

Основные концепции

Что такое Linux

Linux — это бесплатная операционная система (ОС) с открытым исходным кодом, являющаяся альтернативой Microsoft Windows и Apple macOS. Под управлением операционной системы Linux действует большинство серверов в интернете. Linux работает в фоновом режиме на всех смартфонах Android и ноутбуках Chromebook, а также на миллионах подключенных к сети устройств, например маршрутизаторах, брандмауэрах и даже роботизированных системах доения коров (я серьезно). ОС также отлично запускается на ноутбуках и стационарных компьютерах.

Linux формируют четыре компонента (рис. 1.1).

Ядро

Низкоуровневое программное обеспечение. Управляет аппаратным обеспечением и основными функциями, например планированием процессов или работой в сети. Обычно пользователи не взаимодействуют с ядром напрямую.

Программное обеспечение

Тысячи программ для работы с файлами, редактирования текста, разработки ПО, просмотра веб-страниц, аудио, видео, шифрования, математических вычислений... Все эти программы взаимодействуют с ядром. Программы, запускаемые в оболочке командной строки, называются *командами*.



Рис. 1.1. Четыре основных компонента Linux. Низкоуровневые функции ядра вызываются программами, которые вызываются в оболочке. В свою очередь, оболочка может быть запущена из графической среды Рабочего стола

Оболочка

Используется для выполнения команд и отображения результатов. В Linux есть множество командных оболочек с различными возможностями. В этой книге рассматривается оболочка `bash`, которая чаще всего применяется по умолчанию в учетных записях пользователей. Остальные оболочки — `dash`, `fish`, `ksh` (оболочка Korn), `tcsh` (оболочка TC), `zsh` (оболочка Z), а также `busybox`. Все они имеют схожие функции, но их применение может различаться.

Графическая среда Рабочего стола (опционально)

Пользовательский интерфейс (UI) с окнами, меню, значками, поддержкой мыши и другими привычными элементами графического интерфейса пользователя (GUI). Самыми популярными средами являются GNOME и KDE Plasma. Большинство приложений,

созданных для GNOME, могут работать в KDE, и наоборот¹.

Эта книга фокусируется на командной части Linux, а именно на программном обеспечении и оболочке. В Windows и macOS тоже есть интерфейс командной строки (`cmd` и `powershell` в Windows, `Terminal` в macOS), хотя большинство пользователей применяют графический интерфейс и никогда не видят командную строку. В Linux оболочка играет решающее значение: если вы используете ее без оболочки, то многое упускаете.

Linux очень легко модифицировать, так что вы можете найти сотни вариантов оболочки, подходящих для тех или иных нужд. Каждая версия называется *дистрибутивом*. Все дистрибутивы имеют общие компоненты, но могут выглядеть по-разному и включать в себя разное ПО. Самыми популярными дистрибутивами считаются Mint, Ubuntu, Manjaro, Arch, Gentoo, Red Hat и OpenSUSE. Материал данной книги можно применить к любому из них.

Запуск оболочки

Как запускается оболочка? Обычно Linux запускает ее автоматически, например, когда вы входите в систему с помощью `ssh` или аналогичной программы. Первое, что вы увидите, — приглашение оболочки, ожидающее команды.

Иногда оболочку приходится запускать вручную. Это типично для Рабочих столов со множеством значков и меню, в котором нет оболочки. Тогда вам понадобится приложение с графическим интерфейсом — *терминал*, или *терминальная программа*. С помощью терминала можно запустить оболочку в окне. Во врезке «Оболочка и терминал» далее я объясняю разницу между оболочками и терминалами.

¹ GNOME, KDE и другие графические среды используют общий оконный интерфейс, например X Window System или Wayland. Чтобы узнать, какая система у вас, выполните команду `$XDG_SESSION_TYPE`.

