

# Файловая система Linux

## Глава

# 21

## 21.1. РАЗЛИЧНЫЕ ФАЙЛОВЫЕ СИСТЕМЫ

Любая операционная система поддерживает несколько файловых систем, например, Windows поддерживает файловые системы FAT, FAT32, NTFS, ISO9660, UDF и др. Linux — тоже не исключение. Однако нужно различать файловые системы, которые могут использоваться в качестве корневой файловой системы, и файловые системы, которые просто поддерживает ОС, но которые не используются для установки Linux.

В качестве корневой файловой системы применяются следующие файловые системы.

- ext — первая файловая система Linux, использовалась в ранних версиях Linux.
- ext2 — стандартная, но уже устаревшая файловая система Linux. Долгое время использовалась практически во всех дистрибутивах по умолчанию, но была заменена файловой системой ext3.
- ext3 — модифицированная версия файловой системы ext2, но с поддержкой журнала (о том, что это такое, мы поговорим позже), существенно повышающего надежность файловой системы. Максимальный размер раздела с файловой системой ext3 — 4 Тбайт<sup>1</sup>, хотя ядро Linux версии 2.6 поддерживает максимальный размер раздела до 16 Тбайт. Максимальный размер файла — 1 Тбайт.

---

<sup>1</sup> Терабайт (Тбайт) — единица количества информации, равная 1024 Гбайт. — Примеч. ред.

- ext4 — новейшая файловая система Linux. Поддержка ext4 как стабильной файловой системы появилась в ядре Linux версии 2.6.28 — это самая новая версия ядра на момент написания этих строк. Если сравнивать эту файловую систему с ext3, то производительность и надежность новой файловой системы существенно увеличена, а максимальный размер раздела теперь равен 1024 Пбайт<sup>2</sup> (1 Эбайт<sup>3</sup>). Максимальный размер файла больше 2 Тбайт. Ресурс Phoronix ([www.phoronix.com](http://www.phoronix.com)) провел тестирование новой файловой системы на SSD-накопителе (такие накопители устанавливаются в современные нетбуки). Результат, как говорится, на лицо (рис. 21.1): ext4 почти в два раза превзошла файловые системы ext3, XFS, JFS и ReiserFS.
- ReiserFS — основная особенность этой файловой системы заключается в хранении в одном блоке нескольких маленьких файлов. Например, если у вас размер блока 4 Кбайт, то в него поместится до четырех файлов по одному килобайту каждый. Если у вас много маленьких файлов, то такая файловая система — просто находка, ведь она позволяет экономить дисковое пространство. Однако с большими файлами эта файловая система работает медленно, потому она чувствительна к сбоям и ее нужно регулярно дефрагментировать. И еще: если вы решили использовать эту файловую систему, обращайтесь к источнику бесперебойного питания, который предотвратит потерю данных в случае отключения электричества.
- JFS — разработка IBM, обладает высокой производительностью, но оптимизирована под сервер баз данных, поскольку размер блока небольшой — от 512 байт до 4 Кбайт. Если вам приходится работать с большими файлами, например с видео, то файловая система — это не очень удачный выбор.
- XFS — обладает относительно высокой производительностью — она быстрее, чем ext3, ReiserFS и JFS, но медленнее, чем ext4. Устанавливает большой размер блока — до 64 Кбайт, что позволяет ее использовать на графических станциях для обработки видео.

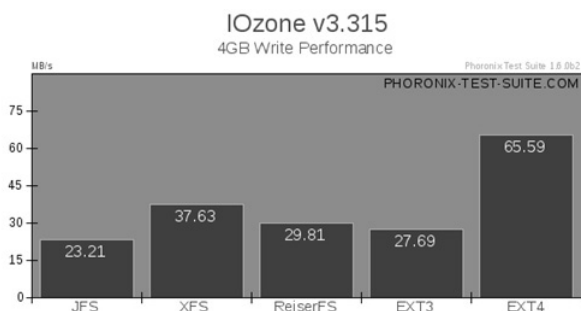


Рис. 21.1. Тест производительности файловых систем, проведенный ресурсом Phoronix

Какую файловую систему выбрать сегодня? Если любите экспериментировать и не планируете хранить на своем компьютере ничего важного, можно выбрать ext4. Ведь она обеспечивает высокую производительность. Но совсем недавно она была нестабиль-

<sup>2</sup> Петабайт (Пбайт) — единица количества информации, равная 1024 Тбайт. — Примеч. ред.

<sup>3</sup> Эксабайт (Эбайт) — единица количества информации, равная 1024 Пбайт. — Примеч. ред.

ной, поэтому если у вас хранится что-то важное, тогда лучше выбрать ext3. Если вы уже установили Linux и при установке были созданы ext3-разделы, то чуть ниже в этой главе будет показано, как перейти на ext4 без потери данных.

## 21.2. КОРНЕВАЯ ФАЙЛОВАЯ СИСТЕМА, СТАНДАРТНЫЕ КАТАЛОГИ

При установке Linux на выбранном вами разделе жесткого диска создается корневая файловая система Linux. Корневая файловая система содержит набор стандартных каталогов и утилит, без которых невозможна работа Linux.

Корневая файловая система обозначается как /. Обратите внимание: в Linux используется прямой слеш, а не обратный (\), как в Windows. Полный путь к файлу обязательно начинается с корневой файловой системы. Вот полный путь к файлу `report.doc`, который находится в домашнем каталоге пользователя `den`: `/home/den/report.doc`.

В табл. 21.1 приведено описание основных каталогов корневой файловой системы

**Таблица 21.1. Основные каталоги корневой файловой системы**

Каталог	Описание
/	Корневой каталог
/bin	Содержит стандартные утилиты Linux
/boot	Содержит конфигурационные файлы загрузчика GRUB, образы ядра, файлы <code>initrd</code>
/dev	Содержит файлы устройств. О том, что это такое, мы поговорим чуть позже
/etc	Содержит конфигурационные файлы операционной системы и всех сетевых служб. Данный каталог подобен реестру Windows, но в Windows общесистемные настройки хранятся в одном большом бинарном файле, а в Linux настройки хранятся в разных конфигурационных файлах, которые можно редактировать обычным текстовым редактором
/home	Содержит домашние каталоги пользователей. В домашних каталогах пользователей хранятся пользовательские файлы, а также пользовательские настройки различных программ
/lib	Здесь находятся различные библиотеки и модули ядра
/misc	В данном каталоге может быть все, что угодно
/mnt	Обычно в этом каталоге содержатся точки монтирования. О монтировании файловых систем мы тоже поговорим отдельно
/proc	Это не совсем обычный каталог, это каталог псевдофайловой системы <code>procfs</code> , которая используется для предоставления информации о процессах
/root	Каталог пользователя <code>root</code> (пользователь с максимальными полномочиями)
/sbin	Набор утилит для системного администрирования, запускать эти утилиты имеет право только <code>root</code>
/tmp	“Мусорка”, т.е. каталог, в котором хранятся временные файлы. Linux, в отличие от Windows, следит за чистотой и регулярно очищает этот каталог

Окончание табл. 21.1

Каталог	Описание
<code>/usr</code>	Содержит пользовательские программы. По размеру это один из самых больших каталогов файловой системы. В этот каталог устанавливаются практически все программы. Также в этом каталоге находятся вспомогательные файлы, необходимые для работы установленных программ. Грубо, конечно, но данный каталог можно сравнить с каталогом Program Files в Windows
<code>/var</code>	Данные системы, которые постоянно изменяются, например, очередь печати, почтовые ящики и т.д.

