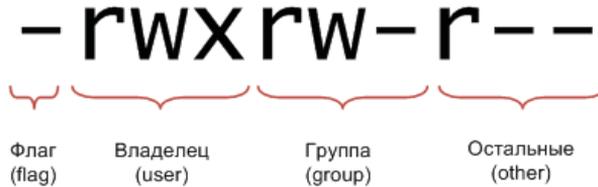


Рис. 21.3. Группы прав доступа

Первый символ (-) — это флаг (*flag*), который указывает на то, что перед нами: файл, каталог и т. п.

Буквенные символы `rwx` 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 1210 apr 29 10:48 /etc/default/grub
sergey@Test-PC:~$

```

Рис. 21.4. Ответ команды `ls -l`Таблица 21.1. Права доступа к файлу `/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 l — 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 используются 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 (`---`), 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.

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-----`, означающие, 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 пушей убедительности выполните команду `ls -l ~/sample/` (рис. 21.6).

Опять отказ в доступе к файлам внутри 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).

На пятой строке вывода команды `ls -l ~` в первой колонке мы видим все те же права `r-----`.

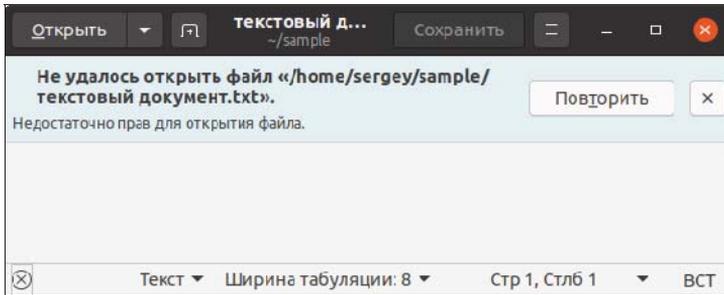


Рис. 21.5. Отказ в доступе к файлу текстовый документ.txt

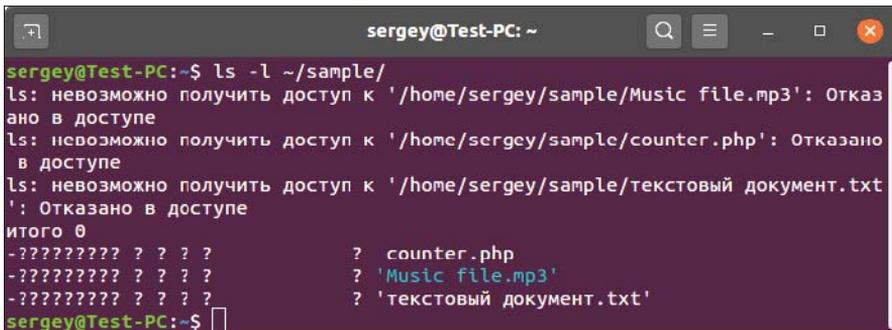


Рис. 21.6. Попытка просмотреть содержимое каталога ~/sample/

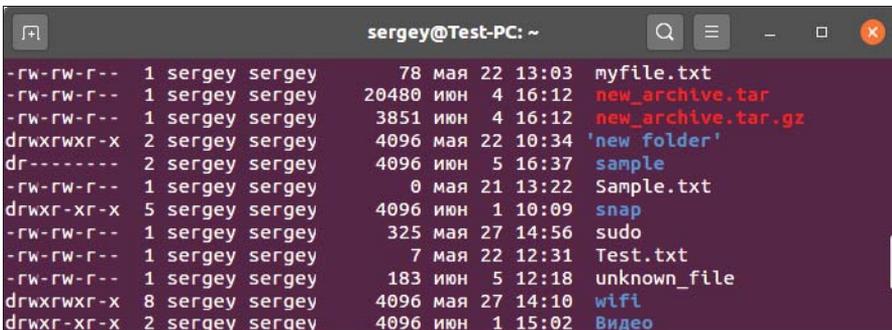


Рис. 21.7. Просмотр прав доступа домашнего каталога текущего пользователя

**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 манипуля-

ции 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 к их атрибутам, но даже в этом случае вам не удастся манипулировать 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 уже привычного для нас буквенного представления прав доступа.

Для примера возьмем права доступа для владельца файла `rw-`. Владельцу 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 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 (восьмеричная)	ВІN (двоичная)	Mask (маска)
0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
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 их виде, 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, суперпользователь 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 резервированные 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 различаться. Этим самым ОС "отличает" друг от друга учетные записи.

*SUID (Set User Identifier)* — бит смены идентификатора пользователя. Смысл этого бита 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 суперпользователя root на время выполнения этой программы.

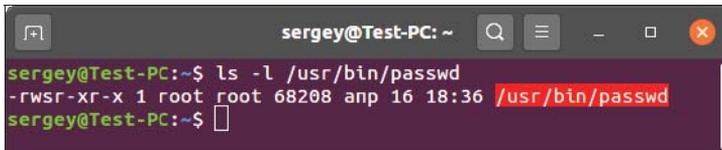
Самый яркий пример — это смена пароля пользователя. Для того чтобы получить доступ к файлу */etc/passwd*, к которому есть доступ только у суперпользователя root (являющегося владельцем файла), необходимо запустить программу */usr/bin/passwd*, у которой установлен бит идентификатора суперпользователя 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 применяется 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).



```
sergey@Test-PC: ~
sergey@Test-PC:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 68208 apr 16 18:36 /usr/bin/passwd
sergey@Test-PC:~$
```

Рис. 21.8. Просмотр символьных битов

Как мы видим, в первой колонке, так называемом триplete 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 суперпользователя `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 суперпользователь 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 такие 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 июн  8 09:15  testdir
-rw-rw-r--  1 sergey sergey         0 июн  8 09:15  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]
```

Необязательный 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 трехбитного 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 от первой цифры в маске. Таким образом, получается  $6 - 0 = 6$ . Затем то же самое проделывается 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, вы больше не будете 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.

Параметр *оператор* определяет ту операцию, которая будет передана программе `chmod`, и принимает значения, перечисленные в табл. 21.8.

Параметр *режимы* определяет, какие именно права будут добавлены или удалены, и принимает значения, перечисленные в табл. 21.9.

Таблица 21.7. Параметр *ССЫЛКИ*

Значение параметра <i>ССЫЛКИ</i>	Класс пользователей	Описание
u	user	Владелец файла
g	group	Группа пользователей файла
o	other	Остальные пользователи
a	all	Все пользователи

Таблица 21.8. Параметр *оператор*

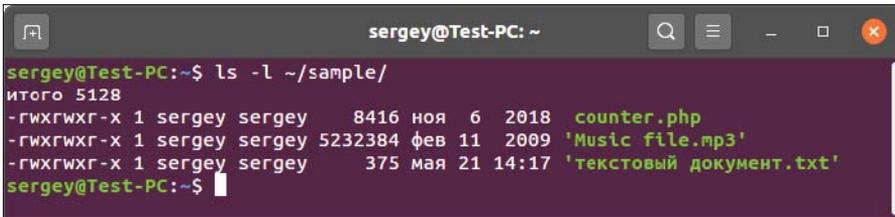
Значение параметра <i>оператор</i>	Описание
+	Добавить права
-	Удалить права
=	Установить права

Таблица 21.9. Параметр *режимы*

Значение параметра <i>режимы</i>	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)
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 5128
-rwxrwxr-x 1 sergey sergey 8416 ноя 6 2018 counter.php
-rwxrwxr-x 1 sergey sergey 5232384 фев 11 2009 'Music file.mp3'
-rwxrwxr-x 1 sergey sergey 375 мая 21 14:17 'текстовый документ.txt'
sergey@Test-PC:~$

```

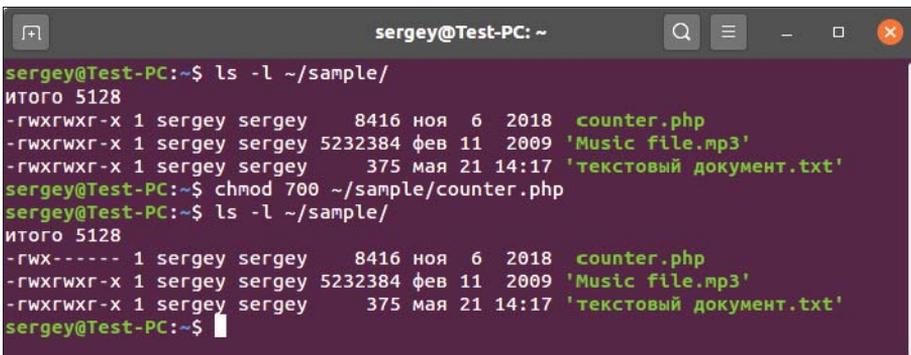
Рис. 21.9. Просмотр прав доступа файлов в каталоге sample

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).



```

sergey@Test-PC: ~
sergey@Test-PC:~$ ls -l ~/sample/
итого 5128
-rwxrwxr-x 1 sergey sergey 8416 ноя 6 2018 counter.php
-rwxrwxr-x 1 sergey sergey 5232384 фев 11 2009 'Music file.mp3'
-rwxrwxr-x 1 sergey sergey 375 мая 21 14:17 'текстовый документ.txt'
sergey@Test-PC:~$ chmod 700 ~/sample/counter.php
sergey@Test-PC:~$ ls -l ~/sample/
итого 5128
-rwx----- 1 sergey sergey 8416 ноя 6 2018 counter.php
-rwxrwxr-x 1 sergey sergey 5232384 фев 11 2009 'Music file.mp3'
-rwxrwxr-x 1 sergey sergey 375 мая 21 14:17 'текстовый документ.txt'
sergey@Test-PC:~$

```

Рис. 21.10. Права доступа к файлу counter.php изменились

Мы задали права к файлу с помощью цифрового представления прав доступа, но помимо него можно задавать права и посредством 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 о 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.

**Таблица 21.10.** Параметр `-RV` и ключ `-v`

Ключ	Описание
<code>-R</code>	Рекурсивное изменение атрибутов каталога и его содержимого
<code>-V</code>	Вывод полной информации и версии программы <code>chattr</code>
<code>-v</code>	Установка номера версии или генерации файла

Параметр *атрибуты* служит для добавления, снятия или установки новых атрибутов и может принимать значения из табл. 21.11.

**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

Таблица 21.11 (окончание)

Значение	Описание
D	Синхронное обновление каталогов. Внесенные изменения синхронно записываются на диск
a	Разрешает лишь добавлять записи. Установить или снять этот атрибут может только суперпользователь root
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 суперпользователь root
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 конец 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

Для установки перечисленных выше атрибутов имеются специальные операторы (табл. 21.12).

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, так же 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).

Как мы можем наблюдать, в первой колонке ответа команды (`-u-----e--`) появился символ `u`. Это означает, что файл помечен как неудаляемый.

```

sergey@Test-PC: ~
sergey@Test-PC:~$ chattr +u ~/sample/counter.php
sergey@Test-PC:~$ lsattr ~/sample/counter.php
-u-----e---- /home/sergey/sample/counter.php
sergey@Test-PC:~$
  
```

Рис. 21.11. Просмотр атрибутов файла

**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, допускаемому 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 purple background. The title bar shows 'sergey@Test-PC: ~'. The terminal output is as follows:

```
261826 snap
264110 sudo
528591 testdir
264980 test.txt
263988 Test.txt
```

Рис. 22.1. Номер индексного дескриптора

Таким образом, у файла `sudo` номер индексного дескриптора равен 264110. 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. 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). Она 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.

Основные значения, которые принимает параметр *ключ*, приведены в 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>	Создает резервную копию для каждого целевого файла в соответствии с заданным аргументом <i>контроль</i> . Допустимые значения этого аргумента следующие: <ul style="list-style-type: none"> <li>• <code>none, off</code> — отключение создания резервных копий;</li> <li>• <code>numbered, t</code> — создание нумерованных копий;</li> <li>• <code>existing, nil</code> — при существовании нумерованных резервных копий будет продолжено создание нумерованных копий. В противном случае будут созданы простые копии;</li> <li>• <code>simple, never</code> — создание простых копий</li> </ul>
<code>-b</code>		Аналогичен ключу <code>--backup</code> , только не позволяет принимать аргумент
<code>-d, -F</code>	<code>--directory</code>	Позволяет суперпользователю <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>	Удаляет существующие целевые файлы
<code>-i</code>	<code>--interactive</code>	Запрашивает подтверждение перед удалением ссылки
<code>-L</code>	<code>--logical</code>	Разыменовывает символическую ссылку, т. е. позволяет получить объект по ссылке
<code>-n</code>	<code>--no-dereference</code>	Рассматривает <i>ИМЯ_ССЫЛКИ</i> как обычный файл, даже если это символическая ссылка на каталог
<code>-P</code>	<code>--physical</code>	Создает жесткую ссылку непосредственно на саму символическую ссылку
<code>-r</code>	<code>--relative</code>	Создает символическую ссылку относительно ее расположения
<code>-s</code>	<code>--symbolic</code>	Создает символическую ссылку вместо жесткой ссылки
<code>-S</code>	<code>--suffix=суффикс</code>	Переопределяет обычный суффикс резервных копий на тот, что задан в аргументе <i>суффикс</i> . По умолчанию суффикс резервных копий — это <code>~</code>
<code>-t</code>	<code>--target-directory=каталог</code>	Задаёт <i>каталог</i> , в котором должны быть созданы указанные ссылки
<code>-T</code>	<code>--no-target-directory</code>	Позволяет всегда считать <i>ИМЯ_ССЫЛКИ</i> 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).

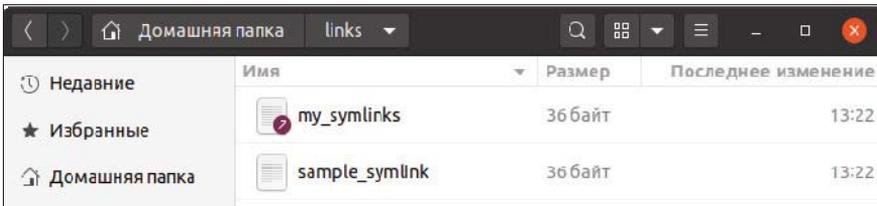


Рис. 22.2. Символическая ссылка `my_symlinks`

Убедимся в том, что по символической ссылке доступен просмотр содержимого файла `sample_symlink`:

```
cat my_symlinks
```

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).

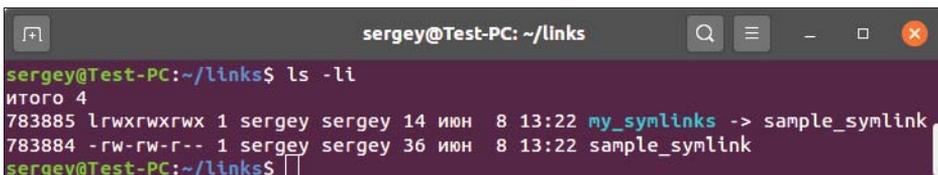


Рис. 22.3. Просмотр содержимого каталога `~/links/`

Нас интересует третья строка вывода Терминала. В первом столбце вывода число 783885 означает номер дескриптора файла, тот самый inode. Это и есть символическая ссылка. Обратите внимание, что индексный дескриптор ссылки и самого файла различается 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.

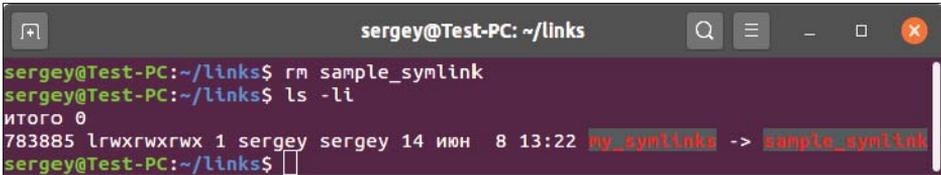
Третьим столбцом значится счетчик жестких ссылок на данный файл. В нашем случае у данного файла только одна жесткая ссылка, которая указывает на сам файл. Счетчик ссылок справедлив только для жестких ссылок и не распространяется на символические ссылки.

Последним столбцом следует строка `"my_symlinks -> sample_symlink"`, которая сообщает, что ссылка `my_symlinks` указывает на файл `sample_symlink`. Указателем в данном случае выступают символы `->`.

Попробуем удалить исходный файл `sample_symlinks` и вывести информацию о содержимом каталога `~/links/`.

```
rm sample_symlink
ls -li
```

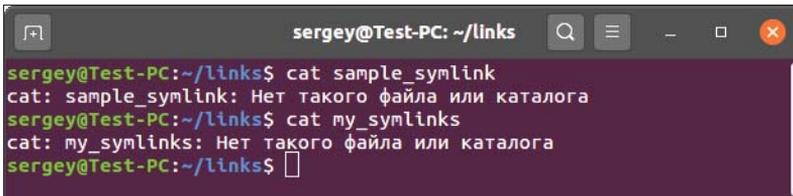
После выполнения этих двух команд мы увидим следующую картину (рис. 22.4).



```
sergey@Test-PC: ~/links
sergey@Test-PC:~/links$ rm sample_symlink
sergey@Test-PC:~/links$ ls -li
итого 0
783885 lrwxrwxrwx 1 sergey sergey 14 июн  8 13:22 my_symlinks -> 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, 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_symlinks
cat: my_symlinks: Нет такого файла или каталога
sergey@Test-PC:~/links$
```

Рис. 22.5. Запрашиваемый файл `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, равно 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.

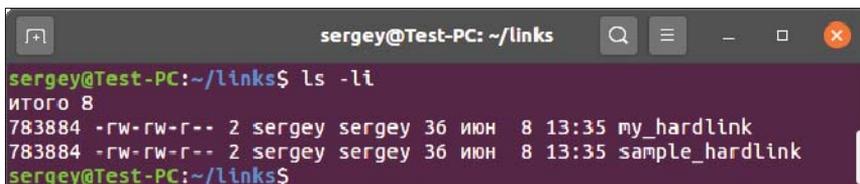
Теперь, если просмотреть содержимое файла по жесткой ссылке, будет выведено содержимое файла `sample_hardlink`:

```
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
783884 -rw-rw-r-- 2 sergey sergey 36 июн  8 13:35 my_hardlink
783884 -rw-rw-r-- 2 sergey sergey 36 июн  8 13:35 sample_hardlink
sergey@Test-PC:~/links$
```

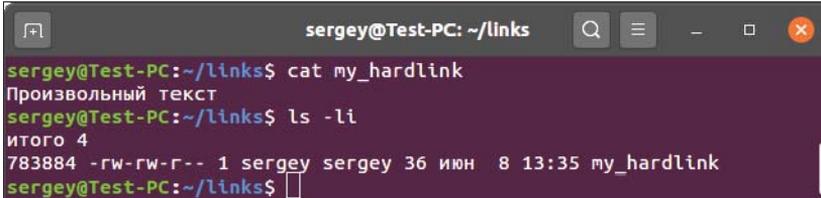
Рис. 22.6. Жесткая ссылка на файл `sample_hardlink`

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$ cat my_hardlink
Произвольный текст
sergey@Test-PC:~/links$ ls -li
итого 4
783884 -rw-rw-r-- 1 sergey sergey 36 июн  8 13:35 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 выяснили, что каждый файл имеет свой уникальный индексный дескриптор — `inode`, который не может повториться в пределах одной файловой системы.

Мы научились создавать и символические, и жесткие ссылки. После этого на конкретных примерах мы посмотрели, 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 автоматическое монтирование доступно не во всех Linux-подобных операционных системах. Так, в серверных версиях операционных систем автоматическое монтирование считается небезопасным и выполняется только системным администратором. Правда, это вовсе не означает, что вы теперь не сможете воспользоваться возможностями автоматического монтирования подключаемых устройств. Вопрос только в том, целесообразно и безопасно ли автоматическое монтирование подключаемых устройств на рабочем сервере.

Совсем по-иному дела обстоят с домашними компьютерами, для которых автоматическое монтирование устройств просто необходимо, поскольку за день мы подключаем несколько устройств и по несколько раз. Именно по этой причине в ОС Ubuntu решено было включить автоматическое монтирование 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).