В каждом дистрибутиве с графической средой Рабочего стола есть хотя бы одна терминальная программа. Чтобы запустить ее, поищите приложение, значок или пункт меню с именем `Terminal`, `Konsole`, `xterm`, `gnome-terminal`, `uxterm` или подобным. Запустите это приложение — откроется терминал. В некоторых средах можно открыть его нажатием клавиш `Ctrl+Alt+T` (нажав и удерживая клавиши `Ctrl` и `Alt`, нажмите клавишу `T`).

Оболочка и терминал

Оболочка — это интерфейс командной строки для запуска команд путем ввода обычного текста. В командной строке отображается приглашение:

→

Терминал — это программа, открывающая новое окно и запускающая оболочку. Я показал терминал на рис. 1.2. Он добавляет меню, полосу прокрутки, функцию копирования и вставки, и также другие функции графического интерфейса, поддерживающие оболочку.

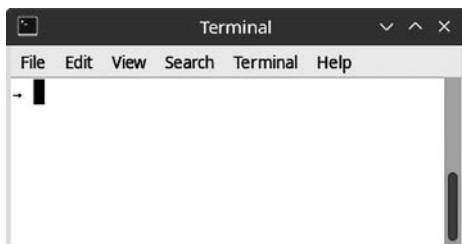


Рис. 1.2. Терминал открывает окно, в котором запускается оболочка

Активация командной строки

Чтобы показать работу Linux, попробуем выполнить в оболочке десять простых команд. Набирайте их *точно* как в книге, включая прописные и строчные буквы, пробелы

и символы после приглашения. В конце каждой команды нажмите клавишу **Enter**¹.

Отобразите календарь на ноябрь 2023 года:

```
→ cal nov 2023
      November 2023
Su Mo Tu We Th Fr Sa
                1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

Отобразим содержимое каталога `/bin`, в котором содержится множество команд:

```
→ ls /bin
bash      less      rm
bunzip2   lessecho  rmdir
busybox   lessfile  rnano
:
```

Подсчитайте количество видимых элементов в домашнем каталоге (здесь я использую переменную `HOME`, о которой расскажу позже):

```
→ ls $HOME | wc -l
8
```

Ваш результат может отличаться

Проверьте, сколько места занято на определенном разделе жесткого диска:

```
→ df -h /
Filesystem      Size    Used Avail Use% Mounted on
/dev/sdb1        78G     30G   48G   61% /
```

Откройте список процессов, запущенных на вашем компьютере (для выхода нажмите клавишу **Q**):

```
→ top -d1
```

¹ Если видите сообщение об ошибке «Команда не найдена», не волнуйтесь — возможно, команда не установлена в вашей системе. См. раздел «Установка ПО» главы 3.

Выведите содержимое файла `/etc/hosts`, в котором указаны имена и адреса компьютеров (на принтер по умолчанию, если он подключен):

→ `lpr /etc/hosts`

Узнайте, когда вы вошли в систему:

→ `last -1 $USER`

```
smith pts/7 :0 Tue Nov 10 20:12 still logged in
```

Скачайте файл `sample.pdf` с сайта книги в текущий каталог без веб-браузера:

→ `curl -O https://linuxpocketguide.com/sample.pdf`

Посмотрите, кому принадлежит доменное имя `oreilly.com` (для перехода на страницу нажмите клавишу Пробел, а для выхода — клавишу Q):

→ `whois oreilly.com | less`

```
Domain Name: OREILLY.COM
Registrar: GODADDY.COM, LLC
:
```

Наконец, очистите терминал:

→ `clear`

Поздравляю, теперь вы пользователь Linux!

Структура команд

Обычно команда Linux состоит из *имени программы*, *параметров* и *аргументов*:

```
wc -l файл
```

Имя программы (`wc`, сокращение от *word count* — подсчет слов) относится к программе на диске, которую оболочка находит и запускает. *Параметры*, обычно начинающиеся с дефиса, влияют на поведение программы. В предыдущей команде параметр `-l` указывает `wc` считать строки, а не слова. Аргумент *файл* определяет файл, с которым будет работать команда `wc`.

Команды могут иметь несколько параметров и аргументов. Параметры могут быть указаны по отдельности или с общим дефисом:

`wc -l -w файл`

Два отдельных параметра

`wc -lw файл`

Объединенные параметры, то же самое, что `-l -w`

ВНИМАНИЕ!

Некоторые программы не распознают объединенные параметры.

Вы также можете указать несколько аргументов:

`wc -l файл1 файл2`

Подсчет строк в двух файлах

Параметры не стандартизированы. Они могут состоять из одного дефиса и символа (например, `-l`), двух дефисов и слова (`--lines`) и т. п. Один и тот же параметр может иметь разное значение для разных программ: в команде `wc -l` параметр `-l` означает «строки текста», а в команде `ls -l` — «длинный вывод». Разные программы могут использовать разные параметры для обозначения одного и того же процесса. Так, параметры `-q` и `-s` означают «тихий запуск». После некоторых параметров следует указать значение, например `-s 10`. Пробел можно не вставлять (`-s10`).

Обычно в качестве аргументов выступают имена файлов для ввода (*источник*) или вывода (*назначение*), но вместо них могут использоваться и имена каталогов, пользователей, хостов, IP-адреса, выражения и прочие данные.

Команда, состоящая из одной программы с параметрами и аргументами, называется *простой*. Далее показана простая команда, которая выводит список пользователей, вошедших в систему на сервере Linux¹:

¹ Пользователь `silver`, указанный в списке дважды, запустил две интерактивные оболочки одновременно.

```
→ who
silver          :0      Sep 23 20:44
byrnes          pts/0    Sep 15 13:51
barrett         pts/1    Sep 22 21:15
silver          pts/2    Sep 22 21:18
```

Команда может вызывать несколько программ одновременно и даже настраивать их на взаимодействие друг с другом. Ниже приведен пример команды, передающей вывод команды `who` команде `wc`, которая подсчитывает строки текста. Результатом будет количество строк в выводе команды `who`:

```
→ who | wc -l
4
```

Вертикальная полоса, называемаяся *пайпом*, обеспечивает перенаправление данных между `who` и `wc`. Опытные пользователи Linux постоянно применяют подобные объединенные команды, называемые *конвейерами*.

Команды могут включать такие элементы языка программирования, как переменные, условия и циклы, о которых я рассказываю в разделе «Программирование с помощью сценариев командной оболочки» главы 6. Например, команда может гласить: «Запустить программу, записать вывод в указанный файл и, если возникнут ошибки, вывести сообщение с ошибкой».

Пользователи и суперпользователи

Linux — это многопользовательская ОС: несколько человек могут одновременно запустить несколько программ на одном компьютере. У каждого из них есть *имя пользователя*, например `smith` или `funkydance`. У каждого есть отдельное рабочее пространство (см. раздел «Домашние каталоги» далее в этой главе), чтобы не мешать друг другу.

Специальные пользователи с именем `root` — *суперпользователи* или администраторы — обладают всеми правами в системе. Суперпользователь может создавать, изменять и удалять любые файлы и запускать любые программы. Обычные пользователи ограничены в правах: хоть они

и могут запустить большинство программ, они не могут вмешиваться в работу других пользователей.

Для выполнения некоторых команд из этой книги требуются права суперпользователя. Я добавляю к таким командам слово `sudo`:

→ **sudo** далее указывается команда, требующая прав суперпользователя

ВНИМАНИЕ!

С помощью команд `sudo` можно вывести из строя вашу систему Linux!

Я подробно рассказываю о привилегиях `sudo` в разделе «Привилегии суперпользователя» главы 3, а пока что достаточно знать, что `sudo` дает вам права суперпользователя. Иногда команда может запросить ваш пароль — например, если вы захотите подсчитать строки в защищенном файле `/etc/shadow`. Чтобы сделать это, выполните следующую команду:

```
→ wc -l /etc/shadow Ошибка!
wc: /etc/shadow: Permission denied
→ sudo wc -l /etc/shadow Запуск команды от лица суперпользователя
[sudo] password: xxxxxxxx
51 /etc/shadow Сработало!
```

Файловая система

Чтобы использовать систему Linux, вы должны научиться работать с файлами и каталогами Linux (они же папки), которые и представляют собой *файловую систему*. В графической среде Рабочего стола вы видите файлы и каталоги. В интерфейсе командной строки можно найти те же файлы и каталоги, однако выполнить это сложнее, поэтому нужно помнить, в каком каталоге вы находитесь и как он связан с другими каталогами. Вы будете использовать такие команды, как `cd` (сменить каталог), для перемещения

между каталогами и `pwd` (вывод рабочего каталога), чтобы отслеживать свое положение в файловой системе.

Разберемся с терминологией. Файлы в Linux организованы в *каталоги*. Последние формируют иерархическую структуру, или *дерево* (рис. 1.3). Один каталог может включать в себя другие каталоги, называемые *подкаталогами*, которые также могут включать в себя другие файлы и подкаталоги. Самый первый, верхний каталог называется *корневым* и обозначается слешем (/)¹.

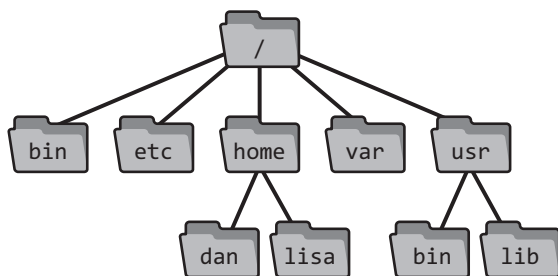


Рис. 1.3. Файловая система Linux (фрагмент). Корневой каталог располагается вверху. Например, путь к папке `dan` выглядит как `/home/dan`

Linux ссылается на файлы и каталоги согласно синтаксису «имен и слешей», называемому *путем*. Например, путь

`/один/два/три/четыре`

ссылается на корневой каталог `/`, содержащий каталог *один*, который содержит каталог *два*, который содержит каталог *три*, который содержит конечный файл или каталог *четыре*. Любой путь, начинающийся со слеша и идущий от корня, называется *абсолютным (полным)* путем.

Пути не обязательно должны быть абсолютными — они могут быть относительными и вести к какому-либо каталогу,

¹ В Linux все файлы и каталоги располагаются в соответствии с иерархией, начиная с корневого каталога. В этом состоит отличие от Windows, где доступ к различным устройствам осуществляется с помощью букв логических разделов.

отличному от корневого. На рис. 1.3 показаны два разных каталога с именем `bin`, абсолютные пути к которым — `/bin` и `/usr/bin`. Если вы будете просто ссылаться на каталог `bin`, то будет непонятно, какой именно вы имеете в виду. Требуется более подробный контекст. Любой путь, не начинающийся со слеша (как в случае с `bin`), называется *относительным*.

Чтобы правильно использовать относительные пути, нужно знать, где вы находитесь в файловой системе Linux. Ваша локация называется *текущим каталогом* (иногда активным каталогом или текущим рабочим каталогом).

У каждой оболочки есть текущий каталог, и когда вы запускаете команды в этой оболочке, они выполняются в текущем каталоге. Например, если в вашей оболочке открыт каталог `/usr`, а вы ссылаетесь на относительный путь `bin`, то полный путь будет выглядеть так: `/usr/bin`. Или если ваш текущий каталог — *один/два/три*, а относительный путь — `a/b/c`, то абсолютный путь будет `/один/два/три/a/b/c`.

Обычно *текущий* каталог оболочки обозначается `.` (точкой), а *родительский* каталог — `..` (двумя точками). Следовательно, если текущий каталог оболочки — *один/два/три*, то `.` относится к этому каталогу, а `..` относится к *один/два*.