## 21.3. ИМЕНА ФАЙЛОВ

Как все мы знаем, в Windows имя файла состоит из двух частей — собственно из имени файла и расширения, позволяющего установить тип файла. В Linux вообще нет такого понятия, как расширение файла. Если в Windows в имени `index.html` последние четыре символа — это расширение файла, то в Linux — это просто четыре символа после последней точки в имени файла.

Относительно имен файлов вам нужно знать еще следующие факты.

- Имя файла не может содержать символы `/ \ ? < > * " |`.
- Максимальная длина имени файла — 254 символа.
- Linux чувствительна к регистру букв в имени файла, например, `INDEX.html` и `Index.html` — это два разных файла.
- Желательно не использовать в именах файлов русские буквы. Linux нормально работает с такими файлами, но могут возникнуть проблемы при передаче таких файлов в другую операционную систему, например в Windows.
- Разделение элементов пути к файлу осуществляется с помощью прямого слеша (`/`), а не обратного (`\`), как в Windows

## 21.4. ФАЙЛЫ УСТРОЙСТВ

Абстракция — это великая сила. Возьмем, например, DOS (именно DOS, потому что с нее началась история Microsoft). Разные разделы жесткого диска в DOS представляются как отдельные диски (в DOS они называются логическими дисками). В Windows то же самое — открываем окно Мой компьютер и видим несколько пиктограмм жестких дисков (рис. 21.2).

Каждый логический диск представляется как отдельный дисковый накопитель. Для пользователя нет разницы, куда он записывает данные — на отдельный физический диск или на раздел физического диска. Посмотрите на рис. 21.2: на самом деле у меня всего два физических диска, а разделов существенно больше.

Linux по уровню абстракции опередила Windows. В Linux каждое отдельное устройство представляется в виде файла файловой системы, и к устройству вы можете обращаться как к обычному файлу! Файлы устройств хранятся в каталоге `/dev`. В табл. 21.2 приведены некоторые популярные файлы из каталога `/dev`:

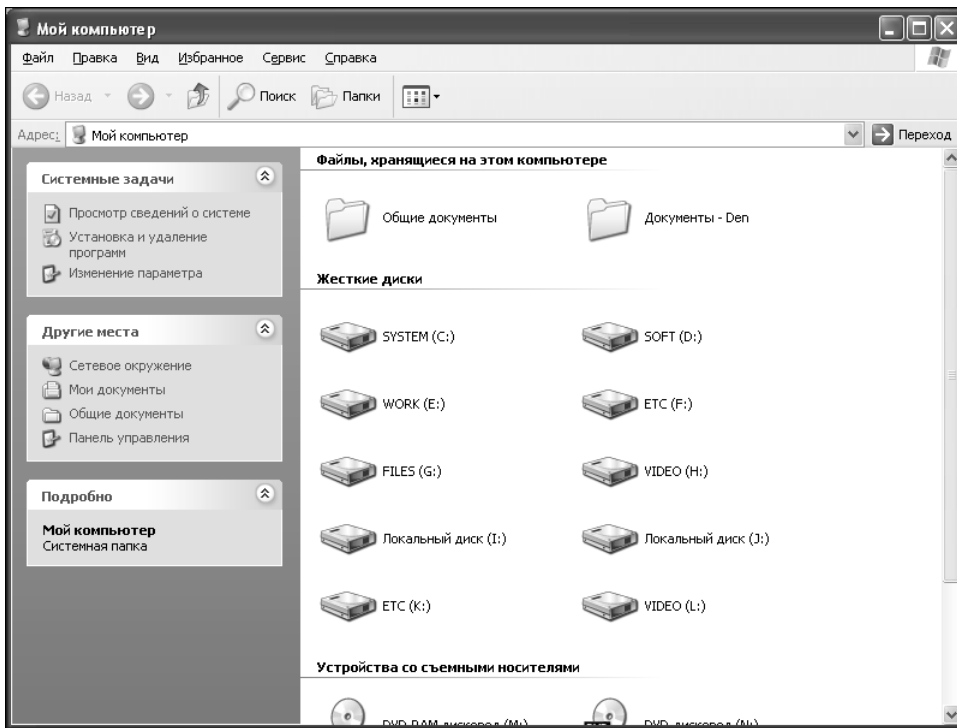


Рис. 21.2. Мой компьютер

Таблица 21.2. Некоторые файлы устройств

Файл	Описание
<code>/dev/modem</code>	Файл модема, но обычно это ссылка или на устройство <code>ttySn</code> , или на <code>ttyUSBn</code>
<code>/dev/ttySn</code> , где $n$ — номер	Файл последовательного порта, <code>ttyS0</code> = COM1, <code>ttyS1</code> = COM2
<code>/dev/ttyUSBn</code> , где $n$ — номер	Обычно используется в случае подключения USB-модема
<code>/dev/scdN</code> , где $N$ — номер	Имя файла привода CD/DVD, $N$ — номер привода
<code>/dev/srN</code> , где $N$ — номер	Ссылка на устройство <code>/dev/scdN</code>
<code>/dev/fdN</code> , где $N$ — номер	Файл дисковода гибких дисков: <code>/dev/fd0</code> — A:, <code>/dev/fd1</code> — B:

Сложнее всего ситуация обстоит с именами жестких дисков. Раньше IDE (PATA) жесткие диски именовались `/dev/hda`, `/dev/hdb`, `/dev/hdc` и `/dev/hdd` — в зависимости от типа подключения диска к интерфейсу (Primary Master, Primary Slave, Secondary Master, Secondary Slave соответственно). SATA/SCSI-диски назывались `/dev/sdX`, где  $X$  — это символ, определяющий порядок подключения диска к интерфейсу, например,  $a$  — первый диск,  $b$  — второй и т.д.