```

1 # /etc/fstab: static file system information.
2 #
3 # Use 'blkid' to print the universally unique identifier for a
4 # device; this may be used with UUID= as a more robust way to name devices
5 # that works even if disks are added and removed. See fstab(5).
6 #
7 # <file system> <mount point> <type> <options> <dump> <pass>
8 # / was on /dev/sda6 during installation
9 UUID=26d013b2-b6fd-4b39-b223-31c66278e32e / ext4 errors=remount-ro 0 1
10 # /home was on /dev/sda7 during installation
11 UUID=5ec7a6ec-dfd9-4380-b217-5ec0c814ee2f /home ext4 defaults 0 2
12 # swap was on /dev/sda5 during installation
13 UUID=5bbb6e79-6edb-4f6e-8ba1-beb7804a3e4b none swap sw 0 0

```

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 наименований

Существует несколько способов наименования 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 файлу 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 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.

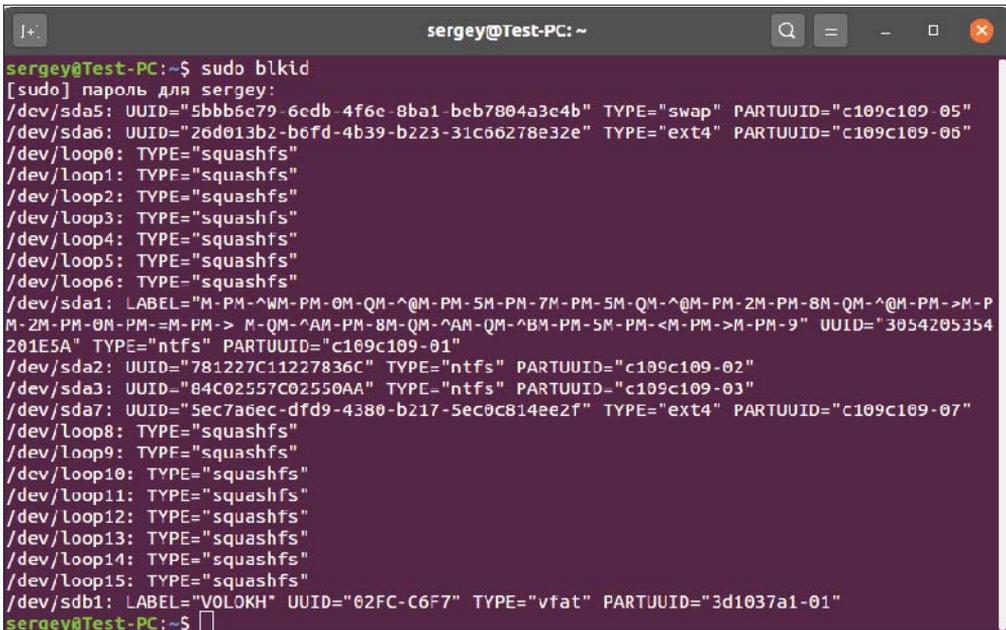
Теперь перейдем к более детальному рассмотрению способов именовании блочных устройств.

## Именование 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:~$ sudo blkid
[sudo] пароль для sergey:
/dev/sda5: UUID="5bbb5ec79-6cdb-4f6c-8ba1-bcb7804a3c4b" TYPE="swap" PARTUUID="c109c109-05"
/dev/sda6: UUID="26d013b2-b6fd-4b39-b223-31c66278e32e" TYPE="ext4" PARTUUID="c109c109-06"
/dev/loop0: TYPE="squashfs"
/dev/loop1: TYPE="squashfs"
/dev/loop2: TYPE="squashfs"
/dev/loop3: TYPE="squashfs"
/dev/loop4: TYPE="squashfs"
/dev/loop5: TYPE="squashfs"
/dev/loop6: TYPE="squashfs"
/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-P
M-2M-PM-0M-PM-^M-PM-> M-QM-^AM-PM-8M-QM-^AM-QM-^BM-PM-5M-PM-^M-PM->M-PM-9" UUID="3054205354
201E5A" TYPE="ntfs" PARTUUID="c109c109-01"
/dev/sda2: UUID="781227C11227836C" TYPE="ntfs" PARTUUID="c109c109-02"
/dev/sda3: UUID="84C02557C02550AA" TYPE="ntfs" PARTUUID="c109c109-03"
/dev/sda7: UUID="5ec7a6ec-dfd9-4380-b217-5ec9c814ee2f" TYPE="ext4" PARTUUID="c109c109-07"
/dev/loop8: TYPE="squashfs"
/dev/loop9: TYPE="squashfs"
/dev/loop10: TYPE="squashfs"
/dev/loop11: TYPE="squashfs"
/dev/loop12: TYPE="squashfs"
/dev/loop13: TYPE="squashfs"
/dev/loop14: TYPE="squashfs"
/dev/loop15: TYPE="squashfs"
/dev/sdb1: LABEL="VOLOKH" UUID="02FC-C6F7" TYPE="vfat" PARTUUID="3d1037a1-01"
sergey@Test-PC:~$
```

Рис. 23.2. Получение UUID подключенных устройств

Для примера рассмотрим следующую строку:

```
/dev/sda6: UUID="26d013b2-b6fd-4b39-b223-31c66278e32e" TYPE="ext4"
PARTUUID="c109c109-06"
```

Данная строка состоит из четырех колонок:

- ◆ /dev/sda6: — имя раздела;
- ◆ UUID="26d013b2-b6fd-4b39-b223-31c66278e32e" — уникальный идентификатор UUID;
- ◆ TYPE="ext4" — тип раздела, т. е. его файловая система;
- ◆ PARTUUID=" c109c109-06" — 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 монтирования, т. к. он уже монтируется при загрузке операционной системы, и поэтому данная строка была показана только в качестве примера.

## Именование посредством имен устройств

Стоит напомнить, что данный способ именованья 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.

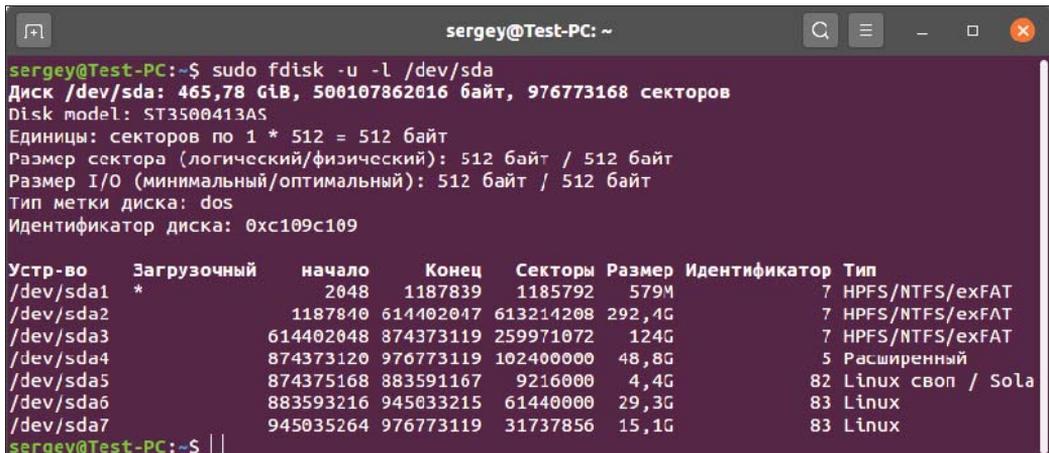
Итак, чтобы получить имена устройств, точнее, имена разделов, выполните следующую команду:

```
sudo fdisk -u -l
```

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
Диск /dev/sda: 465,78 GiB, 500107862016 байт, 976773168 секторов
Disk model: ST3500413AS
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
Тип метки диска: dos
Идентификатор диска: 0xc109c109

Устр-во   Загрузочный   начало      Конец      Секторы  Размер  Идентификатор  Тип
/dev/sda1 *                2048        1187839    1185792    579M    7 HPFS/NTFS/exFAT
/dev/sda2                1187840    614402047  613214208  292,4G  7 HPFS/NTFS/exFAT
/dev/sda3                614402048  874373119  259971072  124G    7 HPFS/NTFS/exFAT
/dev/sda4                874373120  976773119  102400000  48,8G   5 Расширенный
/dev/sda5                874375168  883591167  9216000    4,4G    82 Linux swap / Solaris
/dev/sda6                883593216  945033215  61440000   29,3G   83 Linux
/dev/sda7                945035264  976773119  31737856   15,1G   83 Linux

sergey@Test-PC:~$
```

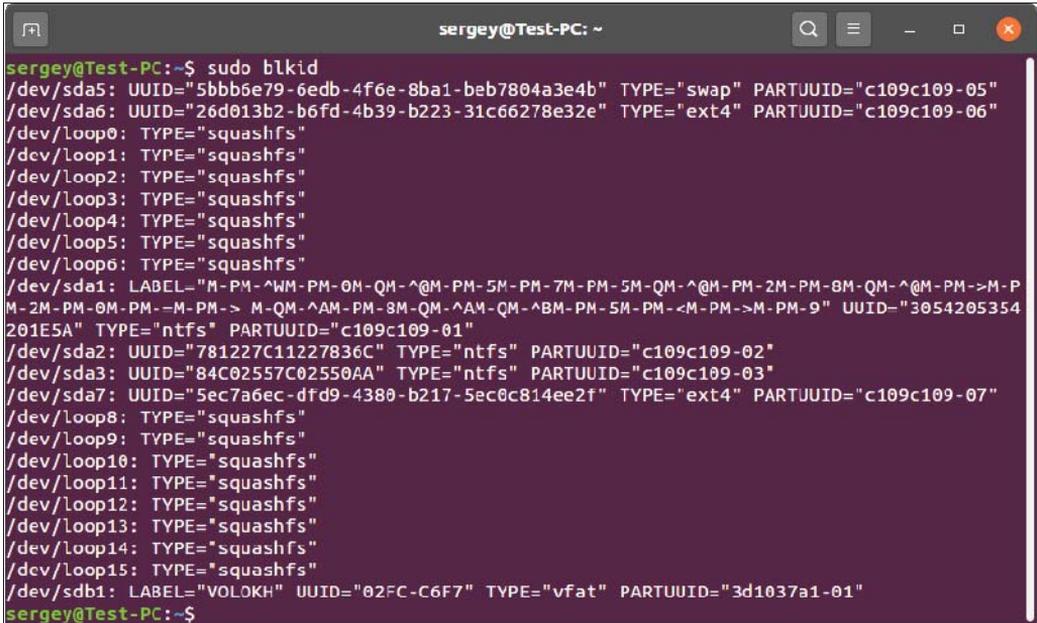
Рис. 23.3. Вывод информации о наименованиях разделов жесткого диска

Вывод команды `fdisk` состоит из восьми колонок, но нас интересует только первая колонка **Устр-во**, в которой перечислены имена разделов, присутствующих в ОС. При таком способе монтирования используются имена устройств, которые указаны в первой колонке вывода команды.

## Именованние посредством меток устройств

Современные файловые системы позволяют задавать метку (label) раздела. Однако это не обязательно, и метку задают в основном для удобства.

Выполните знакомую нам команду `sudo blkid` и внимательно посмотрите результат вывода (рис. 23.4).



```
sergey@Test-PC: ~
sergey@Test-PC:~$ sudo blkid
/dev/sda5: UUID="5bbb5e79-6edb-4f6e-8ba1-beb7804a3e4b" TYPE="swap" PARTUUID="c109c109-05"
/dev/sda6: UUID="26d013b2-b6fd-4b39-b223-31c66278e32e" TYPE="ext4" PARTUUID="c109c109-06"
/dev/loop0: TYPE="squashfs"
/dev/loop1: TYPE="squashfs"
/dev/loop2: TYPE="squashfs"
/dev/loop3: TYPE="squashfs"
/dev/loop4: TYPE="squashfs"
/dev/loop5: TYPE="squashfs"
/dev/loop6: TYPE="squashfs"
/dev/sda1: LABEL="М-ММ-^WM-PM-0М-QМ-^@М-PM-5М-PM-7М-PM-5М-QМ-^@М-PM-2М-PM-8М-QМ-^@М-PM->М-РМ-2М-PM-0М-PM-→М-PM-> М-QМ-^AM-PM-8М-QМ-^AM-QМ-^BM-PM-5М-PM-←М-PM->M-PM-9" UUID="3054205354201E5A" TYPE="ntfs" PARTUUID="c109c109-01"
/dev/sda2: UUID="781227C11227836C" TYPE="ntfs" PARTUUID="c109c109-02"
/dev/sda3: UUID="84C02557C02550AA" TYPE="ntfs" PARTUUID="c109c109-03"
/dev/sda7: UUID="5ec7a6ec-dfd9-4380-b217-5ec6c814ee2f" TYPE="ext4" PARTUUID="c109c109-07"
/dev/loop8: TYPE="squashfs"
/dev/loop9: TYPE="squashfs"
/dev/loop10: TYPE="squashfs"
/dev/loop11: TYPE="squashfs"
/dev/loop12: TYPE="squashfs"
/dev/loop13: TYPE="squashfs"
/dev/loop14: TYPE="squashfs"
/dev/loop15: TYPE="squashfs"
/dev/sdb1: LABEL="VOLOKH" UUID="02FC-C6F7" TYPE="vfat" PARTUUID="3d1037a1-01"
sergey@Test-PC:~$
```

Рис. 23.4. Вывод информации о подключенных устройствах

Обратите внимание на десятую 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`.

В столбце **Метка** перечислены все заданные метки. Подключенные устройства отображаются по отдельности. Для переключения отображения другого диска выберите нужный диск в раскрывающемся списке на панели инструментов.

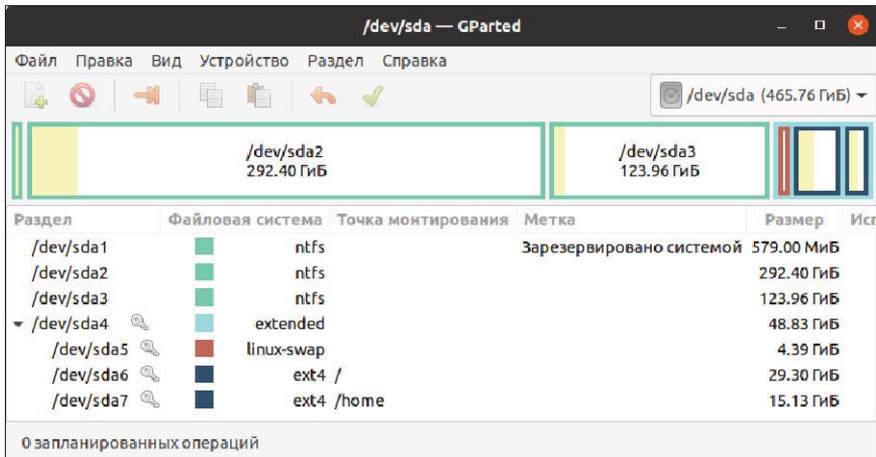


Рис. 23.5. Редактор разделов GParted

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 ввиду этого будет возникать путаница при монтировании устройств.

## Именование 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).

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).

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 `6VMSHN1R` диск получил длинный идентификационный номер `ata-ST3500413AS_6VMSHN1R`. Помимо этого каждый раздел получил суффикс `-part1`, `-part2` и т. д.

```

sergey@Test-PC: ~
sergey@Test-PC:~$ sudo hdparm -i /dev/sda

/dev/sda:

Model=ST3500413AS, FwRev=JC45, SerialNo=6VMSHN1R
Config={ HardSect NotMFM HdSw>15uSec Fixed DTR>10Mbs RotSpdTol>.5% }
RawCHS=16383/16/63, TrkSize=0, SectSize=0, ECCbytes=4
BuffType=unknown, BuffSize=16384kB, MaxMultSect=16, MultSect=16
CurCHS=16383/16/63, CurSects=16514064, LBA=yes, LBASects=976773168
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=no WriteCache=enabled
Drive conforms to: unknown: ATA/ATAPI-4,5,6,7

* signifies the current active mode

sergey@Test-PC:~$

```

Рис. 23.6. Получение информации о жестком диске

```

sergey@Test-PC: ~
sergey@Test-PC:~$ ls -l /dev/disk/by-id/
итого 0
lrwxrwxrwx 1 root root 9 июн  8 14:12 ata-Optiarc_DVD_RW_AD-7260S -> ../../sr0
lrwxrwxrwx 1 root root 9 июн  8 14:12 ata-ST3500413AS_6VMSHN1R -> ../../sda
lrwxrwxrwx 1 root root 10 июн  8 14:12 ata-ST3500413AS_6VMSHN1R-part1 -> ../../sda1
lrwxrwxrwx 1 root root 10 июн  8 14:12 ata-ST3500413AS_6VMSHN1R-part2 -> ../../sda2
lrwxrwxrwx 1 root root 10 июн  8 14:12 ata-ST3500413AS_6VMSHN1R-part3 -> ../../sda3
lrwxrwxrwx 1 root root 10 июн  8 14:12 ata-ST3500413AS_6VMSHN1R-part4 -> ../../sda4
lrwxrwxrwx 1 root root 10 июн  8 14:12 ata-ST3500413AS_6VMSHN1R-part5 -> ../../sda5
lrwxrwxrwx 1 root root 10 июн  8 14:12 ata-ST3500413AS_6VMSHN1R-part6 -> ../../sda6
lrwxrwxrwx 1 root root 10 июн  8 14:12 ata-ST3500413AS_6VMSHN1R-part7 -> ../../sda7

```

Рис. 23.7. Получение информации по разделам жесткого диска

## 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 этой команды практически ничем не отличаются от результатов выполнения тех команд, которые мы рассматривали при описании каждого метода именования. 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 суперпользователя `root`, поэтому запускаем редактирование файла с соответствующими правами:

```
sudo 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`, `xf`s, `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
auto	Позволяет файловой системе монтироваться автоматически либо после выполнения в Терминале команды <code>mount -a</code>

Таблица 23.1 (окончание)

Значение	Описание
noauto	Монтирует файловую систему не автоматически, а по запросу пользователя
defaults	Монтирует файловую систему со значениями по умолчанию, которые соответствуют набору опций: auto, async, exec, dev, rw, suid
dev	На монтируемой файловой системе создает файлы устройств
nodev	Запрещает создание файлов устройств на монтируемой файловой системе
exec	Исполняет бинарные файлы, расположенные на монтируемом разделе или диске
noexec	Запрещает исполнение бинарных файлов. Опцию можно использовать на несистемных разделах, иначе эта настройка приведет к неработоспособности ОС
suid	Записывает 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 — место, куда нужно примонтировать это устройство, например /home, /boot, /mnt. Параметр позволяет выполнять операции с SUID- и SGID-битами
nosuid	Запрещает выполнение операций с SUID- и SGID-битами
sync	Задаёт синхронное выполнение операций ввода-вывода
async	Асинхронно выполняет операции ввода-вывода
user	Разрешает монтирование файловой системы любому пользователю
nouser	Разрешает монтирование файловой системы только суперпользователю root
ro	Монтирует файловую систему только для чтения
rw	Монтирует файловую систему для чтения и записи
atime	Включает информацию о последнем времени доступа при каждом чтении файла
noatime	Отключает 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
```

Программа `blkid` покажет все подключенные блочные устройства вместе с их UUID. Последним в списке будет наш USB-накопитель (рис. 23.8).



```
sergey@Test-PC: ~  
/dev/loop11: TYPE="squashfs"  
/dev/loop12: TYPE="squashfs"  
/dev/loop13: TYPE="squashfs"  
/dev/loop14: TYPE="squashfs"  
/dev/loop15: TYPE="squashfs"  
/dev/sdb1: LABEL="VOLOKH" UUID="02FC-C6F7" TYPE="vfat" PARTUUID="3d1037a1-01"  
sergey@Test-PC:~$
```

Рис. 23.8. Вывод списка подключенных накопителей

В нашем случае подключенный накопитель имеет имя `/dev/sdb1`. Создадим для этого устройства каталог, к которому будет монтироваться файловая система:

```
sudo mkdir /mnt/volokh/
```

Каталог для монтирования создан, осталось составить запись для файла `/etc/fstab`:

```
UUID=02FC-C6F7 /mnt/volokh/ auto defaults 0 2
```

Разберем нашу запись:

- ◆ `02FC-C6F7` — 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 суперпользователя `root`:

```
sudo gedit /etc/fstab
```

В конец файла `/etc/fstab` дописываем нашу строку:

```
UUID=02FC-C6F7 /mnt/volokh/ auto defaults 0 2
```

Сохраняем сделанные нами изменения и перезагружаем операционную систему. После загрузки ОС в каталог `/mnt/volokh/` будет смонтирована файловая система съемного USB-носителя. Это означает, что на панели задач теперь не будет значка нашего USB-носителя, а 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=84C02557C02550AA /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-ST3500413AS_6VMSHN1R-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 вашего устройства, пусть это будет `/dev/sdb1`, и данное устройство необходимо смонтировать в каталог `/mnt/mydoc/`. Для этого выполните следующую команду:

```
sudo mount /dev/sdb1 /mnt/mydoc/
```

#### **ПРИМЕЧАНИЕ**

Помните, что точка монтирования должна существовать, иначе вы получите следующую ошибку: `"/mnt/mydoc/": точка монтирования не существует`.

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/sdb1
```

Данная команда позволит размонтировать указанное устройство. Стоит отметить, что вместо имени устройства вы можете задать монтируемый каталог. Так, следующая команда полностью аналогична предыдущей:

```
sudo umount /mnt/mydoc/
```

Обратите внимание, что и при монтировании, и при размонтировании устройства в файловом менеджере Nautilus в левом навигационном меню остается значок устройства. Дело в том, что размонтирование устройства вовсе не означает извлечение устройства. Для извлечения устройства выполните следующую команду:

```
sudo eject /dev/sdb1/
```

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.

### **ПРИМЕЧАНИЕ**

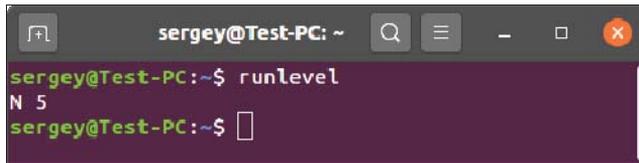
Как мы помним, на смену устаревшей системе BIOS пришла UEFI, а на смену MBR — GPT.

## Уровни выполнения

*Уровень выполнения* (runlevel) — это режим работы ОС, который определяет, какие задачи могут выполняться операционной системой в текущем состоянии.

Обычно встречается семь 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

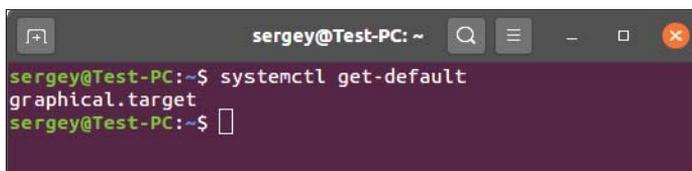
При отключенном графическом режиме ОС Ubuntu будет загружена со значением со вторым уровнем выполнения. В табл. 24.1 приведены все уровни выполнения.

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*). Он является синонимом однопользовательского режима, т. е. уровня 1.