Чтобы перейти из одного каталога в другой, используйте команду `cd`, которая меняет текущий каталог вашей оболочки:

→ `cd /usr/local/bin` *Переход в каталог /usr/local/bin*

В команде `cd` указан абсолютный путь. С помощью `cd` можно выполнять и относительные перемещения:

→ `cd d` *Переход в подкаталог текущего каталога*
 → `cd ../mydir` *Переход в родительский каталог, затем в каталог mydir*

Имена файлов и каталогов могут содержать большинство символов: прописные и строчные буквы¹, цифры, точки,

¹ Имена файлов в Linux чувствительны к регистру, поэтому следует учитывать прописные и строчные буквы.

дефис, знаки подчеркивания и т. д., за исключением символа слеша (/), который используется для разделения каталогов. Для большей эффективности избегайте пробелов, звездочек, знака доллара, круглых скобок и других символов, имеющих специальное значение для оболочки. Такие символы в именах файлов требуют специального экранирования (см. раздел «Заключение в кавычки» далее в этой главе).

Домашние каталоги

В большинстве своем персональные файлы обычных пользователей хранятся в каталоге /home, а суперпользователей — в /root. Домашний каталог чаще всего обозначается как /home/ваше_имя_пользователя (/home/smith, /home/rightman и т. д.). Существует несколько способов перейти или обратиться к домашнему каталогу:

cd

Без аргументов команда возвращает вас (то есть устанавливает текущий каталог оболочки) в домашний каталог.

HOME

Переменная HOME (см. раздел «Переменные оболочки» далее в этой главе) содержит имя вашего домашнего каталога:

→ echo \$HOME	<i>Вывод имени каталога</i>
/home/smith	
→ cd \$HOME/linuxpocketguide	<i>Переход в подкаталог</i>

~

Если вместо имени каталога указать символ «тильда», то он будет интерпретирован оболочкой как имя вашего домашнего каталога.

→ echo ~	<i>Вывод имени каталога</i>
/home/smith	
→ cd ~/linuxpocketguide	<i>Переход в подкаталог</i>

Если за тильдой указано имя пользователя (например, ~fred), то оболочка интерпретирует эту строку в виде домашнего каталога пользователя:

→ cd ~fred	<i>Переход в домашний каталог Фреда, если он существует</i>
→ pwd	<i>Команда "вывода рабочего каталога"</i>
/home/fred	

Системные каталоги

В системе Linux есть десятки тысяч системных каталогов. Они содержат файлы операционной системы, приложения, документацию и практически все, кроме персональных файлов пользователей, которые обычно находятся в каталоге /home.

Если вы не системный администратор, то редко будете сталкиваться с системными каталогами. Но если вы овладеете некоторыми знаниями, то сможете догадаться о значении большинства системных каталогов. Очень часто их имена состоят из трех частей (рис. 1.4).

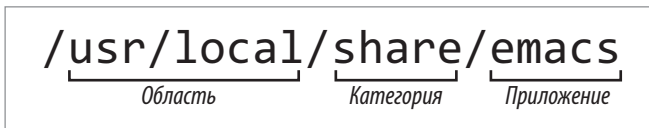


Рис. 1.4. Область, категория и приложение каталога

Путь каталога, часть 1. Область

Область пути каталога описывает назначение всего дерева каталогов. Некоторые из них являются общими:

/	(корень) Системные файлы, относящиеся к вашему дистрибутиву
/usr	(пользователь) Дополнительные системные файлы, относящиеся к вашему дистрибутиву
/usr/local	(локальный пользователь) Системные файлы, не относящиеся к вашему дистрибутиву. Они могут быть уникальными для вашей локальной сети Linux или компьютера