Но потом был разработан новый менеджер устройств — `udev` (он появился в ядре 2.6). Теперь все жесткие диски, вне зависимости от интерфейса (PATA или SATA), называются `/dev/sdX`, где `X` — буква, определяющая порядок подключения диска. Как теперь не запутаться и зачем вообще нужен был `udev`?

Сейчас разберемся. Начнем с того, что практически все современные жесткие диски — SATA, поэтому, если у вас именно такой диск, то особых изменений вы и не заметили. Ваш жесткий диск как назывался `/dev/sda`, так и будет называться.

Однако на самом деле, `UUID` — штука полезная. Представим, что у вас есть PATA-винчестер, а `udev` — нет (допустим, у вас старое ядро). Вы подключили винчестер как первичный мастер и установили на его 5-й раздел Linux, т.е. имя раздела с корневой файловой системой Linux — `/dev/hda5`. Потом вы отправились к соседу за фильмами с этим винчестером. Там вы его подключили как первичный подчиненный (`/dev/hdb`), но когда пришли домой, забыли вернуть джампер выбора режима работы винчестера на место и ошибочно подключили жесткий диск. В итоге наш раздел `/dev/hda5` превратился в `/dev/hdb5`, но система-то ищет `/dev/hda5` и, следовательно, не может загрузиться. А вы ломаете голову, что же случилось.

При использовании `UUID` каждому разделу присваивается его уникальный идентификатор. `UUID` постоянен и не изменится, если вы неправильно подключите жесткий диск. Как выглядят идентификаторы моих разделов для корневой файловой системы и раздела для подкачки, можно узнать из листинга 21.1.

### Листинг 21.1. Фрагмент файла `/etc/fstab` (хранит информацию об автоматически монтируемых файловых системах)

```
# /dev/sda5
UUID=579f797c-944d-4c28-a720-249730705714 / ext3
relatime,errors=remount-ro 0 1
# /dev/sda7
UUID=171e2584-ff16-11dc-843e-f381dfd10b3b none swap sw
```

Узнать, какой идентификатор какому диску соответствует, можно с помощью команды:

```
ls -l /dev/disk/by-uuid/
```

Вообще, `udev` предусматривает несколько схем именования дисков:

```
ls -l /dev/disk/by-uuid   ïï UUID
ls -l /dev/disk/by-id     ïï ID
ls -l /dev/disk/by-label ïï ïâðêâ
ls -l /dev/disk/by-path  ïï ïóðè
```

#### Примечание

Установить метку `ext2/ext3`-раздела можно с помощью команды `sudo e2label /dev/XXX <метка>`.

Введите эти команды и проанализируйте вывод — это будет вашим “домашним заданием”.

## 21.5. ДОМАШНИЙ КАТАЛОГ ПОЛЬЗОВАТЕЛЯ

В каталоге `/home` хранятся домашние каталоги пользователей. Домашний каталог пользователя называется так: `/home/<имя_пользователя>`.

Например:

```
/home/den
```

Домашний каталог пользователя также обозначается как `~`. Например, если вы хотите перейти в свой домашний каталог, то можно вместо `/home/<имя_пользователя>` указать просто `~`:

```
cd /home/den
cd ~
```

Данные команды равносильны при условии, что вы являетесь пользователем `den`.

## 21.6. КОМАНДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ И КАТАЛОГАМИ В КОНСОЛИ

Сейчас, когда в Linux есть отличный графический интерфейс, консолью приходится пользоваться все реже и реже, но в некоторых случаях (сбой X-сервера, установка программ из исходного кода) без консоли (или терминала) вам не обойтись. А чтобы эффективно работать в консоли, нужно знать команды Linux. Команд очень много, поэтому все рассматривать мы не будем, а ограничимся командами для работы с файлами (табл. 21.3) и каталогами (табл. 21.4).

**Таблица 21.3. Команды для работы с файлами**

Команда	Описание
<code>cat имя_файла</code>	Просмотр текстового файла. Обычно используется в паре с командой <code>more</code> для организации постраничного вывода: <code>cat &lt;имя_файла&gt;   more</code>
<code>tac имя_файла</code>	Вывод файла в обратном порядке. Полезна для просмотра файлов журналов, в которых самая свежая информация находится в конце файла. Для организации постраничного просмотра нужно использовать программу <code>more</code> или <code>less</code> : <code>tac &lt;имя_файла&gt;   less</code>
<code>touch имя_файла</code>	Создает пустой файл
<code>echo строка</code>	Выводит указанную строку. С помощью этой команды можно добавить строку в имеющийся файл или же создать новый файл с заданной строкой: <code>echo строка &gt; новый_файл</code> <code>echo строка &gt;&gt; существующий_файл</code> В первом случае, если файл существует, то он будет перезаписан
<code>cp источник_назначение</code>	Копирует файл <i>источник</i> в файл <i>назначение</i> . Если файл <i>назначение</i> существует, то программа спросит вас, нужно ли его перезаписать или нет

Окончание табл. 21.3

Команда	Описание
<code>mv источник назначение</code>	Перемещает файл <i>источник</i> в файл <i>назначение</i> . Если файл <i>назначение</i> существует, то программа спросит вас, нужно ли его перезаписать или нет. Данную команду можно использовать также для переименования файла
<code>less файл</code>	Постраничный вывод файла
<code>locate файл</code>	Выполняет поиск файла
<code>ln файл ссылка</code>	Используется для создания ссылки на указанный файл. Ссылка — это еще одно имя файла. Файл нельзя удалить, если на него указывает хотя бы одна ссылка
<code>rm файл</code>	Удаляет файл
<code>which имполнимый_файл</code>	Используется для поиска программы в каталогах, указанных в переменной окружения <code>PATH</code>
<code>chmod режим файла</code>	Изменение прав доступа к файлу или каталогу. Подробнее см. <code>man chmod</code>
<code>chown пользователь файла</code>	Изменение владельца файла. Подробнее см. <code>man chown</code>

**Таблица 21.4. Команды для работы с каталогами**

Команда	Описание
<code>cd каталог</code>	Изменение каталога
<code>mkdir каталог</code>	Создание каталога
<code>rmdir каталог</code>	Удаление пустого каталога
<code>rm -r каталог</code>	Рекурсивное удаление непустого каталога (будут удалены все подкаталоги заданного каталога и сам каталог)
<code>ls каталог</code>	Вывод содержимого каталога

При работе с каталогами вы должны знать три специальных имени каталогов:

- `~` — домашний каталог пользователя;
- `.` — текущий каталог;
- `..` — родительский каталог.