Для того чтобы узнать уровень инициализации, т. е. какой уровень запуска по умолчанию установлен в ОС Ubuntu, выполните команду `systemctl get-default` (рис. 24.2).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ systemctl get-default  
graphical.target  
sergey@Test-PC:~$
```

Рис. 24.2. Уровень выполнения по умолчанию

Режим запуска `graphical.target` указывает на то, что по умолчанию запускается многопользовательский режим с поддержкой графического режима, т. е. пятый уровень выполнения.

Конфигурация загрузчика GRUB указывает, какой уровень выполнения запускать при загрузке ОС. Для настольных домашних систем это графический режим (`graphical.target`), а для серверной версии ОС Ubuntu — `multi-user.target`.

В случае если графическая среда отсутствует, срабатывает именно второй уровень выполнения `multi-user.target`.

Так как мы пользуемся обычной (не серверной) версией ОС Ubuntu, то по умолчанию для нее применяется пятый уровень выполнения, в чем мы могли убедиться, выполнив команду `runlevel`.

В зависимости от того, какой уровень запуска установлен по умолчанию, операционная система выполняет сценарии в соответствии со списком, приведенным в 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` или `k`. Сценарии, начинающиеся с буквы `s`, предназначены для запуска (от англ. *startup*), а сценарии, начинающиеся с буквы `k`, — для завершения работы (от англ. *kill*).

Во время загрузки операционной системы служба `systemd` "просматривает" каталог `/etc/systemd/system/` для того, чтобы найти установленную по умолчанию цель запуска. После того как цель запуска по умолчанию найдена, начинается запуск системных модулей, связанных с этой целью, чтобы система загружалась со всеми необходимыми запущенными процессами.

Служба `systemd` представляет собой обычный процесс, который работает в фоновом режиме и выполняет свои определенные функции, в конкретном случае — запуск необходимых процессов в зависимости от того или иного уровня выполнения.

Системные цели предназначены для автоматического запуска различных наборов служб в зависимости от функциональности, которая должна быть предоставлена этой целью. Так, для второго уровня цели (запуска) служба `systemd` не станет запускать службу диспетчера рабочих столов. Логично же, если операционная система

запускается в текстовом (консольном) режиме, то запускать графический рабочий стол было бы неразумно и лишено здравого смысла.

## 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 сработает ввиду того, что система инициализации `init` (точнее, система System V `init`) была заменена `upstart`, и в дальнейшем 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

Таблица 24.3 (окончание)

Уровень выполнения	Цели	Описание
4	runlevel4.target, multi-user.target	Многопользовательский режим. Обычно не используется
5	runlevel5.target, graphical.target	Многопользовательский режим с поддержкой графического режима
6	runlevel6.target, reboot.target	Уровень позволяет выполнить действия по перезагрузке операционной системы
emergency	emergency.target	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.

При запуске той или иной цели могут быть запущены так называемые подцели, которые представляют собой подобие модулей, без загрузки которых невозможно продолжение выполнения запуска цели. Так, цель `multi-user.target` запустит подцель `basic.target`, которая, в свою очередь, должна запустить `sockets.target`. Запуск подцелей необходим для нормального функционирования всей системы в целом. Если бы подцели не запускались, то возникла бы ситуация, когда та или иная служба будет отказываться работать по причине того, что для ее работы не запущен нужный сервис.

Для просмотра зависимости цели, например `graphical.target`, выполните следующую команду (рис. 24.3):

```
systemctl list-dependencies graphical.target
```

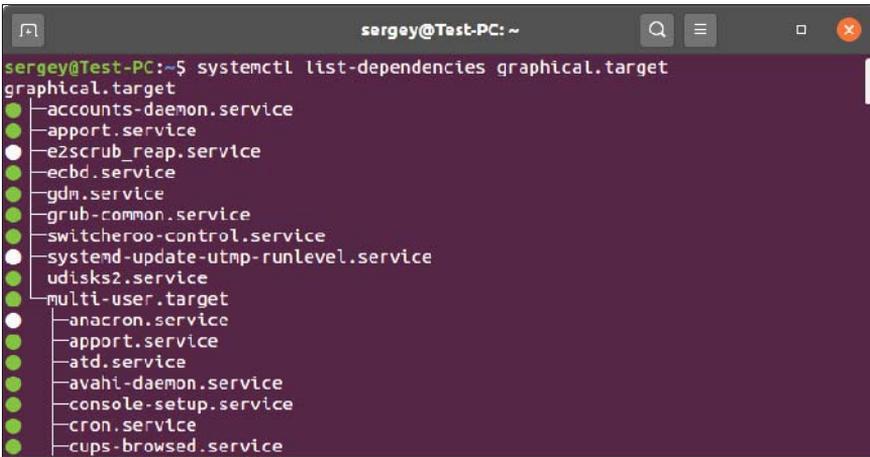
Как можно догадаться из результатов вывода команды, цель `graphical.target` запускает цель `multi-user.target`, у которой также есть свои подцели.

Обратите внимание, что цветные точки напротив цели указывают на следующие состояния:

- ◆ зеленый — служба или цель запущена и работает;
- ◆ красный — служба или цель не запустилась из-за фатальной ошибки;
- ◆ белый — служба или цель неактивна (не запущена), например, из-за того, что данная служба или цель на текущий момент времени остановлена либо не выполнено условие, при котором должен был произойти запуск службы.

Система инициализации `systemd` строилась с учетом максимального быстродействия, базовыми компонентами которой являются юниты.

*Юниты* (от англ. *unit* — единица) — это файлы конфигурации, хранящие информацию о службе, устройстве или сокете.



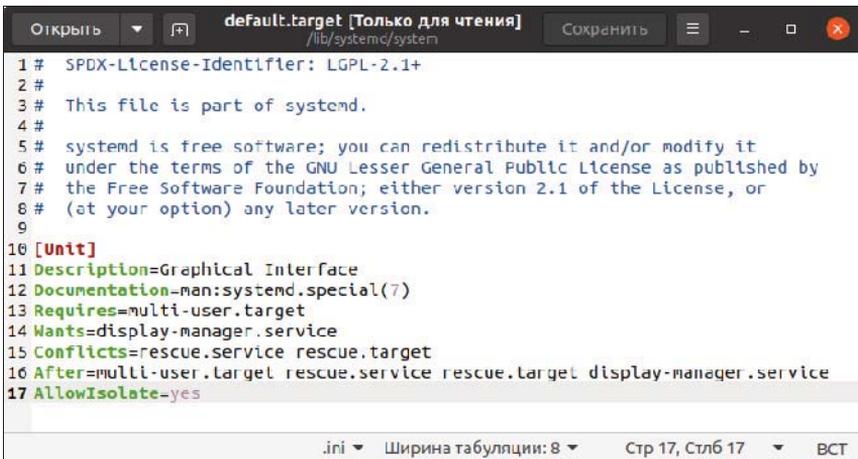
```

sergey@Test-PC: ~$ systemctl list-dependencies graphical.target
graphical.target
●─accounts-daemon.service
●─apport.service
●─e2scrub_reap.service
●─ecbd.service
●─gdm.service
●─grub-common.service
●─switcheroo-control.service
●─systemd-update-utmp-runlevel.service
●─udisks2.service
●─multi-user.target
●─anacron.service
●─appport.service
●─atd.service
●─avahi-daemon.service
●─console-setup.service
●─cron.service
●─cups-browsed.service

```

Рис. 24.3. Зависимость (подцели) цели graphical.target

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. 24.4).



```

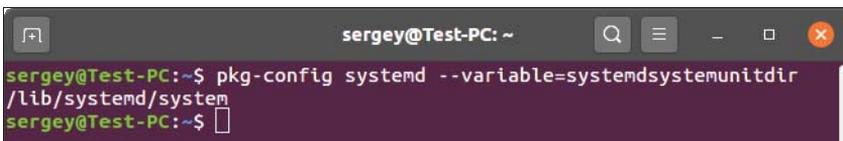
1 # SPDX-License-Identifier: LGPL-2.1+
2 #
3 # This file is part of systemd.
4 #
5 # systemd is free software; you can redistribute it and/or modify it
6 # under the terms of the GNU Lesser General Public License as published by
7 # the Free Software Foundation; either version 2.1 of the License, or
8 # (at your option) any later version.
9
10 [Unit]
11 Description=Graphical Interface
12 Documentation=man:systemd.special(7)
13 Requires=multi-user.target
14 Wants=display-manager.service
15 Conflicts=rescue.service rescue.target
16 After=multi-user.target rescue.service rescue.target display-manager.service
17 AllowIsolate=yes

```

Рис. 24.4. Содержимое цели /lib/systemd/system/default.target

Единицы (юниты) хранятся в каталоге /lib/systemd/system/. Выяснить расположение юнитов можно следующей командой (рис. 24.5):

```
pkg-config systemd --variable=systemdsystemunitdir
```



```

sergey@Test-PC: ~$ pkg-config systemd --variable=systemdsystemunitdir
/lib/systemd/system
sergey@Test-PC: ~$

```

Рис. 24.5. Каталог расположения юнитов

Изменить текущую цель можно так:

```
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`.

Посмотрим, как можно определить цель, которая загружается по умолчанию:

```
systemctl get-default
```

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 или 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 ОС.

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 *юниты* (от англ. *unit* — единица) — 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 *юнит 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*, который 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 необязательный параметр *опции* (табл. 25.1).

Таблица 25.1. Описание опций команды `systemctl`

Сокращенный вариант	Полный вариант	Описание
-a	--all	Отображает все запущенные службы, включая остановленные
-q	--quiet	Выводит минимальную информацию по службам
-t	--type= <i>ТИП</i>	Отображает службы в соответствии с заданным типом
-l	--full	Отображает полное имя единицы
	--runtime	Не сохраняет вносимые изменения после перезагрузки системы
	--no-pager	Отключает постраничный вывод информации
	--state= <i>СОСТОЯНИЕ</i>	Сортирует выводимый список служб по заданному состоянию: <ul style="list-style-type: none"> <li>• <code>LOAD</code> — отображение только загруженных служб;</li> <li>• <code>SUB</code> — отображение состояния активности единицы;</li> <li>• <code>ACTIVE</code> — отображение активных служб</li> </ul>
-s	--signal= <i>СИГНАЛ</i>	Отправляет службе заданный сигнал

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. Команды программы `systemctl`

Команда	Описание
<code>list-units</code>	Выводит список активных юнитов
<code>list-sockets</code>	Выводит список загруженных сокетов, упорядоченных по адресу
<code>start NAME</code>	Запускает службу, имя которой передано в аргументе <i>NAME</i>
<code>stop NAME</code>	Останавливает службу, имя которой передано в аргументе <i>NAME</i>
<code>reload NAME</code>	Обновляет конфигурации службы, имя которой передано в аргументе <i>NAME</i>
<code>restart NAME</code>	Перезапускает службу, имя которой передано в аргументе <i>NAME</i>
<code>try-restart NAME</code>	Перезапускает службу, имя которой передано в аргументе <i>NAME</i> , при условии, что указанная служба активна в данный момент времени

Таблица 25.2 (окончание)

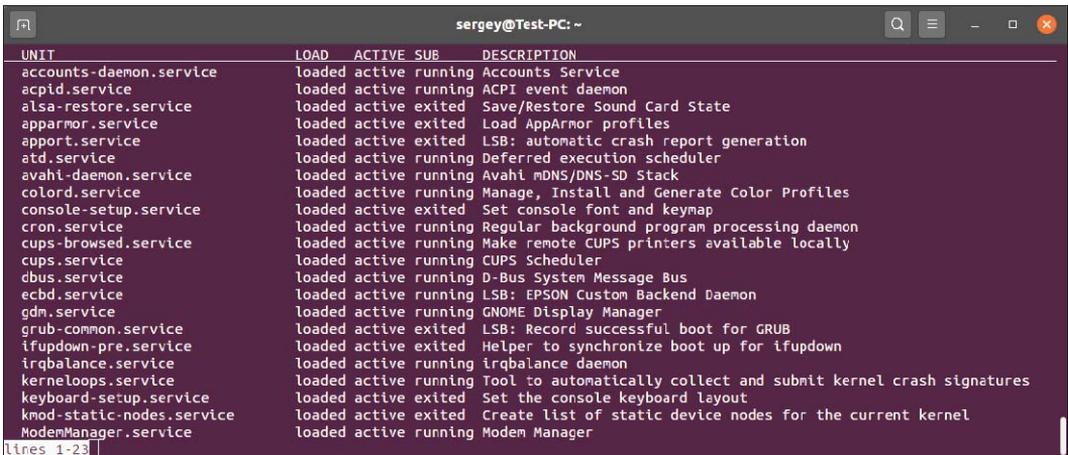
Команда	Описание
<code>reload-or-restart NAME</code>	Обновляет конфигурацию службы, а затем перезапускает ее. В случае если обновление конфигурации не предусмотрено, выполняется только перезапуск службы
<code>isolate NAME</code>	Запускает службу, имя которой передано в аргументе <i>NAME</i> , а все остальные останавливает
<code>kill NAME</code>	Отправляет службе сигнал завершения работы, имя которой передано в аргументе <i>NAME</i>
<code>is-active PATTERN</code>	Проверяет, запущена ли указанная служба, имя которой передано в аргументе <i>PATTERN</i>
<code>is-failed PATTERN</code>	Проверяет, не завершилась ли аварийно служба, имя которой передано в аргументе <i>PATTERN</i>
<code>status [PATTERN PID]</code>	Отображает 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>	Отображает параметры управления службой, имя которой передано в аргументе <i>PATTERN</i> , либо отображает параметры работающих служб
<code>reset-failed [PATTERN]</code>	Перезапускает службы, завершившиеся с ошибкой, имена которых переданы в аргументе <i>PATTERN</i>
<code>list-dependencies [NAME]</code>	Отображает зависимости службы, имя которой передано в аргументе <i>NAME</i>
<code>list-unit-files [PATTERN]</code>	Отображает все файлы, установленные указанной службой
<code>enable NAME</code>	Активирует 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>	Отключает запуск службы, а затем снова включает
<code>is-enabled NAME</code>	Отображает, 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 настало время применить полученные знания на практике. Первым делом отобразим список запущенных служб:

```
systemctl list-units --type=service
```

Данная конструкция позволит вывести список служб в удобном для восприятия виде (рис. 25.1).



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	Load AppArmor profiles
apport.service	loaded	active	exited	LSB: automatic crash report generation
atd.service	loaded	active	running	Deferred execution scheduler
avahi-daemon.service	loaded	active	running	Avahi mDNS/DNS-SD Stack
colord.service	loaded	active	running	Manage, Install and Generate Color Profiles
console-setup.service	loaded	active	exited	Set console font and keymap
cron.service	loaded	active	running	Regular background program processing daemon
cups-browsed.service	loaded	active	running	Make remote CUPS printers available locally
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 Daemon
gdm.service	loaded	active	running	GNOME Display Manager
grub-common.service	loaded	active	exited	LSB: Record successful boot for GRUB
ifupdown-pre.service	loaded	active	exited	Helper to synchronize boot up for ifupdown
irqbalance.service	loaded	active	running	irqbalance daemon
kerneloops.service	loaded	active	running	Tool to automatically collect and submit kernel crash signatures
keyboard-setup.service	loaded	active	exited	Set the console keyboard layout
knod-static-nodes.service	loaded	active	exited	Create list of static device nodes for the current kernel
ModemManager.service	loaded	active	running	Modem Manager

Рис. 25.1. Список запущенных служб

В полученном результате пять 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 можно получить список 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 посмотреть ее текущее состояние:

```
sudo systemctl status cups.service
```

В ответ на это программа `systemctl` отобразит не только текущее состояние службы, но 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: ~
sergey@Test-PC:~$ sudo systemctl status cups.service
● cups.service - CUPS Scheduler
   Loaded: loaded (/lib/systemd/system/cups.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2020-06-09 15:56:04 +06; 9s ago
 TriggeredBy: ● cups.socket
               ● cups.path
   Docs: man:cupsd(8)
  Main PID: 4936 (cupsd)
    Tasks: 2 (limit: 4015)
   Memory: 1.8M
   CGroup: /system.slice/cups.service
           └─4936 /usr/sbin/cupsd -l

июн 09 15:56:04 Test-PC systemd[1]: Started CUPS Scheduler.
sergey@Test-PC:~$
```

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 четыре состояния:

- ◆ `enabled` — служба находится в автозагрузке и будет загружена при старте операционной системы;
- ◆ `disabled` — служба не находится в автозагрузке и не будет запущена при старте операционной системы, однако она может быть запущена вручную;

- ◆ `masked` — служба не находится в автозагрузке и не будет запущена при старте операционной системы, однако запустить вручную ее не удастся;
- ◆ `static` — 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
```

Эта 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, которая 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 этот 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 строку 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* для управления питанием компьютера (табл. 25.3).

**Таблица 25.3.** Параметры управления питанием командой `systemctl`

Команда	Описание
<code>is-system-running</code>	Проверяет, запущены ли все модули и службы ОС
<code>default</code>	Загружает ОС с целью по умолчанию
<code>rescue</code>	Загружает режим, предназначенный для восстановления работоспособности ОС

Таблица 25.3 (окончание)

Команда	Описание
<code>emergency</code>	Загружает аварийную оболочку ОС для восстановления ее работоспособности
<code>halt</code>	Останавливает работу ОС и выключает компьютер
<code>poweroff</code>	Завершает работу ОС и выключает питание компьютера
<code>reboot</code>	Выполняет перезагрузку ОС
<code>kexec</code>	Завершает работу ОС и выполняет ее загрузку с использованием механизма ядра <code>kexec</code> (загрузка ядра ОС с пропуском процедуры аппаратной инициализации)
<code>exit</code>	Выполняет выход из пользовательской учетной записи и останавливает работу системы
<code>switch-root</code> <i>файловая_система</i>	Сменяет корневую файловую систему на файловую систему, указанную в параметре <i>файловая_система</i>
<code>suspend</code>	Переводит систему в ждущий режим
<code>hibernate</code>	Переводит систему в спящий режим
<code>hybrid-sleep</code>	Переводит 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` — предназначен для выбора действия при нажатии кнопки питания;
- ◆ `HandleSuspendKey` — предназначен для выбора действия при нажатии кнопки ждущего режима;
- ◆ `HandleHibernateKey` — предназначен для выбора действия при нажатии кнопки спящего режима;
- ◆ `HandleLidSwitch` — предназначен для выбора действия при закрытии крышки ноутбука.

В качестве значений могут быть использованы следующие параметры:

- ◆ `poweroff` — завершает работу операционной системы и выключает питание;
- ◆ `suspend` — переводит ОС в ждущий режим;
- ◆ `hibernate` — переводит ОС в спящий режим;
- ◆ `ignore` — игнорирует событие;
- ◆ `reboot` — выполняет перезагрузку ОС;
- ◆ `halt` — завершает работу ОС;
- ◆ `lock` — блокирует 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 суперпользователя `root`. Для этого выполните команду `sudo gedit /etc/systemd/logind.conf`.

Для вступления в силу внесенных вами изменений выполните команду:

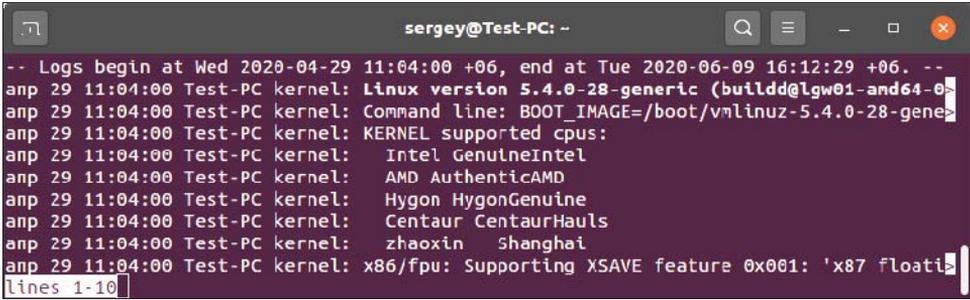
```
systemctl restart systemd-logind
```

## **Ведение журналов 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 их ведения 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`.



```

sergey@Test-PC: ~
-- Logs begin at Wed 2020-04-29 11:04:00 +06, end at Tue 2020-06-09 16:12:29 +06. --
apr 29 11:04:00 Test-PC kernel: Linux version 5.4.0-28-generic (build@lgw01-and64-05
apr 29 11:04:00 Test-PC kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-5.4.0-28-gene
apr 29 11:04:00 Test-PC kernel: KERNEL supported cpus:
apr 29 11:04:00 Test-PC kernel: Intel GenuineIntel
apr 29 11:04:00 Test-PC kernel: AMD AuthenticAMD
apr 29 11:04:00 Test-PC kernel: Hygon HygonGenuine
apr 29 11:04:00 Test-PC kernel: Centaur CentaurHauls
apr 29 11:04:00 Test-PC kernel: zhaoxin Shanghai
apr 29 11:04:00 Test-PC kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floati
lines 1-10

```

Рис. 25.3. Просмотр логов с помощью команды `journalctl`

Для просмотра логов используется команда `journalctl` (рис. 25.3).

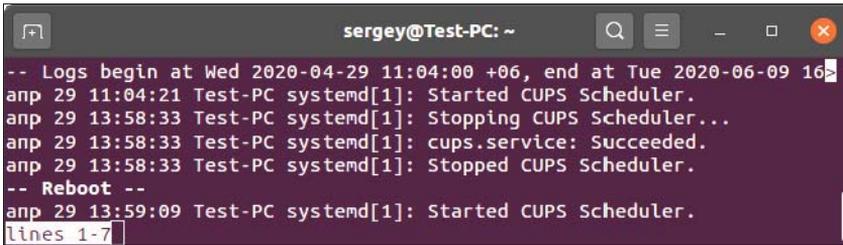
Как вы заметили, если ввести команду `journalctl` без аргументов, она выдаст огромное количество записей. Читать такой список несколько затруднительно. Для вывода записей конкретного юнита, например `cups`, выполните следующую команду:

```
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 Wed 2020-04-29 11:04:00 +06, end at Tue 2020-06-09 16:12:29 +06. --
apr 29 11:04:21 Test-PC systemd[1]: Started CUPS Scheduler.
apr 29 13:58:33 Test-PC systemd[1]: Stopping CUPS Scheduler...
apr 29 13:58:33 Test-PC systemd[1]: cups.service: Succeeded.
apr 29 13:58:33 Test-PC systemd[1]: Stopped CUPS Scheduler.
-- Reboot --
apr 29 13:59:09 Test-PC systemd[1]: Started CUPS Scheduler.
lines 1-7

```

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`, а 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 наконец 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 ОС.