## 21.7. Права доступа к файлам. Владелец файла

Первоначально в DOS, а затем и в Windows не было таких понятий, как “владелец файла” и “права доступа к файлу”. Права доступа к файлам и каталогам появились только в файловой системе NTFS, но фактически NTFS пришла в массы в 2001-2002 годы. Сейчас поясню, почему так поздно. Файловая система NTFS поддерживалась только операционной системой



Windows NT, которая не была столь распространена, как Windows 95/98. Windows 2000, в которой тоже была поддержка NTFS, так же не была очень распространенной. Наибольшей популярностью пользовалась Windows XP, которая поддерживала NTFS. Но многие пользователи просто не использовали всех возможностей NTFS или вообще ее не использовали (ведь Windows XP также поддерживала FAT32): кто — по привычке к FAT, а кто — по незнанию. Поэтому понятие прав доступа будет новым для Windows-пользователей.

В Linux всех пользователей можно разделить на три большие группы.

- **Владелец.** К этой группе принадлежит всего один пользователь, создавший файл или каталог.
- **Группа.** Члены группы, к которой принадлежит владелец. Например, может быть создана группа `users`, к которой будут принадлежать пользователи `ivanov`, `petrov` и `sidorov`. Подробнее о группах, а также о правах членов группы мы поговорим в следующей главе.
- **Прочие.** Все остальные пользователи, которые не являются владельцем файла и которые не входят в группу, к которой принадлежит владелец файла.

Вот что нужно помнить о правах доступа в Linux.

- Вы, владелец файла, можете назначить права доступа к собственным файлам и каталогам для каждой группы пользователей.
- Пользователь может создавать файлы только в пределах домашнего каталога. Доступ к остальным каталогам файловой системы ограничен (как правило, только чтением, без возможности создания новых файлов и изменения уже имеющихся).
- Изменить права доступа к файлу может владелец или пользователь `root`.
- Пользователь `root` обладает максимальными привилегиями, поэтому он может получить полный доступ к любому файлу и каталогу.

Установить права доступа к файлу или каталогу очень просто. Откройте свой домашний каталог, щелкните правой кнопкой на любом файле и выберите команду **Свойства**. В появившемся окне перейдите на вкладку **Права** (рис. 21.3). На этой вкладке вы можете установить права доступа для владельца, группы владельца (если владелец является членом нескольких групп, можно установить отдельные права доступа для каждой группы) и для остальных пользователей. Права доступа могут быть такими:

- **Чтение и запись** — полный доступ к файлу или каталогу;
- **Только чтение** — файл доступен только для чтения;
- **Нет** — в доступе к файлу будут отказано (как в случае открытия файла в режиме только чтение, так и в режиме записи).

Право на выполнение означает, что файл можно будет запускать (как программу). Понятно, что право на выполнение нужно устанавливать только для программ или сценариев командного интерпретатора, иначе все равно система не сможет выполнить файл другого типа. Ну, как, например, можно запустить файл формата MS Word?

Право выполнения для каталога означает, что можно просматривать содержимое этого каталога.

А теперь рассмотрим, как установить права с помощью командной строки Linux. Как мы уже выяснили, в Linux существует три права доступа: чтение (`r` — от *read*), запись (`w` — от *write*), выполнение (`x` от *eXecute*).

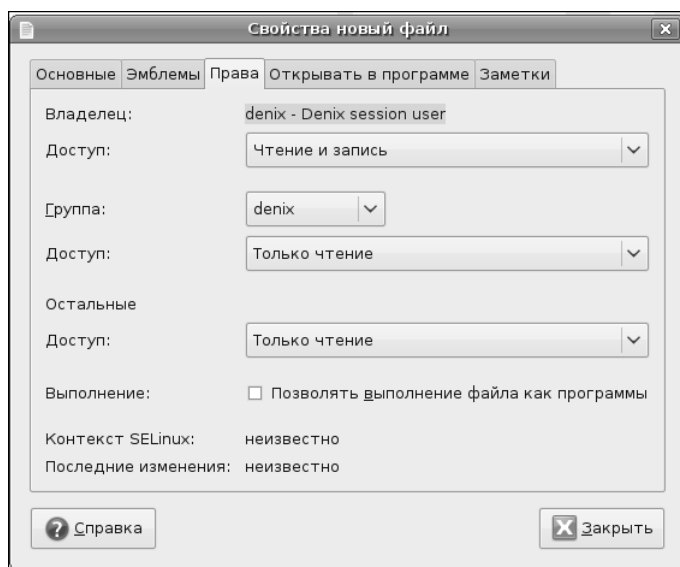


Рис. 21.3. Установка прав доступа

Вы можете установить разные наборы прав доступа для владельца файла, группы владельца и для всех остальных пользователей. Сейчас продемонстрирую, как это выглядит на практике. Откройте терминал и введите команду (имя файла можно ввести любое, главное, чтобы он существовал):

```
ls -l <имя_файла>
-rw-r--r-- 1 den den-group 6051 Ноя 28 14:44 <имя_файла>
```

Нас больше всего интересует строка `-rw-r--r--`. Это и есть права доступа. Начнем по порядку: первый символ (`-`) означает, что перед нами файл, а не каталог. Если бы перед нами был каталог, то вместо тире стоял бы символ `d` (от *directory*).

Идем дальше. Первая тройка символов (после тире) является правами доступа для владельца. Владельцу (пользователь `den`) разрешено читать (`r`), изменять (`w`) файл, но запрещено запускать его на выполнение (третий символ — тире, а не `x`).

Следующие два набора означают права доступа для членов группы владельца (группа `den-group`) и остальных пользователей (они не являются членами группы и владельцем файла). И те, и другие имеют право только читать файл (`r--`).

Нужно отметить, что в Linux права доступа совсем не касаются пользователя `root`. Какие бы права доступа ни установили, пользователь `root` все равно получит доступ к файлу.

Наверняка вы уже побывали не на одном Linux-форуме и видели “магические” команды вроде:

```
chmod 666 файл
chmod 777 каталог
```

На самом деле — это никакие не заклинания, а обычные команды установки прав доступа. Команда `chmod` используется для установки прав доступа к файлу и каталогу. Сначала указываются права доступа, а затем имя файла или каталога. Помните, что вы можете установить права доступа только к “своим” файлам, т.е. к файлам, владельцем ко-

торых вы являетесь. Пользователь `root` имеет право изменить права доступа абсолютно любого файла или каталога.

Итак, разберемся с магией цифр. Посмотрим на набор прав:

```
rw- r-- r--
```

Теперь заменим символы на 1, а тире — на 0. Получим:

```
110 100 100
```

А сейчас заглянем в табличку восьмеричных значений (табл. 21.5).

**Таблица 21.5. Восьмеричная система счисления**

Двоичная система	Восьмеричная
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Подставим полученные двоичные значения в таблицу и в итоге получим набор прав доступа `644` (владельцу разрешено чтение и запись, всем остальным — только чтение). Набор `666` соответствует правам доступа:

```
rw- rw- rw-
```

Это самые небезопасные права доступа, позволяющие всем желающим делать с файлом все, что они хотят (вплоть до удаления). Поэтому права `666` лучше не устанавливать для файлов, содержащих что-то важное. Для таких файлов больше подойдут права `600`, разрешающие операции с файлом только владельцу.

Права `777` обычно устанавливаются для каталогов. Это то же, что и `666`, но для каталога.

Команда `chown` используется для изменения владельца файла. Например:

```
chown user report.txt
```

После выполнения этой команды пользователь `user` станет владельцем файла `report.txt`. Вы можете изменить владельца файла только для файлов, владельцем которых вы являетесь. Пользователь `root` может изменить владельца для любого файла.

## 21.8. ЖЕСТКИЕ И СИМВОЛИЧЕСКИЕ ССЫЛКИ

В Linux вы можете создавать ссылки на файл (или каталог). По сути, один и тот же файл может храниться в файловой системе под разными именами — допускается неограниченное количество ссылок на файл.

Ссылки бывают жесткими и символическими. Жесткие ссылки “жестко” привязаны к файлу: нельзя удалить файл, пока на него указывает хотя бы одна ссылка. Если на файл указыва-

ют только символичные ссылки, вы можете его удалить, потом ссылки окажутся битыми — они будут ссылаться “в никуда”. Жесткие ссылки не могут указывать на файл, который физически находится за пределами файловой системы, например, находится на другом разделе.

Для создания ссылок используется команда `ln`, ее синтаксис следующий:

```
ln [-s] файл ссылка
```

Если указан параметр `-s`, то будет создана символическая ссылка:

```
ln report.doc hard_link.doc
ln -s report.doc sym_link
```

## 21.9. МОНТИРОВАНИЕ

### 21.9.1. Команда `mount`

Особого внимания заслуживает операция монтирования. На вашем жестком диске, кроме Linux-разделов, есть разделы других операционных систем (например, Windows). Для того чтобы получить доступ к файлам, находящимся на этих разделах, нужно примонтировать данные разделы к корневой файловой системе.

Точкой монтирования называется каталог, к которому подмонтирована другая файловая система. Через точку монтирования осуществляется доступ к файлам и каталогам подмонтированного раздела.

Монтирование выполняется с помощью команды `mount` (ее имеет право вводить только пользователь `root`, поэтому запуск команды `mount` должен осуществляться через команду `sudo`):

```
sudo mount раздел точка_монтирования
```

Например:

```
sudo mount /dev/sda1 /mnt/disk_c
```

Точка монтирования (каталог `/mnt/disk_c` в нашем случае) должна существовать на момент монтирования.

Приведенная выше команда монтирует первый раздел на первом IDE-диске (обычно это диск C:) к каталогу `/mnt/disk_c`. Далее вы можете обращаться к файлам и каталогам диска C: через каталог `/mnt/disk_c`:

```
ls /mnt/disk_c
```

Обычно точки монтирования принято создавать в каталоге `/mnt`, но это не принципиально: при желании вы можете подмонтировать раздел к любому другому каталогу.

### 21.9.2. Файл `/etc/fstab`

В Ubuntu монтирование остальных разделов вашего жесткого диска (в том числе Windows-разделов) осуществляется автоматически. Чтобы примонтировать тот или иной раздел, нужно выбрать команду меню **Переход**⇒<Метка диска>. После монтирования раздела на рабочем столе появится пиктограмма для быстрого доступа к нему.

Однако так было не всегда. Раньше, если вы часто обращались к тому или иному Windows-разделу, его нужно было прописать в файле `/etc/fstab`. Тогда заданный раз-

дел монтировался автоматически при загрузке Linux. Сейчас мы рассмотрим формат этого файла. Читать этот пункт или нет — решать вам. Учитывая, что сейчас все упрощено, можно и пропустить, но с другой стороны — это интересно.

Формат файла `/etc/fstab` следующий:

```
Раздел      Точка_монтирования  ФС  Опции      ФР  ФП
```

Здесь `ФС` — тип файловой системы, `Опции` — опции файловой системы, `ФР` — флаг резервного копирования, `ФП` — флаг проверки.

Теперь обо всем по порядку. Предположим, что нам нужно обеспечить автоматическое монтирование раздела `/dev/sda1` (обычно это диск `C:`). Первым делом нужно создать каталог `/mnt/c`, который будет использоваться в качестве точки монтирования. Создать данный каталог имеет право только пользователь `root`, поэтому откройте терминал и введите команду:

```
sudo mkdir /mnt/c
```

Первая обеспечит необходимые для выполнения операции полномочия, а вторая создаст нужный нам каталог.

Потом нужно определить тип файловой системы. Здесь все просто. Скорее всего, вам нужно будет примонтировать `Windows`-раздел, поэтому вариантов будет всего три:

- `vfat` — обычная файловая система `Windows` (`FAT32`);
- `ntfs` — стандартный драйвер для работы с файловой системой `NTFS`, не поддерживает запись данных на раздел;
- `ntfs-3g` — более современный драйвер для `NTFS`, поддерживающий запись `NTFS`-разделов. Модуль ядра `ntfs-3g` есть только в самых современных дистрибутивах, но они и так умеют автоматически монтировать разделы жесткого диска. В более старых дистрибутивах модуль `ntfs-3g` нужно загрузить и установить самостоятельно с сайта [www.ntfs-3g.org](http://www.ntfs-3g.org).

Если вы не можете определить тип файловой системы, попробуйте использовать тип `auto` — автоопределение, но вполне возможно, что в этом случае возникнут некоторые ограничения, например, будет отсутствовать запись с `NTFS`.

Набор опций для `vfat` будет таким:

```
umask=0,utf8
```

В случае с `NTFS` нужно указать следующие опции:

```
defaults,nls=utf8,umask=007,gid=46
```

Последние два флага можно установить в `0`. Флаг резервного копирования вообще оставлен из соображений совместимости — еще со времен, когда резервное копирование выполнялось утилитой `dump`. А проверка файловой системы нужна только для файловых систем `Linux`.

Теперь самое интересное. Для монтирования раздела `/dev/sda1` к каталогу `/mnt/c` нужно добавить в файл `/etc/fstab` одну из следующих строчек (в зависимости от используемой файловой системы):

```
/dev/sda1 /mnt/c vfat umask=0,utf8 0
/dev/sda1 /mnt/c ntfs-3g defaults,utf8,umask=007,gid=46 0 0
```

### Примечание

Вместо имени файла раздела можно указать его `UUID` так: `UUID=579f797c-944d-4c28-a720-249730705714`.