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 ОС. 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 ОС 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).

```

top - 16:20:05 up 15 min, 1 user, load average: 0,43, 0,25, 0,27
Tasks: 195 total, 3 running, 192 sleeping, 0 stopped, 0 zombie
%Cpu(s): 33,3 us, 11,1 sy, 0,0 ni, 55,6 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
Миб Mem : 3901,3 total, 1925,8 free, 693,1 used, 1282,4 buff/cache
Миб Swap: 4500,0 total, 4500,0 free, 0,0 used. 2833,5 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 1463 sergey    20   0  770100 52884 28392 R  60,0   1,3   0:15.21 Xorg
 1701 sergey    20   0 4025856 232552 80056 S  20,0   5,8   0:20.58 gnome-shell
 3466 sergey    20   0  967888 51476 38688 R  20,0   1,3   0:00.89 gnome-terminal-
 3484 sergey    20   0  20528  3880  3372 R  20,0   0,1   0:00.27 top
   1 root      20   0  102264 11852  8580 S   0,0   0,3   0:01.70 systemd
   2 root      20   0   0       0     0 S   0,0   0,0   0:00.00 kthreadd
   3 root      0 -20   0       0     0 I   0,0   0,0   0:00.00 rcu_gp
   4 root      0 -20   0       0     0 I   0,0   0,0   0:00.00 rcu_par_gp
   6 root      0 -20   0       0     0 I   0,0   0,0   0:00.00 kworker/0:0H-kblo+
   7 root      20   0   0       0     0 I   0,0   0,0   0:00.48 kworker/u8:0-phy0
   8 root      0 -20   0       0     0 I   0,0   0,0   0:00.00 mn_percpu_wq
   9 root      20   0   0       0     0 S   0,0   0,0   0:00.13 ksoftirqd/0
  10 root      20   0   0       0     0 I   0,0   0,0   0:00.35 rcu_sched
  11 root      rt   0   0       0     0 S   0,0   0,0   0:00.00 migration/0

```

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:

- ◆ 16:20:05 — текущее системное время;
- ◆ up 15 min — общее время работы системы, которое отсчитывается с момента запуска операционной системы;
- ◆ 1 user — количество открытых пользовательских сессий. В данном случае на компьютере работает один пользователь;
- ◆ load average: 0,43, 0,25, 0,27 — 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:

- ◆ 195 `total` — общее количество запущенных процессов;
- ◆ 3 `running` — общее количество работающих в данный момент процессов;
- ◆ 192 `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:

- ◆ 33,3 `us` — процент использования пользовательскими процессами ресурсов центрального процессора;
- ◆ 11,1 `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";
- ◆ 55,5 `id` — процент времени, когда центральный процессор простаивает без нагрузки;
- ◆ 0,0 `wa` — процент использования ресурсов процессора процессами, которые ожидают выполнения операций;
- ◆ 0,0 `hi` — процент использования ресурсов центрального процессора обработчиками аппаратных прерываний (IRQ);
- ◆ 0,0 `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:

- ◆ 3901,3 `total` — общее количество памяти (физической в 4-й строке и виртуальной в 5-й строке);
- ◆ 31925,8 `free` — количество свободной памяти (отдельно физической и отдельно файла подкачки);
- ◆ 693,1 `used` — количество используемой памяти (отдельно физической и отдельно файла подкачки);
- ◆ 1282,4 `buff/cache` — количество используемой физической памяти в кэше буферов;
- ◆ 2833,5 `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, разобравшись 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).

```

sergey@Test-PC: ~
top - 16:32:35 up 27 min, 1 user, load average: 2,05, 1,00, 0,53
Tasks: 214 total, 1 running, 213 sleeping, 0 stopped, 0 zombie
%Cpu(s): 24,2 us, 9,1 sy, 0,0 ni, 66,6 id, 6,1 wa, 0,0 hi, 0,0 si, 0,0 st
Миб Mem : 3901,3 total, 1086,6 free, 1073,1 used, 1741,6 buff/cache
Миб Swap: 4500,0 total, 4500,0 free, 0,0 used. 2232,7 avail Mem
PID to signal/kill [default pid = 3940] 3940
  PID USER  PR  NI  VIRT  RES  SHR  S  %CPU  %MEM  TIME+  COMMAND
 3940 sergey  20   0 4824672 280328 142912 S 37,5  7,0  0:51.32 chrome
 3077 sergey  20   0 776784 145096 63520 S 18,8  3,6  0:17.01 chrome
 1405 sergey   9 -11 2392548 20264 15880 S 12,5  0,5  0:03.61 pulseaudio
 1463 sergey  20   0 769560 54732 30240 S  6,2  1,4  0:23.17 Xorg
 4090 sergey  20   0 20528 3964 3408 R  6,2  0,1  0:00.01 top
   1 root    20   0 102264 11852 8580 S  0,0  0,3  0:01.83 systemd
   2 root    20   0 0 0 0 S  0,0  0,0  0:00.00 kthreadd
   3 root    0 -20 0 0 0 I  0,0  0,0  0:00.00 rcu_gp
   4 root    0 -20 0 0 0 I  0,0  0,0  0:00.00 rcu_par_gp
   6 root    0 -20 0 0 0 I  0,0  0,0  0:00.00 kworker/0:0H-kblockd
  
```

Рис. 26.2. Завершение работы процесса с PID, равным 3940

После этого нажмите клавишу `<Enter>` и подтвердите остановку процесса, нажав клавишу `<Enter>` (рис. 26.3).

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.

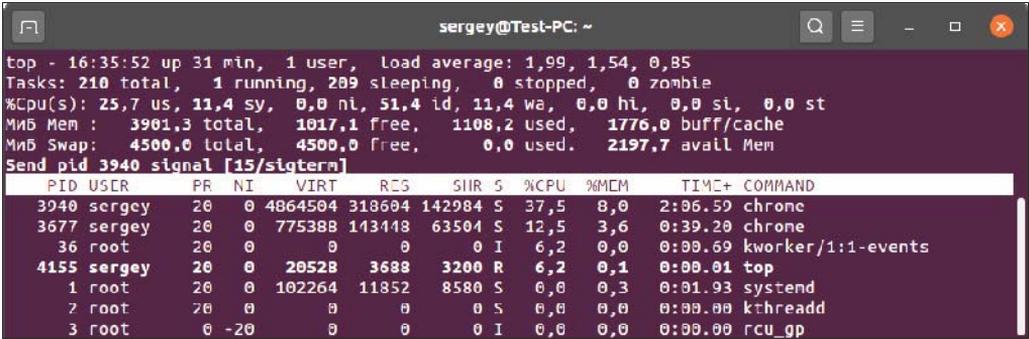


Рис. 26.3. Программа top требует подтверждения завершения работы процесса

**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
-A	Отображает все работающие процессы
-a	Отображает только те процессы, которые связаны с данным Терминалом, за исключением системных процессов сеанса
-d	Отображает все процессы, кроме главных процессов сеанса
-o	Задаёт отображаемые в выводе столбцы

Таблица 26.1 (окончание)

Параметр	Описание
-T	Отображает все процессы, которые были запущены в текущем Терминале
-r	Выводит информацию только о работающих в данный момент процессах
-u	Выводит информацию о процессах в удобном для восприятия виде
-x	Отображает 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      [rcu_gp]
    4 root      [rcu_par_gp]

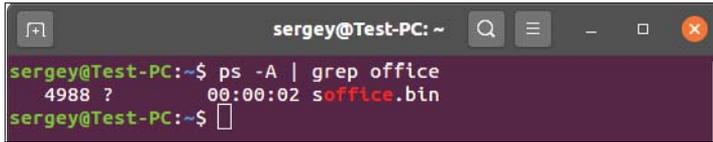
```

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).



```
sergey@Test-PC:~$ ps -A | grep office
4988 ? 00:00:02 soffice.bin
sergey@Test-PC:~$
```

Рис. 26.5. Вывод информации о процессе по искомому слову

Получив идентификатор искомого процесса, нужно завершить процесс следующей командой:

```
kill 4988
```

О том, откуда взялась команда *kill*, мы сейчас и поговорим.

## Завершение процесса командой *kill*

Команда *kill* (от англ. *kill* — убить) предназначена 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 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 встречает 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).

```

sergey@Test-PC:~$ kill -l
1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
6) SIGABRT    7) SIGBUS     8) SIGFPE     9) SIGKILL     10) SIGUSR1
11) SIGSEGV   12) SIGUSR2   13) SIGPIPE   14) SIGALRM    15) SIGTERM
16) SIGSTKFLT 17) SIGCHLD  18) SIGCONT   19) SIGSTOP    20) SIGTSTP
21) SIGTTIN   22) SIGTTOU  23) SIGURG    24) SIGXCPU   25) SIGXFSZ
26) SIGVTALRM 27) SIGPROF  28) SIGWINCH  29) SIGIO      30) SIGPWR
31) SIGSYS    34) SIGRTMIN 35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
sergey@Test-PC:~$

```

Рис. 26.6. Список сигналов команды `kill`

Как видим, сигнал `SIGSTOP` имеет номер 19, а сигнал `SIGKILL` номер 9. Это означает, что вместо 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.

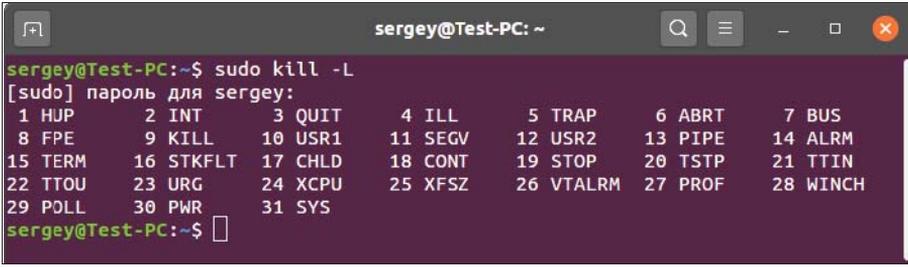
Таблица 26.2. Опции команды `kill`

Сокращенный вариант	Полный вариант	Описание
<code>-&lt;сигнал&gt;</code> <code>-s &lt;сигнал&gt;</code>	<code>--signal &lt;сигнал&gt;</code>	Задаёт имя или номер посылаемого сигнала
<code>-l</code>	<code>--list [сигнал]</code>	Выводит имена сигналов (имя сигнала), соответствующие номеру или имени, переданному в параметре <code>сигнал</code>
<code>-L</code>	<code>--table</code>	Выводит информацию о сигналах в виде таблицы

Если команде `kill` не переданы номер или имя сигнала, по умолчанию считается, что задан сигнал `SIGTERM`, т. е. номер 15. Это означает, что в случае выполнения команды `kill 4988` процессу будет послан сигнал `SIGTERM`, означающий завершение работы процесса им же самим.

Попробуем выполнить следующую команду:

```
sudo kill -L
```



```

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 TGIN
22 TTOU   23 URG    24 XCPU   25 XFSZ    26 VTALRM  27 PROF    28 WINCH
29 POLL   30 PWR    31 SYS
sergey@Test-PC:~$

```

Рис. 26.7. Выполнение команды с root-правами

В ответ на это Терминал отобразит список сигналов, очень похожий на вывод команды `kill -l` (рис. 26.7).

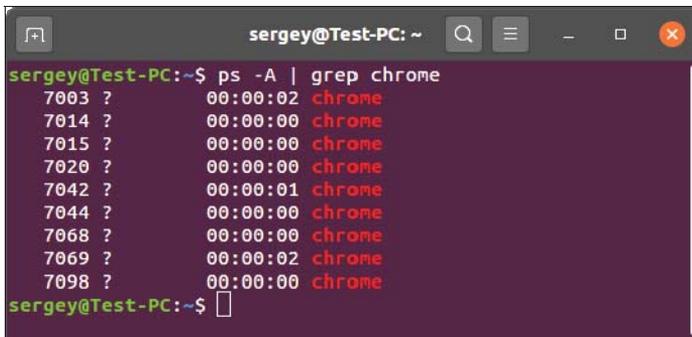
Команда `sudo kill -L` выводит основной список доступных команд без префикса SIG.

Обратите внимание, что для выполнения данной команды необходимы root-права. Если вы попытаетесь выполнить эту команду без соответствующих прав, то получите полный список сигналов, как при выполнении команды `kill -l` (см. рис. 26.6).

В большинстве случаев использование дополнительных опций не требуется. Для остановки процесса достаточно указать его PID, и процесс будет остановлен.

## Завершение процесса командой *killall*

Команда `kill` 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. 26.8)?



```

sergey@Test-PC: ~
sergey@Test-PC:~$ ps -A | grep chrome
7003 ?        00:00:02 chrome
7014 ?        00:00:00 chrome
7015 ?        00:00:00 chrome
7020 ?        00:00:00 chrome
7042 ?        00:00:01 chrome
7044 ?        00:00:00 chrome
7068 ?        00:00:00 chrome
7069 ?        00:00:02 chrome
7098 ?        00:00:00 chrome
sergey@Test-PC:~$

```

Рис. 26.8. 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	Учитывает полные совпадения длинных имен
-I	--ignore-case	Игнорирует регистр написания имени процесса
-g	--process-group	Завершает работу группы процессов
-y	--younger-than <время>	Завершает работу процессов, новее заданного времени
-o	--older-than <время>	Завершает работу процессов, запуск которых произошел раньше указанного времени
-i	--interactive	Запрашивает подтверждение перед завершением процессов
-l	--list	Выводит список имен допустимых сигналов
-q	--quiet	Не выводит подробный отчет
-r	--regex <ИМЯ>	Рассматривает заданное имя как регулярное выражение
-s	--signal <СИГНАЛ>	Направляет процессу указанный сигнал, а не его имя
-u	--user <ПОЛЬЗОВАТЕЛЬ>	Завершает процессы указанного пользователя
-v	--verbose	Выводит ответ в случае успешной обработки процессом отправленного сигнала
-w	--wait	Позволяет дождаться завершения работы процесса
-Z	--context <ИМЯ>	Завершает 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 (рис. 26.9).

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
```

```

sergey@Test-PC: ~
sergey@Test-PC:~$ ps -A | grep chrome
7003 ?      00:00:02 chrome
7014 ?      00:00:00 chrome
7015 ?      00:00:00 chrome
7020 ?      00:00:00 chrome
7042 ?      00:00:01 chrome
7044 ?      00:00:00 chrome
7068 ?      00:00:00 chrome
7069 ?      00:00:02 chrome
7098 ?      00:00:00 chrome
sergey@Test-PC:~$ killall chrome
sergey@Test-PC:~$ ps -A | grep chrome
sergey@Test-PC:~$

```

Рис. 26.9. Процессы с именем `chrome` были завершены

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 перезапуска первой виртуальной консоли и входа в графический режим;
- ◆ `<Ctrl>+<Alt>+<F2>` — для возврата в первую виртуальную консоль в графическом режиме;
- ◆ `<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.

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 gdm
```

После этого дисплейный менеджер GDM (GNOME Display Manager) перезапустится, и вы сможете продолжить работу в графическом режиме, минуя перезагрузку

ОС. К этому методу перезагрузки графического окружения стоит прибегать в случае наблюдаемых сбоев в его работе либо невозможности продолжить работу в ОС.

## 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).

The screenshot shows the htop terminal window with the following content:

```

1  [|||||]          14.8%]   Tasks: 112, 267 thr; 1 running
2  [|||||]          16.4%]   Load average: 0.87 0.95 0.71
Mem[|||||]          927M/3.81G] Uptime: 02:23:39
Swp[|]              0K/4.39G]

  PID USER      PRI  NI  VIRT   RES   SHR  S CPU% MEM%   TIME+  Command
12401 sergey    20   0 3684M 228M 87016 S 12.0  5.9  0:12.42 /usr/bin/gnome-
12282 sergey    20   0 757M 53528 28828 S 10.7  1.3  0:11.13 /usr/lib/xorg/X
12688 sergey    20   0 1064M 90508 50632 S  4.0  2.3  0:08.67 /snap/snap-stor
16965 sergey    20   0 19260  4040  3336 R  2.0  0.1  0:00.10 htop
16867 sergey    20   0 945M 51076 38420 S  0.7  1.3  0:00.48 /usr/libexec/gn
12296 sergey    20   0 757M 53528 28828 S  0.7  1.3  0:00.19 /usr/lib/xorg/X
  1 root      20   0 163M 11704  8324 S  0.0  0.3  0:04.94 /sbin/init spla
 303 root      19  -1 143M 81688 79628 S  0.0  2.0  0:10.47 /lib/systemd/sy
 365 root      20   0 24364  7936  3936 S  0.0  0.2  0:04.16 /lib/systemd/sy
 637 systemd-r 20   0 24176 13508  9364 S  0.0  0.3  0:00.93 /lib/systemd/sy
 704 systemd-t 20   0 90412  6344  5468 S  0.0  0.2  0:00.00 /lib/systemd/sy
 640 systemd-t 20   0 90412  6344  5468 S  0.0  0.2  0:00.55 /lib/systemd/sy
 743 root      20   0 241M  7904  6824 S  0.0  0.2  0:00.26 /usr/lib/accoun
 807 root      20   0 241M  7904  6824 S  0.0  0.2  0:00.10 /usr/lib/accoun
 713 root      20   0 241M  7904  6824 S  0.0  0.2  0:00.49 /usr/lib/accoun
 714 root      20   0 2540  712  648 S  0.0  0.0  0:00.09 /usr/sbin/acpid
F1Help  F2Setup  F3Search F4Filter F5Tree   F6SortBy F7Nice   F8Nice + F9Kill  F10Quit

```

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 который можно из меню запуска приложений, 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
```

Название процесса	Пользователь	% ЦП	ID	Память	Суммарное ч	Суммар
at-spi2-registryd	sergey	0	12461	636,0 KiB	Н/Д	
at-spi-bus-launcher	sergey	0	12365	584,0 KiB	Н/Д	
bash	sergey	0	15621	1,4 MiB	8,0 KiB	
dbus-daemon	sergey	0	11883	2,2 MiB	Н/Д	
dbus-daemon	sergey	0	12370	464,0 KiB	Н/Д	
dconf-service	sergey	0	12444	776,0 KiB	Н/Д	64,0
evolution-addressbook-factory	sergey	0	12689	3,5 MiB	Н/Д	36,0
evolution-alarm-notify	sergey	0	12572	15,4 MiB	Н/Д	
evolution-calendar-factory	sergey	0	12606	3,7 MiB	Н/Д	
evolution-source-registry	sergey	0	12474	3,8 MiB	Н/Д	
gdm-x-session	sergey	0	12280	652,0 KiB	Н/Д	
gjs	sergey	0	12522	4,9 MiB	Н/Д	
gnome-calculator-search-provi	sergey	0	17163	9,1 MiB	Н/Д	
gnome-calendar	sergey	0	16856	14,6 MiB	1,4 MiB	
gnome-control-center-search-p	sergey	0	17161	9,1 MiB	Н/Д	
gnome-keyring-daemon	sergey	0	12269	1000,0 KiB	Н/Д	
gnome-session-binary	sergey	0	12299	1,5 MiB	4,0 KiB	
gnome-session-binary	sergey	0	12387	2,7 MiB	Н/Д	4,0
gnome-session-ctl	sergey	0	12380	420,0 KiB	Н/Д	
gnome-shell	sergey	5	12401	146,5 MiB	1009,0 KiB	148,0
gnome-shell-calendar-server	sergey	0	12470	2,4 MiB	Н/Д	

Рис. 26.11. Системный монитор

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.

По умолчанию Системный монитор отображает процессы текущего пользователя. Для того чтобы просмотреть процессы, требующие привилегий суперпользователя 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.

Все 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. с помощью Терминала. 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

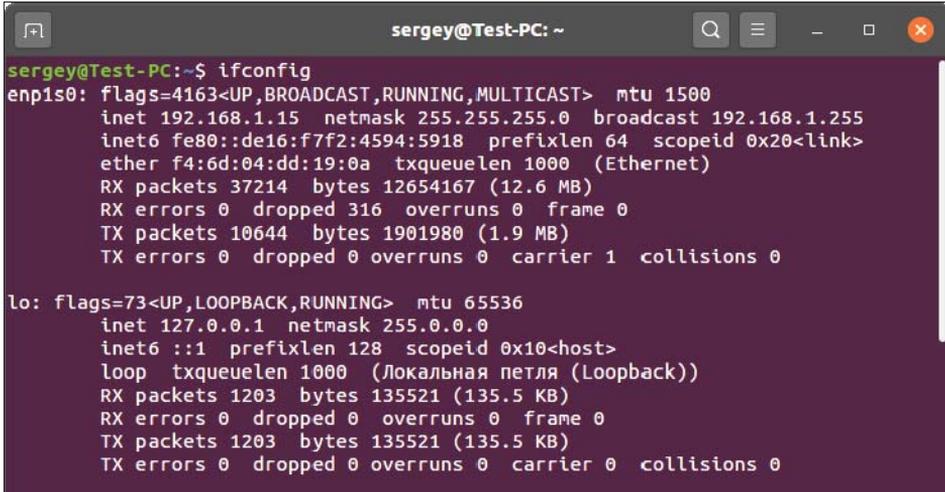
Первым делом необходимо убедиться в том, что сетевая карта вашего компьютера успешно определяется и работает в ОС Ubuntu. Для этого выполните следующую команду:

```
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, вы увидите примерно следующую картину (рис. 27.1).



```
sergey@Test-PC:~$ ifconfig
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.15 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::de16:f7f2:4594:5918 prefixlen 64 scopeid 0x20<link>
    ether f4:6d:04:dd:19:0a txqueuelen 1000 (Ethernet)
    RX packets 37214 bytes 12654167 (12.6 MB)
    RX errors 0 dropped 316 overruns 0 frame 0
    TX packets 10644 bytes 1901980 (1.9 MB)
    TX errors 0 dropped 0 overruns 0 carrier 1 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Локальная петля (Loopback))
    RX packets 1203 bytes 135521 (135.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1203 bytes 135521 (135.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Рис. 27.1. Обнаружена сетевая карта

Команда `ifconfig` показала два сетевых интерфейса: `enp1s0` 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 192...`" 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.