Напомню, что для редактирования файла `/etc/fstab` нужны полномочия `root`, а само редактирование удобно проводить в редакторе `gedit`:

```
gedit /etc/fstab
```

Чтобы изменения вступили в силу, нужно или перезагрузить компьютер или ввести команду:

```
# mount -a
```

### 21.9.3. Проблема с монтированием USB-диска, отформатированного как NTFS

Не получается автоматически подмонтировать USB-диск, который отформатирован как NTFS? Скорее всего, причина заключается в том, что он не был правильно размонтирован в прошлый раз. Но это легко поправить. Откройте терминал и введите команды:

```
sudo mkdir /media/usb
sudo mount -t ntfs-3g <имя> /media/usb -o force
```

Первая команда создает каталог `/media/usb`, а вторая — монтирует к ней тот раздел, который не получилось смонтировать автоматически. Обычно это `/dev/sdb1`, при условии, что у вас один жесткий диск (так как имя `/dev/sda` зарезервировано за жестким диском).

Особенность данной команды — использование опции `force`, что и позволяет смонтировать носитель. После монтирования вы можете обратиться к файлам носителя через каталог `/media/usb`. Если использование этого каталога вам не удобно, тогда размонтируйте носитель, отключите его и подключите заново — все должно работать. Для размонтирования носителя нужно ввести команду:

```
sudo umount /media/usb
```

### 21.9.4. Автоматическое монтирование сменных носителей

Сменные носители (оптические диски (CD/DVD), Flash-диски, цифровые камеры и т.д.) монтируются также автоматически. Как только вы вставите в привод оптический диск или подключите к компьютеру Flash-накопитель (цифровой фотоаппарат, мобильный телефон), на рабочем столе появится пиктограмма для быстрого доступа к носителю данных и откроется окно Обзорателя файлов (рис. 21.4).

Нельзя извлечь диск из привода до тех пор, пока не размонтируете диск. А размонтировать диск нельзя до тех пор, пока его использует хотя бы одна программа. Итак, чтобы извлечь диск из привода, нужно выполнить следующие действия:

- закрыть все программы (например, Обзоратель файлов), которые используют диск;
- щелкнуть правой кнопкой на значке диска на рабочем столе и выполнить команду **Извлечь том**. Также доступна команда **Отсоединить том**. Первая команда отличается от второй (**Отсоединить том**) тем, что проводит не только размонтирование устройства, но и извлечение лотка привода CD/DVD.

Что же касается Flash-дисков, то, хотя их можно извлечь вручную, настоятельно рекомендуется перед физическим извлечением из USB-порта выполнить команду **Отсоединить том**, иначе система не гарантирует сохранность записанных во время работы данных на флешку.

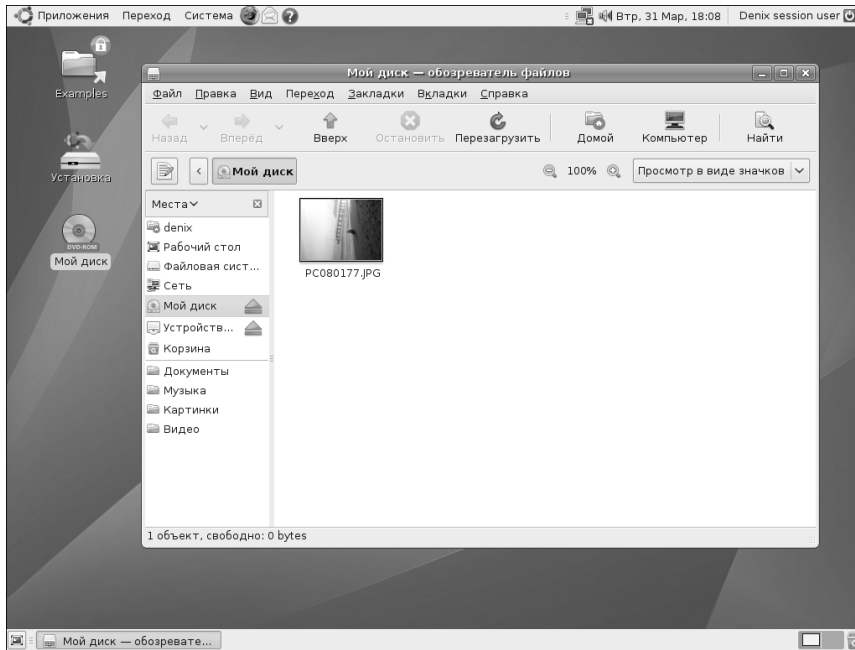


Рис. 21.4. Вставлен DVD-диск

### 21.9.5. Не могу извлечь диск!

Пользователями Ubuntu был замечен небольшой глюк. Вы отсоединили том, нажимаете кнопку Eject на вашем приводе CD/DVD, лоток выдвигается, но сразу же задвигается обратно — диск достать вы не успеете. Чтобы “вылечить” систему, откройте файл `/etc/udev/rules.d/60-persistent-storage.rules`:

```
sudo gedit /etc/udev/rules.d/60-persistent-storage.rules
```

Найдите в нем следующий фрагмент:

```
# skip unpartitioned removable media devices from drivers which do not
# send "change" events
ENV{DEVTYPE}=="disk", KERNEL!="sd*|sr*", ATTR{removable}=="1",
GOTO="persistent_storage_end"
# import filesystem metadata
IMPORT{program}="vol_id --export $tempnode"
```

После второй строки добавьте следующую строку:

```
ENV{DEVTYPE}=="disk", KERNEL=="sr*", ATTR{removable}=="1",
GOTO="persistent_storage_end"
```

У вас должно получиться следующее:

```
# skip unpartitioned removable media devices from drivers which do not
# send "change" events
ENV{DEVTYPE}=="disk", KERNEL!="sd*|sr*", ATTR{removable}=="1",
GOTO="persistent_storage_end"
+ENV{DEVTYPE}=="disk", KERNEL=="sr*", ATTR{removable}=="1",
```

```
GOTO="persistent_storage_end"
# import filesystem metadata
IMPORT{program}="vol_id --export $tempnode"
```

Сохраните файл и перезагрузите компьютер.

## 21.10. ЖУРНАЛИРУЕМАЯ ФАЙЛОВАЯ СИСТЕМА. ПАРАМЕТРЫ ЖУРНАЛА

Файловая система Linux является *журналируемой*. Сейчас поясню, что это такое. Перед выполнением каких-либо операций файловая система создает “план действий”, который называется в журнал. Затем она действует по четко составленному плану.

Если произошел сбой, например, элементарное отключение питания, то при следующем запуске проводится проверка журнала и попытка восстановить систему до нужного состояния.

Журнал существенно повышает надежность файловой системы. Практически все современные файловые системы для Linux/Unix являются журналируемыми.

Файловая система ext3, которая используется в любом современном дистрибутиве по умолчанию, имеет три режима работы журнала:

- `ordered` — в данном режиме в журнал заносится информация только об изменении служебных данных файловой системы. Режим используется по умолчанию, это компромисс между отказоустойчивостью и скоростью;
- `journal` — используйте, если хотите минимизировать потери данных в случае сбоя. В этом режиме в журнал записываются и пользовательские данные, и служебные данные файловой системы. Самый медленный режим;
- `writeback` — самый быстрый, но самый бесполезный, толку от него мало.

Установить режим работы журнала можно с помощью опции файловой системы `data` (в файле `/etc/fstab`):

```
раздел точка_монтирования ext 3 data=режим 1 1
```

Режим работы журнала можно задавать только для файловой системы ext3!

Пример:

```
/dev/sda5 /ext3 defaults,data=journal 1 1
```

## 21.11. ПРОВЕРКА ФАЙЛОВОЙ СИСТЕМЫ

Для проверки файловой системы используется программа `fsck` (сокращение от *file system check*). С помощью данной программы вы можете проверять не только файловые системы Linux, но и Windows-разделы (только `vfat`). Единственное требование — перед началом проверки проверяемая файловая система должна быть размонтирована. Поэтому последовательность действий для проверки файловой системы будет следующей:

```
sudo umount <имя_раздела>
sudo <имя_раздела>
```

Например, для проверки `/dev/sda1` можно использовать команды:

```
sudo umount /dev/sda1
sudo fsck /dev/sda1
```



## 21.12. ПРОГРАММА GPARTED

В Ubuntu есть превосходный редактор разделов диска, умеющий создавать, удалять, изменять размер и перемещать разделы — программа `gparted`. Редактор разделов по умолчанию не устанавливается (исключение — дистрибутив `Debian`, в котором `gparted` установлена по умолчанию). Для его установки нужно подключиться к Интернету и ввести команду:

```
sudo apt-get install gparted
```

После этого запустить программу можно командой меню Система⇒Администрирование⇒Редактор разделов (рис. 21.5).

Подробно рассматривать программу мы не будем, поскольку ее интерфейс интуитивно понятен. Для создания нового раздела нужно щелкнуть на кнопке **Создать**, а чтобы было место для создания раздела, перед этим нужно или удалить один из разделов (кнопка **Удалить**), или изменить раздел, уменьшив один из разделов (кнопка **Изменить размер или переместить**).

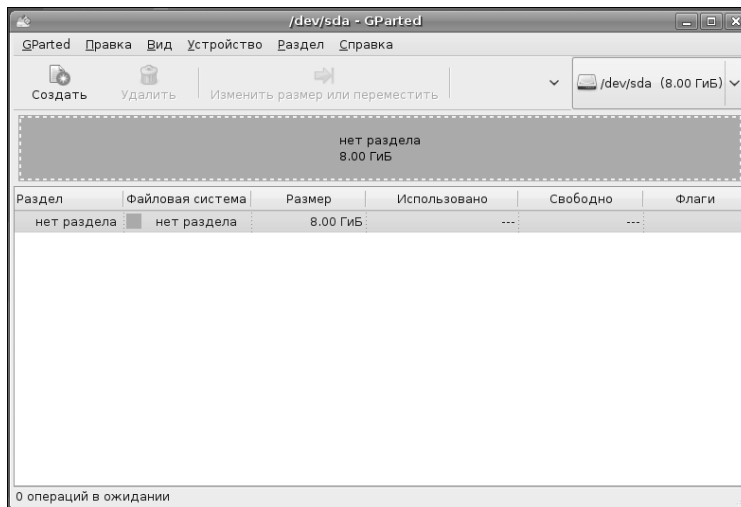


Рис. 21.5. Редактор разделов

## 21.13. ПЕРЕХОД НА EXT4 БЕЗ ПОТЕРИ ДАННЫХ

Сейчас мы разберемся, как перейти на `ext4` без потери данных. Однако сразу должен вас предупредить: после выполнения приведенных действий ваш `ext3`-раздел больше нельзя будет смонтировать как `ext3` — только как `ext4`.

Откройте терминал и введите команду:

```
sudo tune2fs -O extents,uninit_bg,dir_index /dev/имя_устройства
```

На момент ввода этой команды устройство должно быть размонтировано. Если вы хотите преобразовать корневую файловую систему в `ext4`, то данную команду нужно вводить с `LiveCD`.

После этого нужно проверить файловую систему:

```
sudo fsck -pf /dev/имя_устройства
```

Затем можно смонтировать файловую систему так:

```
mount -t ext4 /dev/имя_устройства /точка_монтирования
mount -t ext4 /dev/disk/by-uuid/UUID-устройства /точка_монтирования
```

Если ваш раздел автоматически монтируется через файл `/etc/fstab`, не забудьте исправить файловую систему на ext4:

```
UUID=UUID-раздела /точка ext4 defaults,errors=remount-ro,relatime 0 1
```

Если вы изменили тип файловой системы корневого раздела, то вам нужно отредактировать файл `/boot/grub/menu.lst` и добавить опцию `rootfstype=ext4` в список параметров ядра, например:

```
title          Ubuntu
root           (hd0,1)
kernel        /boot/vmlinuz-2.6.28.1 root=UUID=879f797c-944d-4c28-
a720-249730705714 ro quiet splash rootfstype=ext4
initrd        /boot/initrd.img-2.6.28.1
quiet
```

## 21.14. РАБОТА С ФАЙЛАМИ И КАТАЛОГАМИ СРЕДСТВАМИ GNOME

Меню **Переход** содержит команды перехода в различные места файловой системы: в домашний каталог, в каталог рабочего стола и другие места (рис. 21.6).

Зайдите в ваш домашний каталог. В нем вы найдете следующие стандартные каталоги (рис. 21.7).

- **Desktop.** Рабочий стол пользователя (все файлы и каталоги, созданные на рабочем столе, помещаются в этот каталог).
- **Документы.** Документы пользователя (OpenOffice сохраняет документы именно в этот каталог).
- **Музыка, Картинка, Видео.** Предназначены для хранения, соответственно, музыки, картинок и видео.
- **Общедоступная.** Публичный каталог, в него вы должны помещать файлы, которые должны быть доступны другим пользователям (но одного помещения файлов в этот каталог мало, нужно еще установить специальные права доступа, об этом мы поговорим чуть позже).
- **Шаблоны.** Предназначен для хранения различных шаблонов.