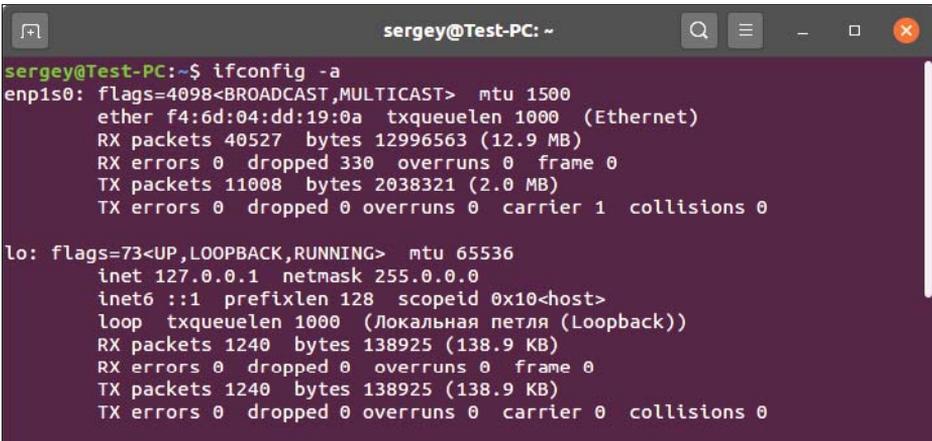
**ПРИМЕЧАНИЕ**

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 `enp1s0` (рис. 27.2).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ ifconfig -a  
enp1s0: flags=4098<BROADCAST,MULTICAST> mtu 1500  
ether f4:6d:04:dd:19:0a txqueuelen 1000 (Ethernet)  
RX packets 40527 bytes 12996563 (12.9 MB)  
RX errors 0 dropped 330 overruns 0 frame 0  
TX packets 11008 bytes 2038321 (2.0 MB)  
TX errors 0 dropped 0 overruns 0 carrier 1 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
inet 127.0.0.1 netmask 255.0.0.0  
inet6 ::1 prefixlen 128 scopeid 0x10<host>  
loop txqueuelen 1000 (Локальная петля (Loopback))  
RX packets 1240 bytes 138925 (138.9 KB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 1240 bytes 138925 (138.9 KB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Рис. 27.2. Обнаружен отключенный сетевой интерфейс

Обратите внимание, что у нас отсутствует вторая строка "inet addr...", как было показано на рис. 27.1. Отсутствие этой строки говорит о том, что данный сетевой интерфейс не задействован. Давайте исправим это и задействуем его:

```
sudo ifconfig enp1s0 up
```

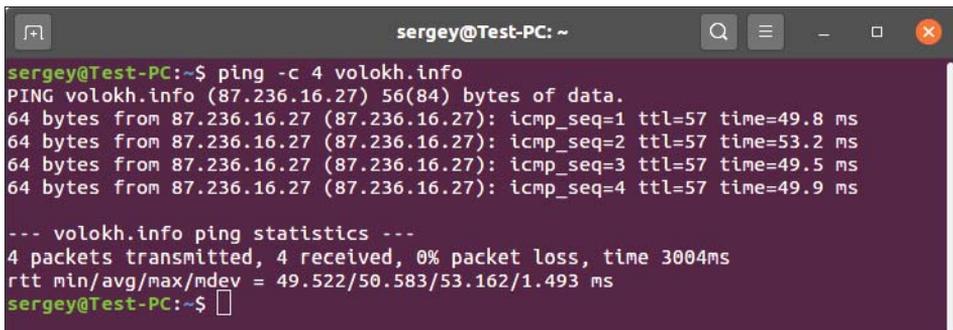
Теперь наш сетевой интерфейс `enp1s0` включен и работоспособность сети должна восстановиться.

## Проверка соединения с Интернетом

В том случае, когда все сетевые интерфейсы включены и настроены, а соединение с удаленным узлом все так же отсутствует, рекомендуется прибегнуть к команде `ping`, которая покажет, доступен удаленный узел или нет:

```
ping -c 4 volokh.info
```

При доступности удаленного узла (сайта) результат будет примерно таким, как представлено на рис. 27.3.



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ ping -c 4 volokh.info  
PING volokh.info (87.236.16.27) 56(84) bytes of data.  
64 bytes from 87.236.16.27 (87.236.16.27): icmp_seq=1 ttl=57 time=49.8 ms  
64 bytes from 87.236.16.27 (87.236.16.27): icmp_seq=2 ttl=57 time=53.2 ms  
64 bytes from 87.236.16.27 (87.236.16.27): icmp_seq=3 ttl=57 time=49.5 ms  
64 bytes from 87.236.16.27 (87.236.16.27): icmp_seq=4 ttl=57 time=49.9 ms  
  
--- volokh.info ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3004ms  
rtt min/avg/max/mdev = 49.522/50.583/53.162/1.493 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 87.236.16.27 (87.236.16.27): icmp_seq=1 ttl=57 time=49.8 ms
```

Разобьем этот ответ на составляющие:

- ◆ `64 bytes from 87.236.16.27 (87.236.16.27)` — получено 64 байта от узла с IP-адресом `87.236.16.27`;
- ◆ `icmp_seq=1` — номер отправленного пакета;
- ◆ `ttl=57` — время жизни полученного пакета;
- ◆ `time=49.8` — 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 3004ms
```

Данная строка сообщает, что было отправлено 4 пакета, из которых были получены все 4, ноль пакетов было потеряно, и на все это понадобилось 3004 миллисекунды.

В случае недоступности удаленного узла ответ был бы примерно таким, как в листинге 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 192.168.1.15 icmp_seq=1 Destination Host Unreachable
```

```
--- 192.168.1.5 ping statistics ---
```

```
4 packets transmitted, 0 received, +1 errors, 100% packet loss, time 3079ms  
pipe 4
```

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 удается 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 `tracert`.

Программа `tracert` — это служебная программа, не входящая в состав ОС Ubuntu Linux и предназначенная для определения маршрутов следования переданных данных в сети.

Для установки программы `tracert` 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 (5.255.255.5), 30 hops max, 60 byte packets
 1  _gateway (192.168.1.230)  0.270 ms  0.628 ms  0.620 ms
 2  10.254.253.253 (10.254.253.253)  1.410 ms  1.410 ms  1.399 ms
 3  * * *
 4  rt1.omkc.ru (217.25.208.157)  2.625 ms  2.528 ms  2.909 ms
 5  omk02.transtelecom.net (188.43.2.66)  2.902 ms  2.893 ms  2.883 ms
 6  mskn17ra-lo1.transtelecom.net (217.150.55.21)  33.691 ms  32.871 ms  32.790 ms
 7  Yandex-gw.transtelecom.net (188.43.3.213)  31.527 ms  30.554 ms  31.531 ms
 8  yandex.ru (5.255.255.5)  36.300 ms  vla-32z1-eth-trunk1-1.yndx.net
    (87.250.239.159)  34.663 ms  yandex.ru (5.255.255.5)  36.285 ms
```

В полученном результате трассировки до указанного узла мы можем увидеть 8 строк, которые называются *прыжками*.

#### **ПРИМЕЧАНИЕ**

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 до 8. Лишь только на 8-й цепочке пакет доходит до указанного узла.

Таким способом мы можем проследить путь 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).

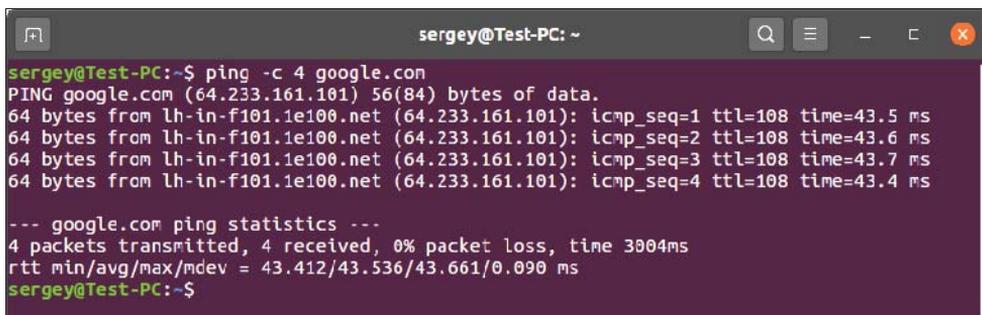
#### **Листинг 27.3. Трассировка узла google.com**

```
tracert to google.com (64.233.161.100), 30 hops max, 60 byte packets
 1  _gateway (192.168.1.230)  0.366 ms  0.723 ms  0.709 ms
 2  10.254.253.253 (10.254.253.253)  1.497 ms  1.486 ms  1.473 ms
 3  mx480.omkc.ru (217.25.208.197)  1.460 ms  1.447 ms  1.432 ms
 4  rt1.omkc.ru (217.25.208.157)  2.217 ms  2.204 ms  2.190 ms
 5  omk02.transtelecom.net (188.43.2.66)  2.577 ms  2.563 ms  2.953 ms
 6  mskn17ra-lo1.transtelecom.net (217.150.55.21)  32.771 ms  46.408 ms  33.206 ms
 7  google-gw.transtelecom.net (217.150.44.9)  31.522 ms  30.579 ms  31.473 ms
 8  108.170.250.146 (108.170.250.146)  32.246 ms 108.170.250.99
                                     (108.170.250.99)  33.446 ms  32.599 ms
 9  216.239.50.46 (216.239.50.46)  43.766 ms 216.239.50.44 (216.239.50.44)
                                     42.403 ms 216.239.50.2 (216.239.50.2)  42.782 ms
10  108.170.235.64 (108.170.235.64)  47.915 ms 172.253.65.82 (172.253.65.82)
                                     41.808 ms 172.253.66.110 (172.253.66.110)  44.025 ms
11  108.170.233.161 (108.170.233.161)  42.667 ms 142.250.56.127
                                     (142.250.56.127)  45.906 ms 216.239.62.9 (216.239.62.9)  41.232 ms
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
```

**ПРИМЕЧАНИЕ**

Полный список параметров команды `traceroute` вы найдете в справочной информации, выполнив команду `man traceroute`.

Как видите, пакет потерялся после 11-го прыжка. Символы астериска (звездочки) на 12–21 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).

A screenshot of a terminal window titled "sergey@Test-PC: ~". The terminal shows the execution of a ping command: "sergey@Test-PC:~\$ ping -c 4 google.com". The output displays four successful ping responses from the IP address 64.233.161.101, each with a TTL of 108 and a response time around 43.5 ms. Below the individual responses, a summary line reads: "--- google.com ping statistics ---". The statistics show that 4 packets were transmitted and received with 0% packet loss, and the round-trip time (rtt) is 3004ms. The terminal prompt "sergey@Test-PC:~\$" is visible at the bottom.

```
sergey@Test-PC:~$ ping -c 4 google.com
PING google.com (64.233.161.101) 56(84) bytes of data:
64 bytes from lh-in-f101.1e100.net (64.233.161.101): icmp_seq=1 ttl=108 time=43.5 ms
64 bytes from lh-in-f101.1e100.net (64.233.161.101): icmp_seq=2 ttl=108 time=43.6 ms
64 bytes from lh-in-f101.1e100.net (64.233.161.101): icmp_seq=3 ttl=108 time=43.7 ms
64 bytes from lh-in-f101.1e100.net (64.233.161.101): icmp_seq=4 ttl=108 time=43.4 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 43.412/43.536/43.661/0.090 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 с помощью закрытия неиспользуемых портов и отключения ненужных служб.

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, и 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 уже предустановлена в ОС Ubuntu. Если по какой-то причине данная программа у вас отсутствует, ее можно найти и установить как в Менеджере приложений Ubuntu, так с помощью следующей команды:

```
sudo apt install remmina
```

Установите Remmina (если она у вас отсутствует) и запустите данное приложение (рис. 28.1).

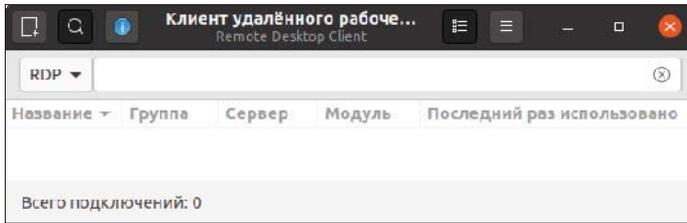


Рис. 28.1. Клиент удаленного рабочего стола Remmina

Для создания подключения к удаленному рабочему столу Windows нажмите кнопку **Новый профиль соединения** в левом верхнем углу окна. Откроется окно создания нового профиля (рис. 28.2).

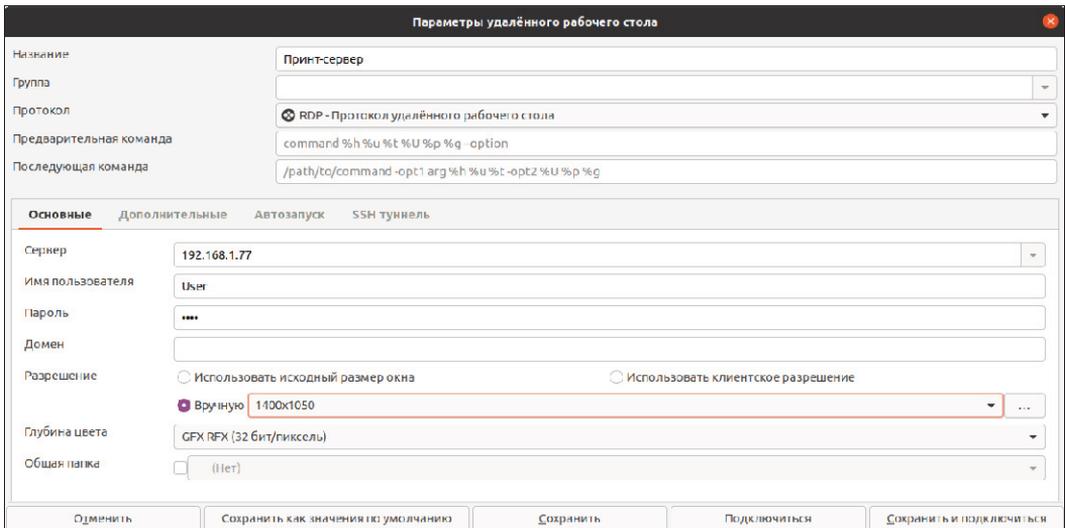


Рис. 28.2. Создание нового профиля для подключения

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 исходного размера окна. 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).

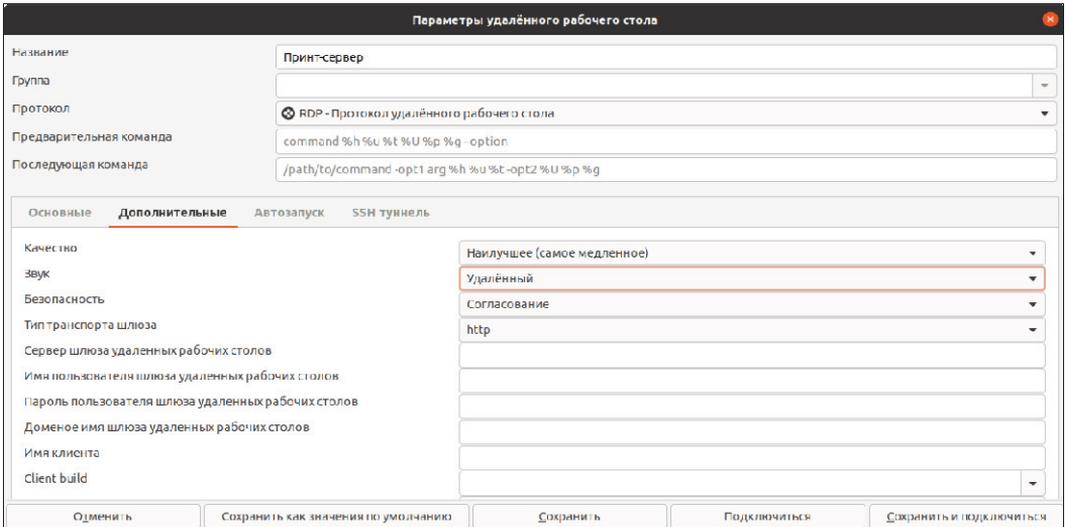


Рис. 28.3. Настройка параметров звука и отображения

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, рекомендуется настроить параметры шифрования передаваемых данных на вкладке **SSH туннель**. 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).

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).

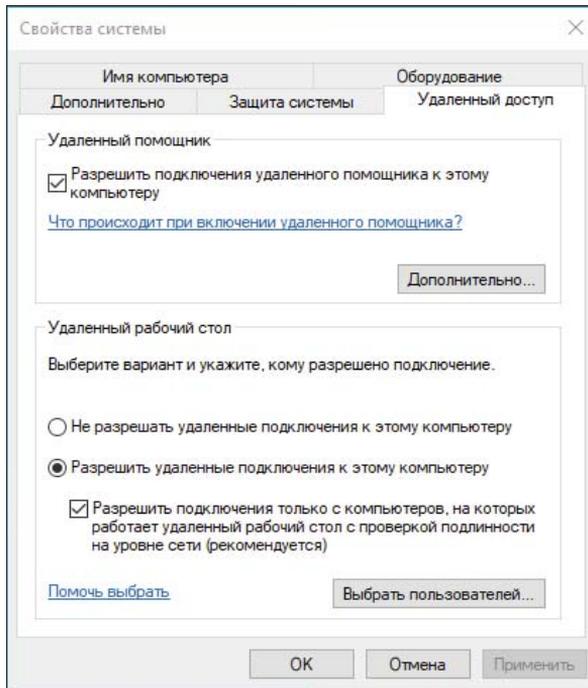


Рис. 28.4. Окно настроек удаленного доступа в ОС Windows 10

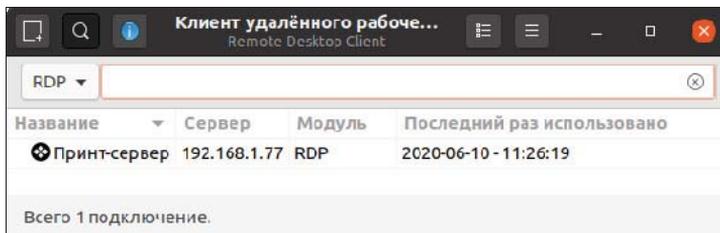


Рис. 28.5. Окно программы Remmina с созданным подключением

Для быстрого подключения просто два раза щелкните по созданному подключению, и удаленный рабочий стол будет открыт.

## Подключение с помощью AnyDesk

AnyDesk — это программа для подключения к удаленному рабочему столу, поддерживаемая множеством операционных систем, а также имеющая огромный функционал по управлению удаленным компьютером.

Программа доступна для загрузки на сайте разработчика <https://anydesk.com/>. Кроме того, вы можете установить TeamViewer с помощью следующей команды:

```
sudo apt install anydesk
```

Разумеется, сначала необходимо подключить репозиторий с данной программой:

```
wget -qO - https://keys.anydesk.com/repos/DEB-GPG-KEY | apt-key add -
echo "deb http://deb.anydesk.com/ all main" > /etc/apt/sources.list.d/anydesk-
stable.list
sudo apt update
```

Самым простым способом установки является загрузка deb-пакета с сайта разработчика.

После установки откроется главное окно программы AnyDesk (рис. 28.6).

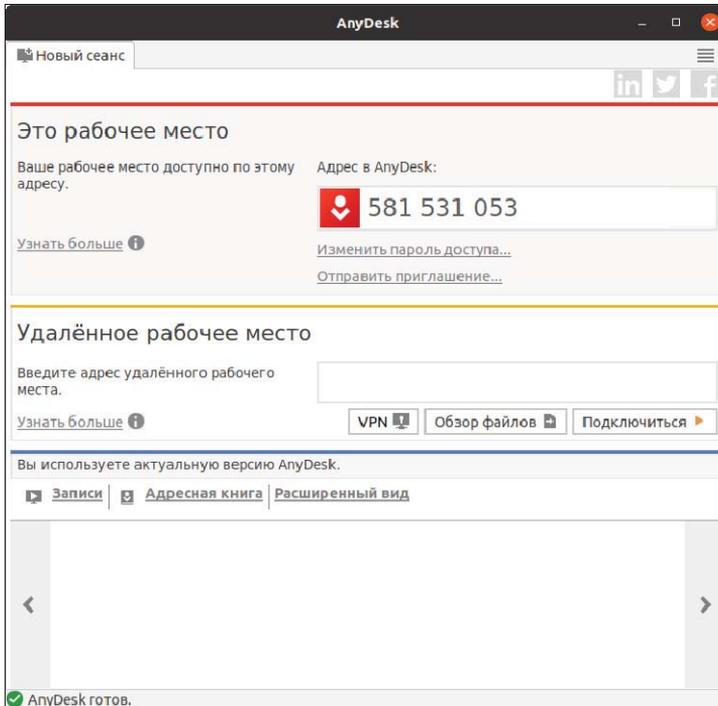


Рис. 28.6. Главное окно программы AnyDesk

Особенность работы AnyDesk 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 верхней части главного окна программы, нажимаете кнопку **Подключиться**, и удаленная сторона принимает входящее подключение.

#### **ПРИМЕЧАНИЕ**

Программа AnyDesk позволяет настроить неконтролируемый удаленный доступ. Это значит, что, введя эти данные, к вашему компьютеру можно будет получить удаленный доступ в любой удобный для вас момент времени. При этом на удаленном ком-

пьютере не требуется подтверждение на входящее подключение. Однако стоит помнить, что удаленный компьютер должен быть в это время включен, иначе вы не сможете к нему подключиться.

Программа AnyDesk бесплатна только для некоммерческого использования. Это означает, что вы вправе запускать ее только для личных целей. Для использования AnyDesk 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).

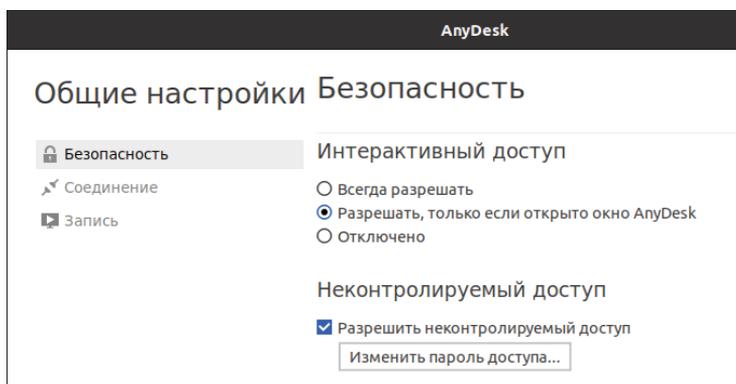


Рис. 28.7. Настройка неконтролируемого (постоянного) доступа

После этого вам не потребуется постоянно спрашивать ID и подтверждать доступ к удаленному компьютеру, а нужно будет только однажды ввести указанные данные при настройке неконтролируемого доступа в программе AnyDesk, и все последующие подключения будут выполняться автоматически (рис. 28.8).

Программа AnyDesk 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.

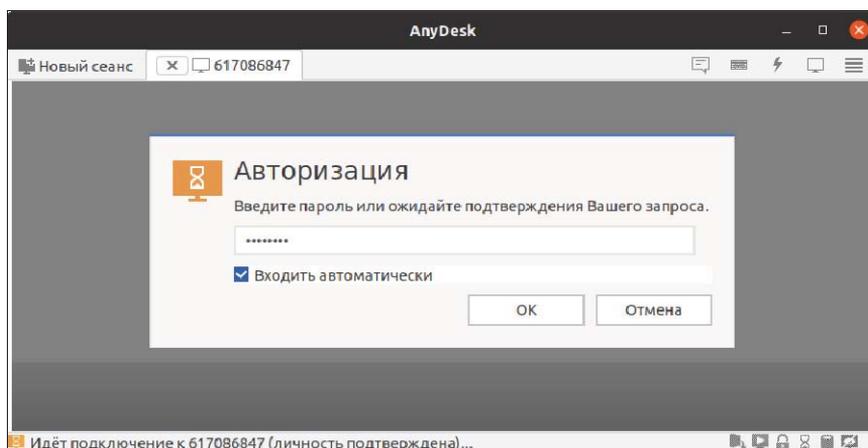


Рис. 28.8. Авторизация подключения по указанному паролю

## 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 AnyDesk.

Знания, полученные в этой главе, помогут в решении возникших проблем у неопытных пользователей, а также в том случае, когда нужно получить удаленный доступ к своему же компьютеру.



# 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 монополюно 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-режиме (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,

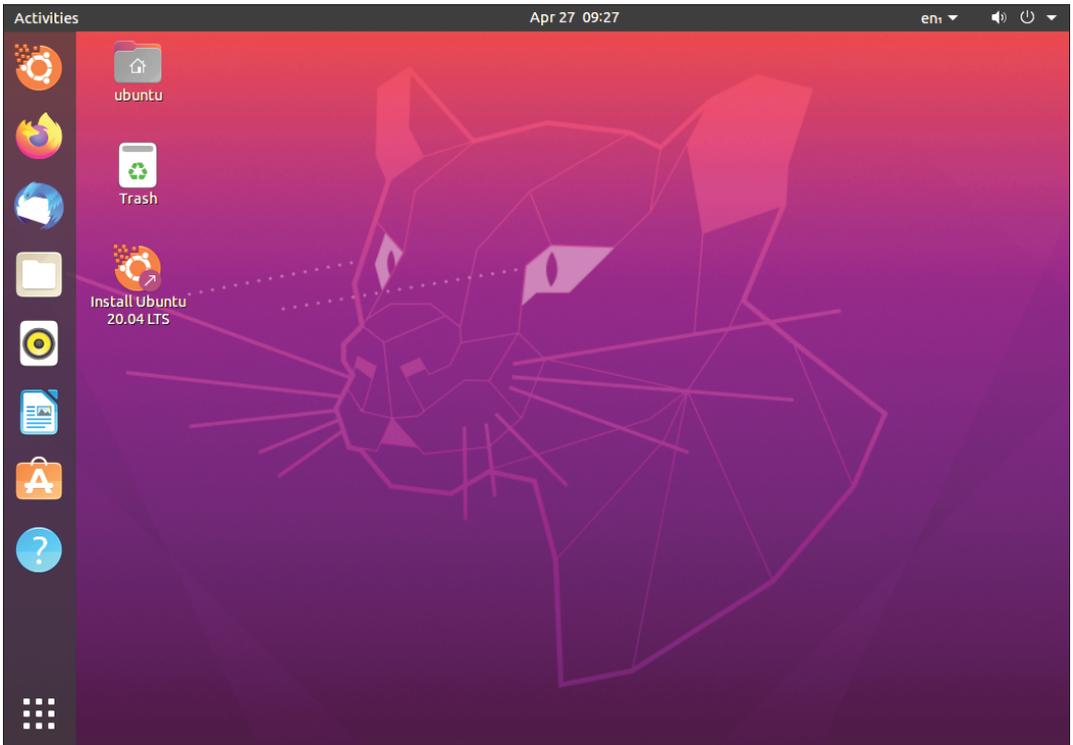


Рис. 29.1. ОС Ubuntu загрузилась в Live-режиме

```

ubuntu@ubuntu: ~
Тип метки диска: dos
Идентификатор диска: 0xc109c109

Устр-во   Загрузочный  начало      Конеч      Секторы  Размер  Идентификатор  Тип
/dev/sda1 *                2048        1187839    1185792   579M    7              HPFS/N
/dev/sda2                1187840    614402047  613214208 292,4G   7              HPFS/N
/dev/sda3                614402048  874373119  259971072 124G     7              HPFS/N
/dev/sda4                874373120  976773119  102400000 48,8G    5              Расшир
/dev/sda5                874375168  883591167  9216000    4,4G     82             Linux
/dev/sda6                883593216  945033215  61440000   29,3G    83             Linux
/dev/sda7                945035264  976773119  31737856   15,1G    83             Linux

Диск /dev/sdb: 7,22 GiB, 7747928064 байт, 15132672 секторов
Disk model: USB DISK 2.0
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
Тип метки диска: dos
Идентификатор диска: 0x1b425128

Устр-во   Загрузочный  начало      Конеч      Секторы  Размер  Идентификатор  Тип
/dev/sdb1 *                2048        15132671   15130624   7,2G    c W95 FAT32 (
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
```

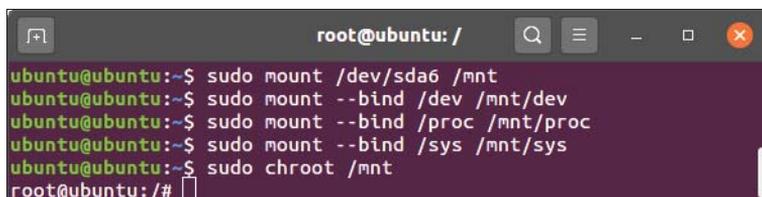
После монтирования корневого каталога нам осталось смонтировать еще несколько каталогов:

```
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).



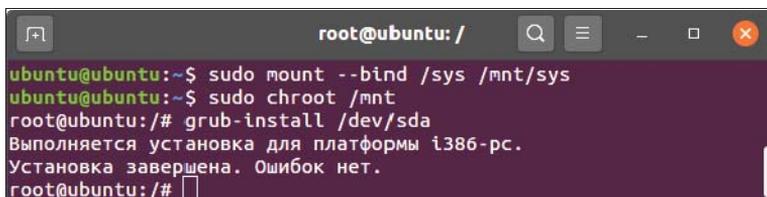
```
root@ubuntu: /
ubuntu@ubuntu:~$ sudo mount /dev/sda6 /mnt
ubuntu@ubuntu:~$ sudo mount --bind /dev /mnt/dev
ubuntu@ubuntu:~$ sudo mount --bind /proc /mnt/proc
ubuntu@ubuntu:~$ sudo mount --bind /sys /mnt/sys
ubuntu@ubuntu:~$ sudo chroot /mnt
root@ubuntu:/#
```

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).



```
root@ubuntu: /
ubuntu@ubuntu:~$ sudo mount --bind /sys /mnt/sys
ubuntu@ubuntu:~$ sudo chroot /mnt
root@ubuntu:/# grub-install /dev/sda
Выполняется установка для платформы i386-pc.
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 10 (рис. 29.5).

Теперь, после того как обновлен загрузчик, мы можем выйти из созданного временного окружения:

```
exit
```

```

root@ubuntu: /
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Генерируется файл настройки grub ...
Найден образ linux: /boot/vmlinuz-5.4.0-33-generic
Найден образ initrd: /boot/initrd.img-5.4.0-33-generic
Найден образ linux: /boot/vmlinuz-5.4.0-31-generic
Найден образ initrd: /boot/initrd.img-5.4.0-31-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
Найден windows 10 на /dev/sda1
завершено
root@ubuntu:/#

```

Рис. 29.5. Найден образ ядра ОС Linux и ОС Windows 10

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 загрузаться.

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

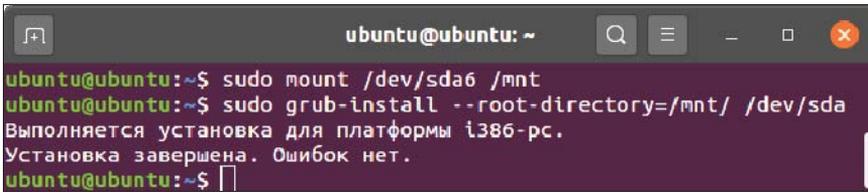
```

Корневая файловая система смонтирована, устанавливаем загрузчик:

```

sudo grub-install --root-directory=/mnt/ /dev/sda

```



```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ sudo mount /dev/sda6 /mnt  
ubuntu@ubuntu:~$ sudo grub-install --root-directory=/mnt/ /dev/sda  
Выполняется установка для платформы i386-pc.  
Установка завершена. Ошибок нет.  
ubuntu@ubuntu:~$ █
```

Рис. 29.6. Успешная установка загрузчика GRUB

Сообщение об отсутствии ошибок говорит о том, что загрузчик успешно установлен (рис. 29.6).

Размонтируем раздел с корневой файловой системой:

```
sudo umount /dev/sda6
```

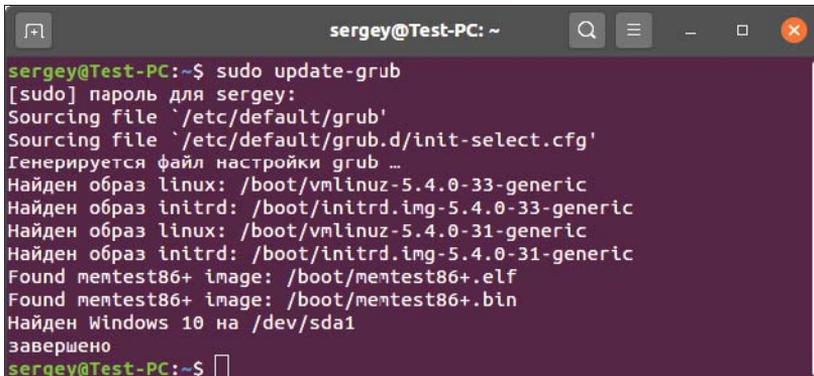
Выполняем перезагрузку уже знакомым нам способом:

```
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 пользоваться как ОС Ubuntu, так и ОС Windows 10.



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ sudo update-grub  
[sudo] пароль для sergey:  
Sourcing file `/etc/default/grub'  
Sourcing file `/etc/default/grub.d/init-select.cfg'  
Генерируется файл настройки grub ...  
Найден образ linux: /boot/vmlinuz-5.4.0-33-generic  
Найден образ initrd: /boot/initrd.img-5.4.0-33-generic  
Найден образ linux: /boot/vmlinuz-5.4.0-31-generic  
Найден образ initrd: /boot/initrd.img-5.4.0-31-generic  
Found mentest86+ image: /boot/mentest86+.elf  
Found mentest86+ image: /boot/mentest86+.bin  
Найден Windows 10 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.

```

GNU GRUB version 2.04

Minimal BASH-like line editing is supported. For the first word,
TAB lists possible command completions. Anywhere else TAB lists
possible device or file completions. ESC at any time exits.

grub> _

```

Рис. 29.8. Режим командной строки в загрузчике GRUB

Итак, когда появится меню загрузчика GRUB, нажмите клавишу <e>. Так вы запустите консоль загрузчика (рис. 29.8).

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.

```

GNU GRUB version 2.04

Minimal BASH-like line editing is supported. For the first word,
TAB lists possible command completions. Anywhere else TAB lists
possible device or file completions. ESC at any time exits.

grub> ls
(hd0) (hd0,msdos6) (hd0,msdos5) (hd0,msdos3) (hd0,msdos2) (hd0,msdos1)
grub> _

```

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 которого 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 посмотрим 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/
swapfile.sys 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/ boot/ swapfile etc/ media/ bin dev/ home/ lib lib32 lib64 libx32
mnt/ opt/ proc/ root/ run/ sbin snap/ srv/ sys/ tmp/ usr/ var/ cdrom/
grub> _
```

Рис. 29.11. Найден раздел с корневой файловой системой ОС Ubuntu

Загрузим ядро ОС Ubuntu Linux следующей командой:

```
linux /boot/vmlinuz root=/dev/sda6
```

После этого определим `initrd` (от англ. *Initial RAM Disk* — диск в оперативной памяти для начальной инициализации):

```
initrd /boot/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 этой главы помогут нам оперативно устранять неисправности, возникающие при работе с операционной системой Ubuntu Linux.



# Установка и использование 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 выполняют в виртуальной машине Oracle VM VirtualBox.

Начинающие пользователи боятся устанавливать незнакомую ОС 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.

В настоящий 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.

## Возможные проблемы при использовании виртуальной машины

Одной самой распространенной проблемой использования ОС Ubuntu в виртуальной машине может стать низкое разрешение экрана. Такое нередко случается, когда операционная система и аппаратное обеспечение компьютера не поддерживают всех необходимых функций для комфортного использования виртуальной машины, и мы можем увидеть следующую картину (рис. 30.1).

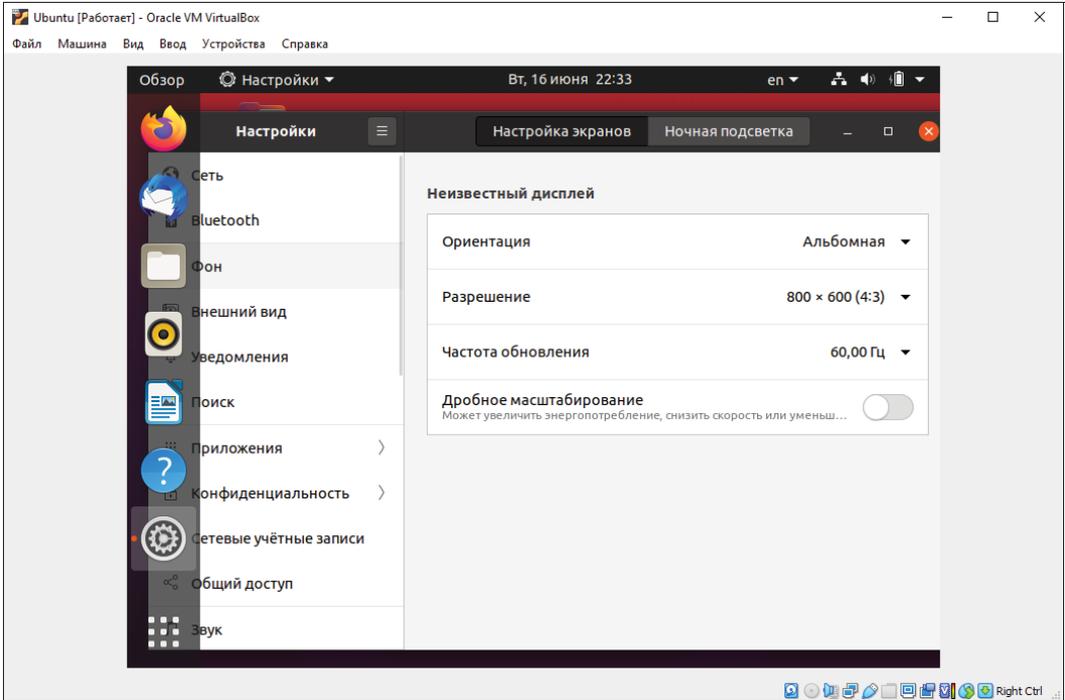


Рис. 30.1. Низкое разрешение экрана

В данном случае максимально доступное разрешение экрана очень низкое и часть элементов не умещается на экране. Даже если развернуть окно виртуальной машины на весь экран, то часть экрана все равно 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, даже из этой. Поэтому не стоит бросать изучение ОС Ubuntu Linux, а имеет смысл решить эту и другие проблемы установкой дополнений.

## 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 выберите пункт меню **Устройства**, а затем пункт **Подключить образ диска Дополнений гостевой ОС...**. Откроется окно автоматического запуска приложения (рис. 30.2).

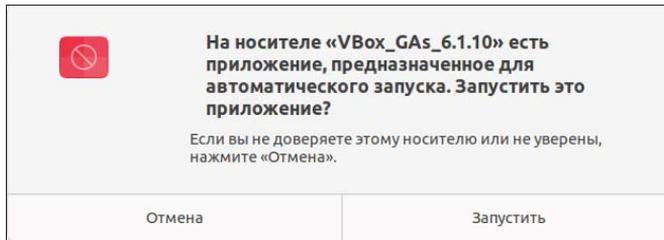


Рис. 30.2. Окно автоматического запуска приложения

Нажмите кнопку **Запустить**, на запрос аутентификации введите свой пароль учетной записи и нажмите кнопку **Подтвердить** (рис. 30.3).

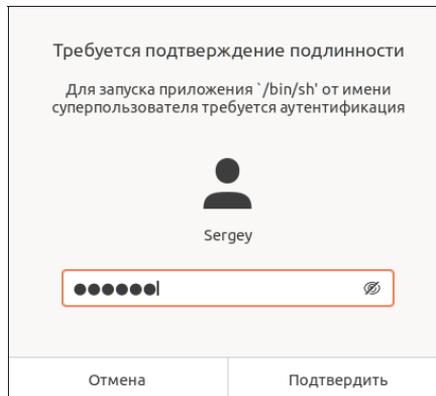


Рис. 30.3. Запрос аутентификации

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...".

```

VirtualBox Guest Additions installation
Verifying archive integrity... All good.
Uncompressing VirtualBox 6.1.10 Guest Additions for Linux.....
VirtualBox Guest Additions installer
Copying additional installer modules ...
Installing additional modules ...
VirtualBox Guest Additions: Starting.
VirtualBox Guest Additions: Building the VirtualBox Guest Additions kernel
modules. This may take a while.
VirtualBox Guest Additions: To build modules for other installed kernels, run
VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup <version>
VirtualBox Guest Additions: or
VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup all
VirtualBox Guest Additions: Building the modules for kernel 5.4.0-37-generic.

This system is currently not set up to build kernel modules.
Please install the gcc make perl packages from your distribution.
VirtualBox Guest Additions: Running kernel modules will not be replaced until
the system is restarted
Press Return to close this window...

```

Рис. 30.4. Установка дополнений гостевой операционной системы

Устанавливаемые дополнения являются обычным файлом VBoxGuestAdditions.iso, который расположен в каталоге C:\Program Files\Oracle\VirtualBox\.

После установки дополнения гостевой операционной системы закройте Терминал. На этом установка дополнений гостевой ОС будет закончена.

## Подключение общей папки

Виртуальная машина Oracle VM VirtualBox устроена таким образом, что вы не сможете использовать общий буфер обмена между вашей основной операционной системой и той, что вы установили в виртуальной машине. Например, вы не сможете скопировать файл из ОС Ubuntu, которая установлена в виртуальной машине, и поместить его в основную ОС. Однако эта проблема решается настройкой общей папки, точнее, общего каталога для обмена файлами между основной и гостевой ОС.

Делается это следующим образом. В главном меню Oracle VM VirtualBox выделите вашу гостевую операционную систему и выберите пункт меню **Настройка**, откроется окно настроек общих папок (рис. 30.5).

В правой части окна нажмите кнопку , чтобы открыть окно добавления общей папки.

В пункте **Путь к папке** укажите путь к каталогу в основной ОС, который вы хотите указать в качестве общей папки. Затем установите флажок **Авто-подключение**. В качестве точки подключения можно указать путь /mnt/share/ (рис. 30.6).

Для сохранения изменений нажмите кнопку **ОК**.

Это еще не все. Если теперь запустить ОС Ubuntu в виртуальной машине и перейти в каталог /mnt/share/, можно увидеть картину, как на рис. 30.7.

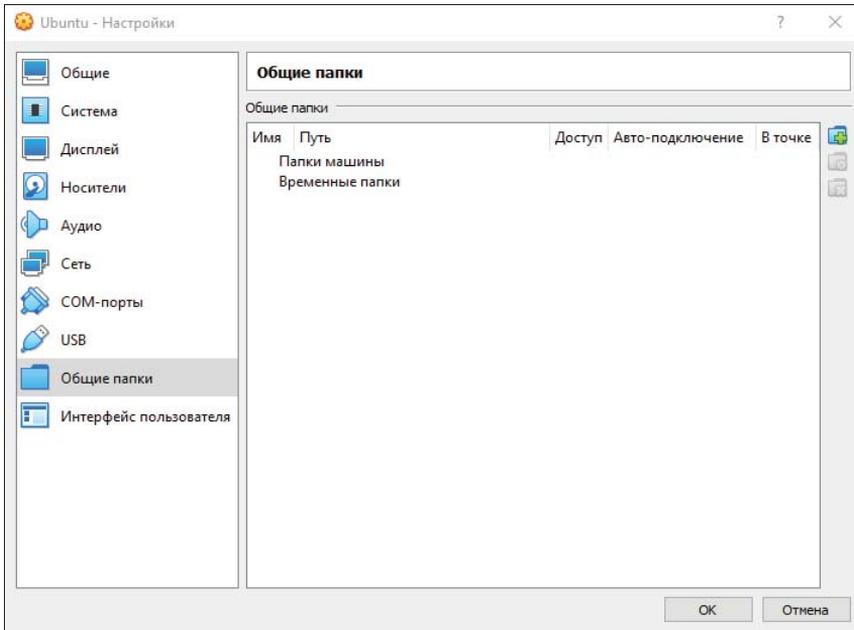


Рис. 30.5. Окно настройки общей папки

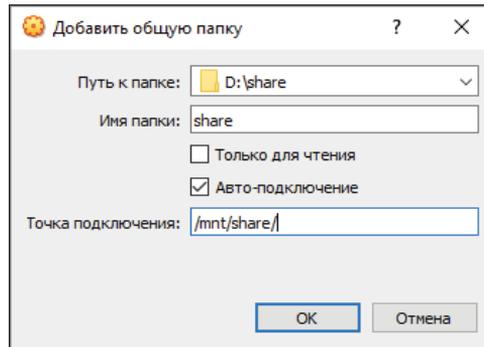


Рис. 30.6. Добавление общей папки

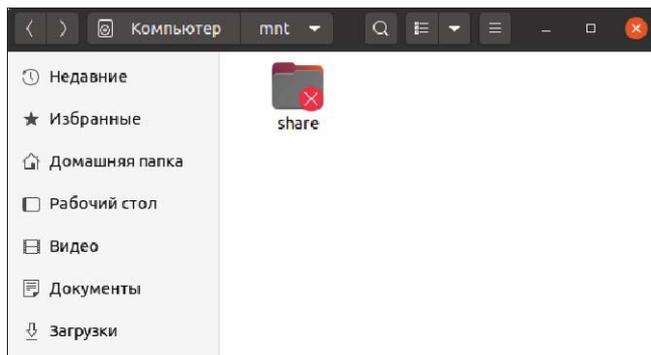


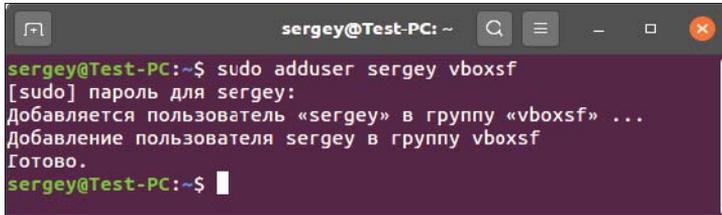
Рис. 30.7. Доступ к общей папке запрещен

Если попытаться открыть данную расшаренную (общую) папку, появится окно ввода вашего пароля учетной записи, и только после аутентификации вы сможете пользоваться созданной общей папкой. Согласитесь, это не совсем удобно. Давайте исправим это.

Откройте Терминал и выполните следующую команду:

```
sudo adduser sergey vboxsf
```

Данная команда добавит пользователя `sergey` в группу `vboxsf`, что позволит пользоваться общим каталогом, не вводя каждый раз пароль учетной записи (рис. 30.8).