Но приведенные выше каталоги — это еще не все. В вашем домашнем каталоге содержится много скрытых файлов и каталогов. Обычно в скрытых файлах и каталогах хранятся настройки различных приложений, в том чис-

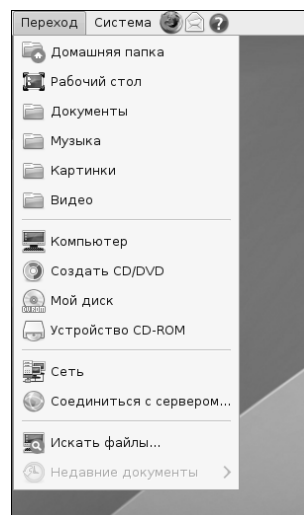


Рис. 21.6. Меню **Переход**

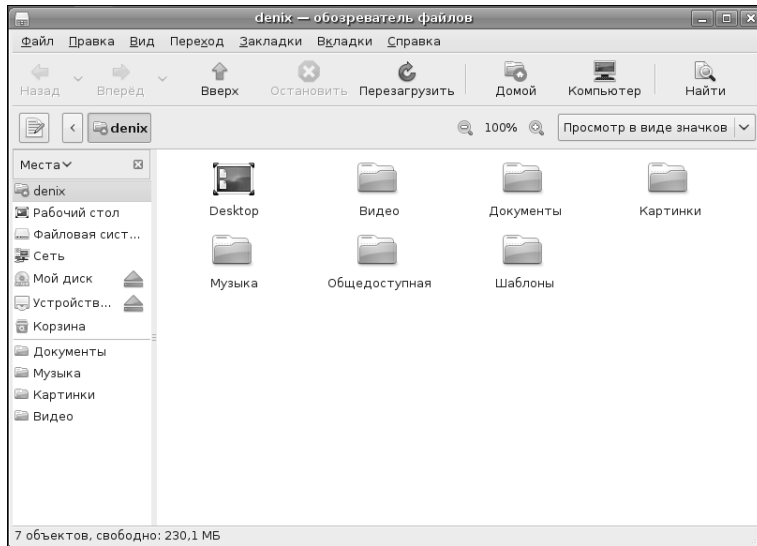


Рис. 21.7. Содержимое домашнего каталога (без скрытых файлов)

ле и пользовательские настройки GNOME. Для отображения скрытых файлов и каталогов выполните команду **Вид**⇒**Показывать скрытые файлы**. Скрытым считается файл или каталог, имя которого начинается с точки, например, `gimp` — это обычный каталог, а `.gimp` — скрытый каталог.

Как видно из рис. 21.8, для каждого приложения создан свой скрытый каталог, в котором хранится конфигурация программы.

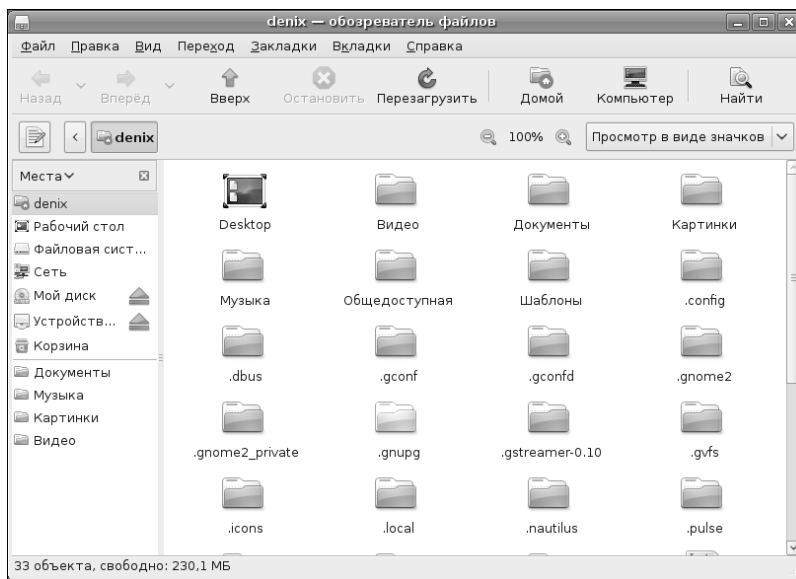


Рис. 21.8. Реальное содержимое домашнего каталога. Отображаются скрытые файлы

Работать с файлами и каталогами так же просто, как и в Windows. Щелкните на файле или каталоге, и появится контекстное меню, с помощью которого можно выполнить операции над файлом (рис. 21.9). В этом меню также есть команда **Создать архив**, что очень удобно. В Windows для того, чтобы появилась подобная команда, нужно установить программу-архиватор, а в Linux все есть сразу после установки операционной системы. Штатный архиватор в Linux поддерживает не только сугубо Linux-форматы, но и обычный формат ZIP, который можно распаковать в любой операционной системе.

Подробно рассматривать работу с файлами и каталогами в GNOME мы не будем — уж очень все просто.

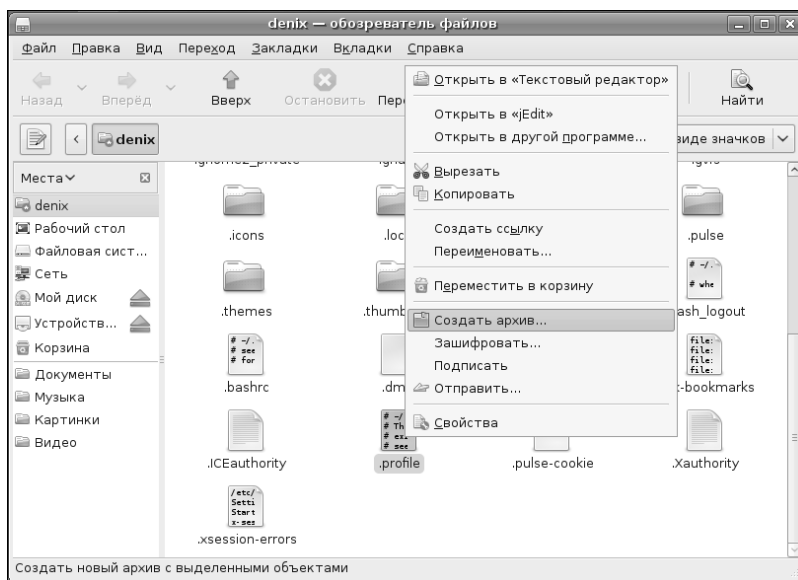


Рис. 21.9. Операции над файлом

## 21.15. КАК УЗНАТЬ, СКОЛЬКО МЕСТА ОСТАЛОСЬ НА ДИСКЕ?

Выполните команду меню **Приложения**⇒**Стандартные**⇒**Анализатор использования диска** или откройте терминал и введите команду **df**.