```
sergey@Test-PC: ~  
sergey@Test-PC:~$ sudo adduser sergey vboxsf  
[sudo] пароль для sergey:  
Добавляется пользователь «sergey» в группу «vboxsf» ...  
Добавление пользователя sergey в группу vboxsf  
Готово.  
sergey@Test-PC:~$
```

Рис. 30.8. Добавление пользователя в группу

## Выводы

Применение виртуальной машины Oracle VM VirtualBox с установленной гостевой операционной системой позволяет не только изучать ОС, но и "безболезненно" экспериментировать с ОС Ubuntu. При этом пользователю не нужно больше бояться выполнять то или иное действие. Даже если что-то пойдет не так, можно просто переустановить ОС в виртуальной машине и продолжить ее изучение, а еще лучше сделать снимок ОС, и если что-то пойдет не так, всегда можно запустить предыдущую конфигурацию без переустановки всей операционной системы.

Благодаря виртуальной машине теперь ничто не препятствует комфортному изучению ОС Ubuntu Linux, и вам не нужно бояться того, что ваши неправильные действия навредят системе.

## ГЛАВА 31



# Создание и установка сертификатов Let's Encrypt

## Сертификаты Let's Encrypt

Let's Encrypt — это некоммерческий центр сертификации, предоставляющий бесплатные сертификаты для Web-сайтов. Выданные сертификаты позволяют осуществлять TLS-шифрование (HTTPS) сайтов.

Сейчас сложно найти сайт, который не использует сертификат для TLS-шифрования трафика. Особенно это касается банковской сферы, там, где требуется передача пользовательских данных по защищенному каналу во избежание перехвата трафика.

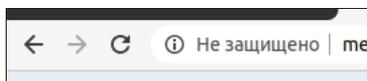
Ранее сертификаты можно было получить только за определенную плату. С появлением Let's Encrypt ситуация в корне поменялась, и теперь у абсолютно всех владельцев сайта появилась возможность установки бесплатного сертификата для его личного сайта.

## Какие бывают сертификаты

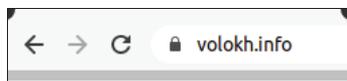
Как вы догадались, сертификаты бывают платными и бесплатными. В чем их различие и зачем нужны платные сертификаты, если уже существуют бесплатные? Давайте разберемся в этом вопросе и рассмотрим основные отличительные особенности этих двух видов сертификатов.

Итак, любой сертификат, будь то платный либо бесплатный, предназначен для подтверждения подлинности доменного имени (рис. 31.1).

При получении бесплатного сертификата Let's Encrypt от владельца сайта требуется только подтвердить его права на сам сайт, так называемый DV SSL (domain



а



б

Рис. 31.1. Сравнение сайта без сертификата (а) и с установленным сертификатом (б)

validation). Однако при этом ни личность владельца, ни его название компании не будут проверены никоим образом и не будут нигде отображены.

В случае с выдачей платного сертификата у владельца сайта есть возможность не только подтвердить право владения сайта, но и подтвердить организацию OV SSL (organization validation). Такой сертификат могут получить любое юридическое лицо или организация, которые желают, чтобы в сертификате отображалось не только право владения сайтом, но и то, что такой субъект действительно существует.

Помимо этого платный сертификат выдается на срок от 1 года и более, в то время как бесплатный сертификат от Let's Encrypt будет выдан только на 90 дней.

Это основные различия между платным и бесплатным сертификатами.

## Способы создания сертификата Let's Encrypt

В настоящее время многие хостинг-провайдеры предоставляют удобный интерфейс для генерации и установки сертификата Let's Encrypt в панели управления хостингом, который позволяет буквально в два клика создать и установить сертификат на ваш сайт (рис. 31.2).

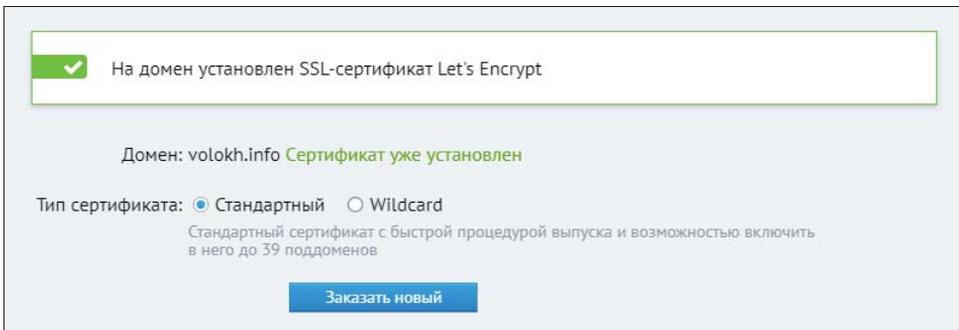


Рис. 31.2. Возможность создания сертификата в панели управления хостингом

Однако не все хостинг-провайдеры предоставляют такую возможность, и устанавливать сертификат в этом случае придется вручную.

Компания Let's Encrypt предусмотрела такую возможность и даже создала специальную программу Certbot, которая позволяет создать запрос на новый сертификат из ОС Linux и затем установить его вручную на хостинге.

Так как ручной способ создания сертификата является самым сложным и в то же время самым интересным, мы будем рассматривать именно его. К тому же данный способ всегда помогает в случаях, когда возможности управления хостингом слишком ограничены.

## Создание сертификата Let's Encrypt

Для того чтобы создавать сертификаты в ОС Ubuntu, необходимо установить программу Certbot. Это выполняется следующей командой:

```
sudo apt install letsencrypt
```

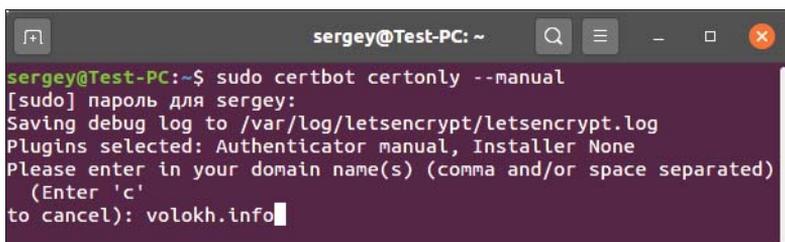
Данная команда установит все зависимости для работы с программой Certbot, и на то не потребуется много времени.

Теперь начинается самое интересное: генерация запроса на создание сертификата.

Итак, у нас имеется домен **volokh.info**, для которого нам необходимо выпустить сертификат. Для этого в Терминале выполняем следующую команду:

```
sudo certbot certonly --manual
```

В ответ на это Certbot запросит указать имя домена, для которого необходимо создать сертификат (рис. 31.3).

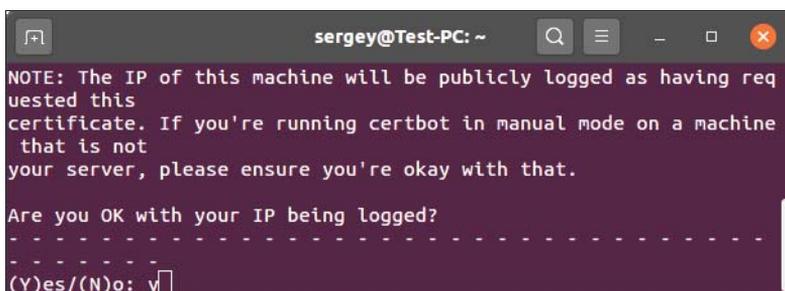


```
sergey@Test-PC: ~  
sergey@Test-PC:~$ sudo certbot certonly --manual  
[sudo] пароль для sergey:  
Saving debug log to /var/log/letsencrypt/letsencrypt.log  
Plugins selected: Authenticator manual, Installer None  
Please enter in your domain name(s) (comma and/or space separated)  
(Enter 'c'  
to cancel): volokh.info
```

Рис. 31.3. Задание домена для генерации сертификата

Введите домен и нажмите <Enter>.

На следующем шаге создания сертификата необходимо согласиться с сохранением вашего IP-адреса в журнале событий генерации сертификата (рис. 31.4).



```
sergey@Test-PC: ~  
NOTE: The IP of this machine will be publicly logged as having req  
uested this  
certificate. If you're running certbot in manual mode on a machine  
that is not  
your server, please ensure you're okay with that.  
  
Are you OK with your IP being logged?  
-----  
(Y)es/(N)o: y
```

Рис. 31.4. Запрос на сохранение IP-адреса в лог-файлах

Нажмите клавишу <Y>, а затем <Enter>.

Теперь необходимо в корневом каталоге вашего сайта создать каталог `.well-known` с подкаталогом `acme-challenge` (рис. 31.5).

```

sergey@Test-PC: ~
-----
Create a file containing just this data:

O-_gb-4W4cRShjzgmZWXYbahDoIfY2io73PFDFh2jI.5SCLxJnXaLT7oJBrQpKGqC
YUwhBBQ4DZh0WluQ10W4Q

And make it available on your web server at this URL:

http://volokh.info/.well-known/acme-challenge/O-_gb-4W4cRShjzgmZW
XYbahDoIfY2io73PFDFh2jI
-----
Press Enter to Continue

```

Рис. 31.5. Запрос на создание файла для верификации домена

В каталоге `acme-challenge` необходимо создать текстовый файл `O-_gb-4W4cRShjzgmZWXYbahDoIfY2io73PFDFh2jI` с содержимым из листинга 31.1.

**Листинг 31.1. Содержимое файла `O-_gb-4W4cRShjzgmZWXYbahDoIfY2io73PFDFh2jI`**

```
O-_gb-4W4cRShjzgmZWXYbahDoIfY2io73PFDFh2jI.5SCLxJnXaLT7oJBrQpKGqCYUwhBBQ4DZh0
WluQ10W4Q
```

После того как вы создали необходимые указанные каталоге, а также файл с требуемым содержимым, можно нажать клавишу `<Enter>`.

**ПРИМЕЧАНИЕ**

Обратите внимание, что если у вас в корневом каталоге вашего сайта имеется файл `.htaccess`, в котором прописаны правила, меняющие имена ссылок на страницы или запрещающие доступ к файлам, то данный файл лучше всего временно переименовать для того, чтобы программа Certbot смогла проверить ваше право владения запрашиваемым сайтом.

```

sergey@Test-PC: ~
Press Enter to Continue
waiting for verification...
Cleaning up challenges

IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:
   /etc/letsencrypt/live/volokh.info/fullchain.pem
   Your key file has been saved at:
   /etc/letsencrypt/live/volokh.info/privkey.pem
   Your cert will expire on 2020-09-15. To obtain a new or tweaked
   version of this certificate in the future, simply run certbot
   again. To non-interactively renew *all* of your certificates, run
   "certbot renew"
 - If you like Certbot, please consider supporting our work by:

   Donating to ISRG / Let's Encrypt:  https://letsencrypt.org/donate
   Donating to EFF:                  https://eff.org/donate-le

sergey@Test-PC:~$

```

Рис. 31.6. Запрос на создание сертификата выполнен успешно

В случае успеха (рис. 31.6) в каталоге `/etc/letsencrypt/live/volokh.info/` будут созданы четыре файла:

- ◆ `privkey.pem` — приватный ключ вашего сертификата;
- ◆ `fullchain.pem` — файл сертификата, который включает в себя сертификат сервера и цепочки;
- ◆ `chain.pem` — сертификат цепочки;
- ◆ `cert.pem` — сертификат сервера.

После получения файлов сертификата нам необходимо их установить. Сейчас этим и займемся.

## Установка сертификата в панели управления хостингом

Нам понадобятся только два файла: `privkey.pem` и `fullchain.pem`. Откройте их в текстовом редакторе Gedit:

```
sudo gedit /etc/letsencrypt/live/volokh.info/privkey.pem
/etc/letsencrypt/live/volokh.info/fullchain.pem
```

Данные файлы являются обычными текстовыми файлами, поэтому они открываются в обычном текстовом редакторе Gedit.

### **ПРИМЕЧАНИЕ**

Сам процесс установки сертификата у разных хостинг-провайдеров очень схож по своей сути. Отличие будет заключаться только в интерфейсе панели управления хостингом. Поэтому описываемые ниже действия будут незначительно отличаться в зависимости от того или иного хостинг-провайдера.

После того как вы открыли содержимое файлов `privkey.pem` и `fullchain.pem`, перейдите в панель управления вашим хостингом и найдите раздел, предназначенный для установки собственного SSL-сертификата (рис. 31.7).

В поле ввода **Сертификат** скопируйте содержимое файла `fullchain.pem`, а в поле **Приватный ключ** — содержимое файла `privkey.pem`. Нажмите кнопку **Установить**, и сертификат будет успешно установлен.

### **ПРИМЕЧАНИЕ**

Сертификат будет активирован на хостинге в течение нескольких минут. Поэтому в первые минуты после установки сертификата на вашем сайте все еще будет присутствовать старый сертификат.

Таким образом, выполняется установка сертификата на сайт в ручном режиме. Это самый сложный метод, но он универсален в большинстве случаев и выручает, когда хостинг-провайдер предоставляет ограниченную функциональность управления хостингом. Разумеется, процесс установки сертификата можно провести прямо на сервере, минуя ручное копирование и установку, но это в том случае, если ваш хостинг-провайдер предоставляет вам такую возможность, что бывает очень редко. Как правило, такая возможность присутствует только на VPS- или VDS-хостинге.

Бесплатный SSL сертификат    Заказ SSL сертификата    Установка SSL сертификата

Если ранее вы уже приобрели SSL сертификат, то в этом окне вы можете установить его.

Сертификат: X4Po1QYz+3dszkDqMp4fklxBwXRsw10KXzPMTZ+sOPAveyxindmjkW8lGy+QsRlGPFz+G6Z6h7mjem0Y+iWlkYcV4PIWL1iwBi8saCbGS5jN2p8M+X+Q7UNKEkROb3N6KOqkqm57TH2H3eDIAkSnh6/DNFu0Qg==-----END CERTIFICATE-----

Приватный ключ: -----BEGIN PRIVATE KEY-----MIIIEvwlBADANBgkqhkiG9w0BAQEFAASCBCkkggSlAgEAAoIBA QDmNEn8vKmA4EeYJQQrsXPAYMLXS7Stq8BfcR+99yaaVSDJRPYKuioQKp0dnwkp2pMSX1moe+xarLbudVyjfd41fhrdioaWwauPWrh5rDh2XgOj7kCrc1fmJL36OvEd4XGnUcGT62nWkku

Домен: volokh.info

Выделенный IP: Не использовать

Установить

Рис. 31.7. Меню установки собственного сертификата

## Проверка правильности установки сертификата

После того как мы установили сертификат, будет не лишним проверить правильность его установки.

### Результат проверки SSL

Домен:  ПРОВЕРИТЬ

✔ Сертификат проверен и установлен правильно

DST Root CA X3  
Let's Encrypt Authority X3  
volokh.info

**Информация о сервере**

IP адрес(а)

- 87.236.16.27

Сервер

- nginx-reuseport/1.13.4

Домен		volokh.info
Сертификат	Альтернативные имена (SANs)	volokh.info
	Тип сертификата	Подтверждение организации (OV)
Подлинность	Действителен с	✔ 17 июня 2020
	Истекает	✔ 15 сентября 2020 (Истекает через 89 дней)
	OSCP статус	✔
Подробности	CLR статус	Не применимо
	Алгоритм	2048 bit RSA key
	Подпись	sha256WithRSAEncryption
Подписи	MDS	24282e1dc05d14b543c156f95c1843bf
	SHA1	50b38e9bc1f18e39f80729a7ad428914771b8e9
	Серийный номер	414426033471498857652444922100510681433625

Рис. 31.8. Сертификат установлен корректно

Самый простой способ — зайти на ваш сайт, в адресной строке щелкнуть по значку защищенного соединения и уже там проверить дату выпуска сертификата и пути сертификации. Но не лучше ли воспользоваться специальными сервисами? Давайте так и поступим: перейдите по адресу [https://www.leaderssl.ru/tools/ssl\\_checker](https://www.leaderssl.ru/tools/ssl_checker), в поле ввода введите адрес вашего сайта и нажмите кнопку **ПРОВЕРИТЬ** (рис. 31.8).

В случае возникновения каких-либо ошибок сервис сообщит об этом.

## Продление сертификата

Продление сертификата подразумевает замену старого сертификата новым, т. е. перевыпуск сертификата. Для продления (перевыпуска) сертификата необходимо проделать все те же самые действия, что мы выполняли при генерации сертификата, за исключением установки Certbot, т. к. эта программа уже установлена. Таким образом, для перевыпуска сертификата достаточно выполнить команду

```
sudo certbot certonly --manual
```

и следовать инструкциям Certbot, а потом установить сертификата в панели управления хостингом.

### **ПРИМЕЧАНИЕ**

Существует возможность настройки автоматического продления и установки сертификата. Ее мы рассматривать не будем по причине того, что такая функциональность доступна не у всех хостинг-провайдеров, а только у тех, кто предоставляет услуги VPS- или VDS-хостинга.

## Выводы

Мы познакомились с сертификатами шифрования трафика и узнали, какими они бывают. Мы научились устанавливать утилиту Certbot, которая предназначена для генерации сертификатов, а также выяснили, как подключать созданный сертификат в панели управления хостингом.

Данные знания будут полезны тем, кто занимается обслуживанием интернет-сайтов и отвечает за их функциональность.



## ГЛАВА 32

# Настройка файлового сервера Samba

## Файловый сервер Samba

Samba — это пакет программного обеспечения в ОС Linux, который предоставляет общий сетевой доступ к сетевым дискам и принтерам по протоколу SMB/CIFS в пределах локальной сети предприятия или домашней сети.

Доступ файловому серверу может осуществляться с любой операционной системы, будь то Linux или Windows.

Файловый сервер Samba хорош тем, что данное программное решение является бесплатным как для личного, так и коммерческого использования. При всем этом данное программное решение разворачивается на ОС Linux, что также означает отсутствие затрат на покупку лицензии на операционную систему.

## Подготовка перед установкой Samba

Перед установкой файлового сервера Samba необходимо определиться с каталогом, который будет выступать в качестве файлового обменника. Отличным решением будет создание каталога `samba` в домашнем каталоге вашей учетной записи, который вынесен на отдельный раздел жесткого диска или на отдельный жесткий диск. Например, каталог `/home/sergey/samba/` будет разбит на подкаталоги в соответствии со структурными подразделениями:

- ❑ 105 — отдел кадров;
- ❑ 109 — бухгалтерия;
- ❑ 112 — технический отдел;
- ❑ 215 — IT-отдел.

Таким образом, каталог `/home/sergey/samba/109/` будет предназначен для обмена файлами между сотрудниками бухгалтерии, а каталог `/home/sergey/samba/112/` — для файлов сотрудников технического отдела и т. д.

Такая организация каталогов позволит упорядочить процесс обмена файлами. К примеру, сотрудник отдела кадров хочет передать файл, предназначенный для

технического отдела, поэтому он поместит его в каталог `/home/sergey/samba/112/`. Конечно, можно создать общий каталог и организовать обмен только в одном каталоге, но это не совсем удобное решение. Представьте ситуацию, когда в организации работают не 10, а 100 сотрудников и более, и если хотя бы часть из них будет одновременно пользоваться одной общей папкой, то в этом каталоге будет столько неразберихи, что придется пересматривать все файлы в поисках того, который предназначался именно вам. А если в каталоге уже будет файл с одноименным названием, то данный файл будет просто перезаписан тем файлом, который был скопирован в этот каталог последним, т. к. большинство пользователей не вчитываются в сообщения диалоговых окон и просто соглашаются, нажав кнопку **ОК** или **Далее**. Теперь понятно, для чего нужно создавать подкаталоги в папке `/home/sergey/samba/`?

Хорошо. Теперь создадим для примера два каталога:

```
mkdir -p /home/sergey/samba/105/  
mkdir -p /home/sergey/samba/215/
```

Данные команды создадут каталоги 105 и 215 в каталоге `/home/sergey/samba/`.

#### **ПРИМЕЧАНИЕ**

Числа 105 и 215 означают номера кабинетов отдела. Данное наименование каталогов выбрано для удобства дальнейшего сопровождения и поддержки файлового сервера Samba.

Сейчас у вас возникнет резонный вопрос: почему для удобства не назвать каталог не только номером, но и названием отдела? Можно, но бессмысленно, и вскоре вы в этом убедитесь.

После создания каталогов нам необходимо настроить права доступа для того, чтобы остальные пользователи файлового сервера Samba могли не только просматривать список файлов, но и создавать и удалять файлы:

```
chmod -R 0777 /home/sergey/samba/105/  
chmod -R 0777 /home/sergey/samba/215/
```

Подготовительные действия окончены, перейдем к установке Samba.

## **Установка файлового сервера Samba**

Для того чтобы установить Samba, выполните следующую команду:

```
sudo apt install -y samba samba-client
```

Данная команда позволит нам установить ПО, необходимое для клиентского подключения к файловому серверу Samba.

Файловый сервер Samba мы установили, но для того, чтобы он заработал, необходимо настроить конфигурационный файл `/etc/samba/smb.conf`. Однако перед любым редактированием конфигурационных файлов рекомендуется сделать их резервные копии на случай, если возникнет нештатная ситуация, и тогда можно будет безболезненно "откатиться" на исходную рабочую конфигурацию:

```
sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.bak
```

Ну что ж, теперь можно приступить к редактированию самого конфигурационного файла `smb.conf`. Откройте данный файл в текстовом редакторе Nano или Gedit с правами администратора и замените его содержимое строками из листинга 32.1.

**Листинг 32.1. Содержимое файла `/etc/samba/smb.conf`**

```
#=== Global Settings ===
[global]
security = user
workgroup = WORKGROUP
server string = Samba Server %v
guest account = nobody
map to guest = bad user
null passwords = yes
encrypt passwords = true
#=== Share Definitions ===
[105 - Отдел кадров]
comment = Начальник отдела Иванова М.П.
path = /home/sergey/samba/105/
browseable = yes
guest ok = yes
writable = yes
public = yes
[215 - IT - отдел]
comment = Начальник отдела Петров С.В.
path = /home/sergey/samba/215/
browseable = yes
guest ok = yes
writable = yes
public = yes
```

Сохраните проделанные изменения и перезагрузите службу `smbd`:

```
sudo service smbd restart
```

Если теперь в Проводнике Windows обратиться по адресу `\\192.168.1.77\`, откроется только что созданный файловый сервер Samba (рис. 32.1).

**ПРИМЕЧАНИЕ**

Обратите внимание, что IP-адрес `192.168.1.77` может отличаться в зависимости от того, какой IP-адрес назначен вашей машине. Соответственно, при обращении к вашему файловому обменнику Samba необходимо подставлять именно ваш IP-адрес.

Таким простым способом создается простейший файловый сервер Samba, который доступен не только в ОС Linux, но и в Windows.

Мы рассмотрели далеко не все возможности файлового сервера Samba, т. к. при желании и необходимости можно организовать доступ к файловому серверу по логину и паролю, а также по IP-адресу в пределах локальной сети. Все это позволит

вам более тонко и гибко настроить безопасность файлового сервера. Однако в большинстве случаев организации простого доступа к файловому серверу Samba достаточно для большинства предприятий, которым необходим простой обмен файлами между сотрудниками организации.

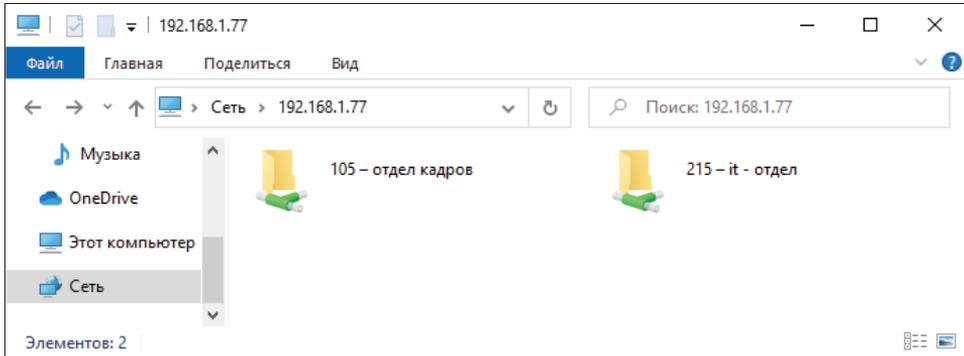


Рис. 32.1. Корень файлового сервера Samba

## Пояснение директив конфигурационного файла smb.conf

Для того чтобы понимать, что именно означает та или иная директива созданного нами конфигурационного файла, разберем более детально используемые директивы нашего файла.

Итак, файл начинается с символа решетки # (диез), который обозначает начало строки с комментарием. Далее идет простой текстовый комментарий `=== Global Settings ==`, означающий, что в данном разделе будут заданы глобальные настройки, т. е. общие настройки, которые влияют на конфигурацию Samba в целом.

На второй строке директивой идет секция `[global]`, в которой и описываются глобальные настройки файлового сервера Samba.

### **ПРИМЕЧАНИЕ**

Существуют также секции `[public]` и `[private]`, которые задают настройки публичного и приватного доступа.

Третьей строкой в параметре `security` мы задаем режим безопасности нашего сервера и передаем ему значение `user`, которое говорит о том, что аутентификация приватных каталогов будет основываться на логине и пароле.

Параметр `workgroup` задает рабочую группу для машин на ОС Windows, и его значение равно одноименному параметру.

Параметр `server string` задает строку комментария к файловому серверу. В нашем случае мы присваиваем значение `Samba Server %v` с выводом версии сервера. Версию сервера определяет значение `%v`.

Параметр `guest account` задает имя пользователя, которое будет использовано для доступа к файловому серверу Samba, в настройках сервиса которого указан параметр `guest ok = yes`. В нашем случае мы указываем значение `nobody`, означающее, что доступ будет осуществлен с правами гостевой учетной записи.

Параметр `map to guest` задает способ обработки запросов аутентификации при подключении к серверу Samba. У нас указано значение `bad user`, т. е. при вводе неправильного пароля учетной записи такой запрос будет отклонен.

Параметр `null passwords` определяет возможность подключения учетной записи с пустым паролем. Значение `yes` разрешает таким учетным записям подключение к файловому серверу.

Параметр `encrypt passwords` позволяет задать шифрование паролей учетных записей, созданных для подключения к файловому серверу Samba. Значение `yes` устанавливает шифрование созданных паролей.

Далее идет описание конфигурации общих директорий, о чем говорит комментарий `=== Share Definitions ===`.

Параметр `[105 - Отдел кадров]` задает имя сетевого каталога, которое будет отображаться в Проводнике Windows или в файловом менеджере в ОС Ubuntu (рис. 32.2).

### СОВЕТ

Подключиться к файловому серверу Samba из ОС Ubuntu можно напрямую из файлового менеджера Nautilus. Для этого в левой части окна нажмите на **Другие места**, внизу окна в адресной строке наберите `smb://192.168.1.77/` и нажмите кнопку **Подключиться**.

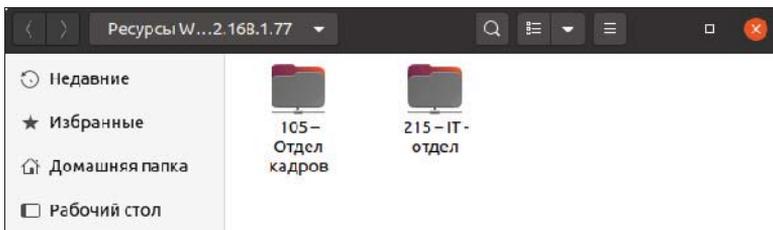


Рис. 32.2. Файловой сервер в ОС Ubuntu

### ПРИМЕЧАНИЕ

Обратите внимание, что имя папки в ОС Windows указано в нижнем регистре (см. рис. 32.1.), а ОС Ubuntu указано так, как мы задали это в конфигурационном файле `/etc/samba/smb.conf`. Такое поведение связано с особенностями файловой системы ОС Windows и ОС Linux. Для того чтобы задать чувствительность к регистру, добавьте параметр `case sensitive` со значением `yes` в конфигурационный файл `/etc/samba/smb.conf`.

Вот сейчас настал момент, когда необходимо прояснить возникший у вас в начале главы вопрос, почему мы изначально задавали имена каталогам без наименования отделов. Дело в том, что имя каталога можно указать в конфигурационном файле

нашего файлового сервера Samba. Для этого мы и написали в квадратных скобках имя каталога — 105 - Отдел кадров.

Данное решение позволяет не задавать длинные имена каталогам, что, в свою очередь, упрощает администрирование файлового сервера Samba. Отличное решение, не так ли? Поехали дальше.

Параметр `comment` позволяет задавать дополнительный комментарий к сетевому каталогу. Этот комментарий мы можем увидеть в Проводнике Windows, если наведем курсор на папку и подождем несколько секунд. Данный параметр не играет никакой роли в конфигурации сервера, но иногда бывает полезным, когда дополнительно требуется сопроводить пояснениями сетевой каталог.

В параметре `path` мы указываем полный путь к нашему каталогу — `/home/sergey/samba/105/`.

Параметр `browseable` позволяет установить видимость в сетевом окружении. Значение `yes` указывает на то, что каталог будет виден в сетевом окружении.

Параметр `guest ok` задает возможность авторизации без пароля, о чем говорит значение `yes`.

Параметр `writable` разрешает (`yes`) или, наоборот, запрещает (`no`) запись в данный каталог.

И последний параметр `public` определяет, будет ли у авторизованных пользователей доступ на чтение данного каталога. Значение `yes` позволяет пользователям работать с данным каталогом.

Для каталога `/home/sergey/samba/215/` настройки задаются аналогичным образом.

Как видите, нет ничего сложного в настройке файлового сервера Samba.

## Выводы

Мы познакомились с файловым сервером Samba. Мы научились устанавливать файловый сервер Samba, а также настраивать права доступа к созданным каталогам. Кроме того, на примере мы узнали о том, как правильно организовать структуру и задавать имена каталогов, предназначенных для общего доступа в пределах локальной сети.

Полученные знания будут полезны тем, кто занимается администрированием сетевой инфраструктуры организации. Хотя файловый сервер Samba и не требует постоянного обслуживания, но знать, как именно создавать дополнительные каталоги и производить их настройку, необходимо не только системному администратору, но и опытному пользователю, который мечтает создать свой домашний файловый сервер.

# 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 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/9785977567619.zip> можно скачать электронный архив с материалами, не вошедшими в печатную версию книги. Эта ссылка доступна также со страницы книги на сайте издательства [www.bhv.ru](http://www.bhv.ru).

Содержимое электронного каталога представлено в табл. П1.

*Таблица П1. Описание электронного каталога*

Папки	Описание
\ Приложения	Дополнительные материалы по работе с прежними версиями ОС Ubuntu: <ul style="list-style-type: none"><li>• "Настройка подключения к Интернету в программе NetworkManager";</li><li>• "Настройка спящего режима";</li><li>• "Увеличение разрешения контейнера VirtualBox в ОС Windows 7";</li><li>• "Установка драйверов"</li></ul>
\volokh.info	Избранные статьи и заметки из блога автора <b>volokh.info</b>

# Предметный указатель

## A

ACPI 354  
Advanced Linux Sound  
Architecture (ALSA) 225  
◇ AlsaMixer 229  
APN 88  
Apt 256  
Audacious, аудиоплеер 273

## B

bash 150  
BIOS 48, 340  
Brasero, запись дисков 276  
Btrfs 39

## C

Chromium, браузер 274  
Common UNIX Printing System  
(CUPS) 233  
◇ настройка общего доступа  
238  
Coordinated Universal Time 112  
Cpyleft 23

## D

Debian 20  
Double Commander 276  
Dpkg 253  
DSL-соединение:  
◇ настройка 87  
◇ проблема с подключением 90

## E

EFI 49  
ext 38  
ext3 38  
ext4 39  
Ext4 67  
Extended Boot Record (EBR) 42  
Extended File System 38

## F

Fedora 20  
Filesystem Hierarchy Standard  
(FHS) 125  
Fourth extended file system 39  
Freax 16

## G

GDM 368  
Gedit, редактор 115  
General Public License 18, 23  
GIMP, графический редактор  
276  
GNU 17, 18, 19, 23  
GNU/Linux 18  
GParted, метка устройства 332  
GParted Partition Editor 53, 54  
GPT 42  
GRand Unified Bootloader  
(GRUB) 72, 115, 340  
◇ восстановление загрузчика 388  
GUID Partition Table 42

## I

ICMP Echo-Reply 374  
ICMP Echo-Request 374  
init, программа 340  
Initial RAM Disk 391

## J

JFS 40

## L

Let's Encrypt 399  
◇ проверка правильности  
установки 404  
◇ продление сертификата 405  
◇ создание сертификата 401  
◇ установка сертификата 403  
Linux 16  
Linux Mint 21  
Live-режим 51, 62, 73  
◇ восстановление загрузчика  
389  
▫ GRUB 385  
Long Term Support (LTS) 29

## M

Master boot record (MBR) 42  
Minix 16  
Minix File System 38

## N

Nautilus 120, 205  
NetworkManager 74, 90

## O

Open Sound System (OSS) 225  
Oracle VM VirtualBox 393  
◇ установка дополнений 395

## P

Personal Packages Archive (PPA)  
266  
Pinta, графический редактор 276  
Power-On Self-Test (POST) 71  
PPPoE-соединение 81  
pppoeconf 81

## Q

qBittorrent 275

## R

Red Hat Enterprise Linux 19  
Reiser4 39  
ReiserFS 39  
Rufus 35

## S

Samba 406  
◇ конфигурационный файл  
409  
◇ организация каталогов 406  
◇ установка 407  
SGID 308  
Slackware 19  
Slackware Linux 21  
Snap-пакет 268  
Sticky 308  
Sudo 148, 200  
sudo, утилита 148  
SUID 308  
Swapfile 45  
Systemd 343

## U

Ubuntu 20  
Ubuntu Desktop 29  
Ubuntu Linux 28  
Ubuntu Server 29  
UNetbootin 33, 50  
Unified Extensible Firmware  
Interface (UEFI) 49  
UNIX 17

USB-модем 87  
 ◇ подключение 87  
 User Identifier (UID) 199  
 UTC 112  
 UUID 329

**V**

Visudo, утилита 208  
 VLC, видеоплеер 274

**X**

XFS 39

**W**

Wi-Fi 74  
 Wine 278

**Z**

ZFS 40

**A**

Архив:  
 ◇ несжатый 288  
 ◇ пакетов персональный 114  
 ◇ сжатый 288, 291  
 Архиватор:  
 ◇ bzip2 295  
 ◇ gzip 293  
 ◇ tar 289  
 Ассоциация файлов 141  
 Аудиодрайвер 225

**B**

Быстрый запуск приложений  
 107

**B**

Видеодрайвер AMD Catalyst 222  
 Виртуальная машина 393

**Г**

Группы пользователей 300

**Д**

Дерево каталогов 124  
 Дескриптор индексный inode 319  
 Дефрагментация диска 46  
 Дистрибутив 18  
 Драйвер дополнительный 212

**Ж**

Журнал 355

**З**

Зависимость пакетов 247  
 Загрузчик GRUB 72  
 Запись:  
 ◇ главная загрузочная 42  
 ◇ учетная:  
 □ root 145  
 □ администратора 144  
 □ обычная 144  
 □ тип 147

**И**

Именованние дисков 122  
 Имя пользователя, получение  
 163

**К**

Карта сетевая:  
 ◇ Broadcom 217  
 ◇ Realtek 214  
 Каталог:  
 ◇ /bin/ 128  
 ◇ /boot/ 129  
 ◇ /cdrom/ 129  
 ◇ /dev/ 129  
 ◇ /etc/ 129  
 ◇ /home/ 130  
 ◇ /lib/ 130  
 ◇ /lib64/ 130  
 ◇ /lost+found/ 131  
 ◇ /media/ 129, 131  
 ◇ /mnt/ 131  
 ◇ /opt/ 131  
 ◇ /proc/ 131  
 ◇ /root/ 132  
 ◇ /run/ 132  
 ◇ /sbin/ 132  
 ◇ /snap/ 132  
 ◇ /srv/ 133  
 ◇ /sys/ 133  
 ◇ /tmp/ 133  
 ◇ /usr/ 133  
 ◇ /var/ 133  
 ◇ копирование 172  
 ◇ переименование 176  
 ◇ перемещение 176  
 ◇ поиск 186  
 ◇ пользователя, домашний  
 128

◇ создание 168  
 ◇ удаление 169, 170  
 Клавиша <Super> 109  
 Команда:  
 ◇ cat 178  
 ◇ cd 154  
 ◇ chattr 315

◇ chmod 312  
 ◇ chown 315  
 ◇ clear 161  
 ◇ cp 172  
 ◇ echo 153  
 ◇ exit 161  
 ◇ file 297  
 ◇ find 186  
 ◇ grep 188  
 ◇ history 160  
 ◇ journalctl 356  
 ◇ kill 364, 366  
 ◇ killall 366  
 ◇ less 182  
 ◇ ln 321  
 ◇ ls 156  
 ◇ lsattr 317  
 ◇ man 158  
 ◇ mkdir 168  
 ◇ more 182  
 ◇ mv 176  
 ◇ pwd 156  
 ◇ rm 170  
 ◇ rmdir 169  
 ◇ shutdown 196  
 ◇ sudo 200  
 ◇ tac 181  
 ◇ touch 166  
 ◇ umask 311  
 ◇ uname 154  
 ◇ which 188  
 ◇ who 164  
 ◇ whoami 163  
 Консоль 149  
 Корень файловой системы 124  
 Корректировка времени 112

**Л**

Линус Торвальдс 16  
 Лог 355  
 Локализация 95

**М**

- Маска пользовательская 311
- Менеджер:
  - ◇ архивов 285, 286
  - ◇ обновлений 92
    - обновление приложений 92
    - сведения об обновлениях 92
  - ◇ приложений Ubuntu 248
    - удаление программ 269
    - установка:
      - BitTorrent-клиента 275
      - Google Chrome 252
      - qBittorrent 249
      - браузера 274
      - графического редактора 276
      - мультимедийных проигрывателей 273
      - программы для работы с CD/DVD-дисками 277
      - файлового менеджера 276
- Меню:
  - ◇ главное 105
  - ◇ приложений 105
- Метка устройства 329
- Монтирование:
  - ◇ автоматическое 335
  - ◇ ручное 338
  - ◇ с помощью метки устройства 338

**Н**

Настройки системы 108

**О**

Операционная система, остановка 196

**П**

- Панель задач 101
- Панель запуска 253
- Панель основного меню 99
- Параметры 108
  - ◇ системы, смена раскладки клавиатуры 110
- Патрик Фолькердинг 19
- Подключение типа Ethernet 77
- Пользователь:
  - ◇ root 199
  - ◇ сведения 164
- Программа:
  - ◇ kill 364
  - ◇ ping 374
  - ◇ ps 362
  - ◇ traceroute 375
- Прыжок 376

**Р**

- Рабочее место 103
- Рабочий стол 98, 103
- Раздел:
  - ◇ жесткого диска 41
  - ◇ первичный 42
  - ◇ расширенный 42
- Расширение файла 136
- Режим спящий 113
- Репозиторий 114, 264
  - ◇ добавление 266
  - ◇ управление 265
- Ричард Столлман 17

**С**

- Свободное программное обеспечение (СПО) 23
- Своп-файл *См.* Файл подкачки
- Сектор загрузочный 42
- Сервер файловый 406
  - ◇ конфигурационный файл 409
  - ◇ организация каталогов 406
  - ◇ установка 407
- Сертификат 401
  - ◇ Let's Encrypt 399
  - ◇ проверка правильности установки 404
  - ◇ продление 405
  - ◇ установка 403
- Система файловая 38
  - ◇ корневая 127
  - ◇ инициализации systemd 348
- Системный монитор 369
  - ◇ top 358
- Служба 347
- Ссылка 320
  - ◇ жесткая 321
  - ◇ символическая 320
  - ◇ создание 321
- Суперпользователь root 143

**Т**

- Таблица разделов 42
- Текстовый редактор 208
- Терминал 76, 149
  - ◇ восстановление загрузчика 388
  - ◇ добавление репозитория 267
  - ◇ запуск:
    - Windows-приложений 283
    - X-диагностики 396
  - ◇ команды 153
  - ◇ подсказки 159
  - ◇ просмотр прав доступа к файлу 302
  - ◇ удаление Bluefish 270

- ◇ установка:
  - Atom 267
  - Avidemux 260
  - Wine 278
- Точка монтирования 43, 67, 327
- ◇ основная 43

**У**

- Удаленный рабочий стол 378
  - ◇ AnyDesk 381
  - ◇ Remmina 378
- Уровень выполнения 340

**Ф**

- Файл:
  - ◇ /etc/systemd/journald.conf 355
  - ◇ logind.conf 354
  - ◇ sudoers 200, 207
  - ◇ ассоциация 141
  - ◇ вывод содержимого:
    - на экран 178
    - в обратной последовательности 181
  - ◇ исполняемый 139
  - ◇ копирование 172
  - ◇ переименование 176
  - ◇ перемещение 176
  - ◇ подкачки 45
    - создание 57
  - ◇ поиск 186
    - критерии 188
    - шаблона 188
  - ◇ просмотр содержимого 182
  - ◇ создание 166, 167
  - ◇ слияние с другим файлом 180
  - ◇ текстовый 138
  - ◇ удаление 170
  - ◇ установка времени изменения 166

**Ц**

Цель 343

**Э**

Экранирование 155  
Экстент 318

**Ю**

Юнит 344, 348

**Я**

Язык системы 95
 

- ◇ установка и удаление языков 96