

# Содержание

<b>Об авторе .....</b>	<b>15</b>
Благодарности к первому изданию (2005 г.)	16
Благодарности ко второму изданию (2015 г.)	17
От издательства	18
<b>Введение .....</b>	<b>21</b>
На кого рассчитана эта книга	22
О втором издании	24
Основные соглашения	26
<b>Глава 1. Общие сведения о работе с командной строкой .....</b>	<b>29</b>
Файлы, и ничего, кроме файлов	29
Максимальная длина имени файла	30
Регистр символов в именах файлов	31
Специальные символы в именах файлов	32
Символы групповых операций	34
Специальные файлы, на которые влияет командная строка	39
Если на экране слишком много информации, сделайте перезагрузку	43
Выводы	43
<b>Глава 2. Основные команды .....</b>	<b>45</b>
Вывод списка файлов и каталогов	46
Вывод содержимого произвольного каталога	46
Использование символов групповых операций при определении содержимого каталога	47
Просмотр содержимого подкаталогов	48
Вывод содержимого каталога в один столбец	50
Вывод содержимого каталога с запятыми в качестве разделителей	50
Отображение скрытых файлов и каталогов	51
Отображение информации о типах файлов	52
Отображение информации в цвете	53
Информация о правах доступа и владельцах файлов	55
Вывод информации в обратном порядке	60

Сортировка по дате и времени	61
Сортировка содержимого каталога по размеру файлов	62
Представление размеров файлов в кило-, мега- и гигабайтах	63
Определение пути к текущему каталогу	64
Переход в другой каталог	66
Переход в рабочий каталог	66
Переход в предыдущий каталог	67
Выводы	67
<b>Глава 3. Создание и уничтожение.....</b>	<b>69</b>
Изменение сведений о времени	69
Установка произвольного времени для файла	71
Создание нового пустого файла	73
Создание нового каталога	73
Создание нового каталога и необходимых подкаталогов	74
Копирование файлов	75
Копирование файлов с использованием символов групповых операций	77
Вывод подробной информации о копировании файлов	79
Как предотвратить копирование поверх важных файлов	80
Копирование каталогов	82
Использование команды <b>cp</b> для создания резервных копий	83
Перемещение и переименование файлов	84
Переименование файлов и каталогов	87
Как Linux хранит файлы	88
Создание ссылки на другой файл или каталог	90
Удаление файлов	97
Удаление нескольких файлов с помощью символов групповых операций	99
Как предотвратить удаление важных файлов	99
Удаление пустого каталога	100
Удаление файлов и каталогов, содержащих данные	101
Удаление проблемных файлов	102
Выводы	104
<b>Глава 4. Получение информации о командах.....</b>	<b>105</b>
Получение информации о командах с помощью команды <b>man</b>	106
Получение кратких сведений о команде	109
Поиск команды по выполняемым ею действиям	110
Просмотр страницы справочной системы, посвященной конкретной команде	112

Получение информации о командах с помощью <code>info</code>	114
Навигация в системе <code>Info</code>	115
Определение путей к исполняемым, исходным файлам и страницам справочного руководства	118
Сведения об экземпляре программы для запуска	120
Выяснение интерпретации команды	121
Выводы	123
<b>Глава 5. Объединение команд .....</b>	<b>125</b>
Последовательное выполнение нескольких команд	125
Выполнение команды при условии успешного завершения предыдущих	128
Выполнение команды при условии, что предыдущая завершилась с ошибкой	130
Использование выходных данных одной команды при вызове другой	131
Входной и выходной потоки	132
Передача выходных данных одной команды на вход другой	134
Перенаправление выходных данных в файл	136
Как предотвратить перезапись файла при перенаправлении	137
Перенаправление выходных данных и запись их в конец файла	138
Использование содержимого файла в качестве входных данных	139
Сочетание перенаправления ввода и вывода	140
Одновременный вывод данных в файл и поток <code>stdout</code>	142
Выводы	143
<b>Глава 6. Просмотр содержимого файлов (как правило, текстовых) .....</b>	<b>145</b>
Распознавание типа файла	145
Вывод содержимого файла в <code>stdout</code>	148
Конкатенация файлов и вывод их в <code>stdout</code>	149
Конкатенация файлов и запись результатов в другой файл	150
Конкатенация файлов и нумерация строк	151
Постраничный вывод текста	152
Поиск с помощью программы постраничного просмотра	155
Редактирование файлов, отображаемых средствами постраничного просмотра	156
Просмотр первых десяти строк файла	157
Просмотр первых десяти строк нескольких файлов	157
Просмотр произвольного числа строк из файлов	158
Просмотр указанного количества байтов из начала файла	159
Просмотр последних десяти строк файла	162

Просмотр последних десяти строк нескольких файлов	162
Просмотр произвольного числа последних строк из файлов	164
Просмотр обновляемых строк в конце файла	164
Выводы	166
<b>Глава 7. Обработка текстовых файлов с помощью фильтров .....</b>	<b>167</b>
Подсчет количества слов, строк и символов в файле	168
Количество строк в файле	171
Выбор отдельного столбца данных в размеченном файле	172
Сортировка содержимого файла	175
Числовая сортировка содержимого файла	177
Удаление из файла строк-дубликатов	179
Замена заданных символов другими	183
Замена повторяющихся символов одним экземпляром	184
Удаление заданных символов	185
Преобразование текста в файле	189
Вывод на печать конкретных полей из файла	195
Выводы	198
<b>Глава 8. Владельцы файлов и права доступа .....</b>	<b>201</b>
Вход под другим именем	202
Вход под другим именем и с другими переменными окружения	203
Вход под именем пользователя root	204
Вход под именем пользователя root с его переменными окружения	204
Изменение групп для файлов и каталогов	206
Рекурсивное изменение принадлежности каталога группе	207
Изменение владельцев файлов и каталогов	209
Изменение владельца и группы для файлов и каталогов	210
Общие сведения о правах доступа	212
Изменения прав доступа к файлам и каталогам с использованием символьных обозначений	214
Изменения прав доступа к файлам и каталогам с использованием числовых обозначений	216
Рекурсивное изменение прав	220
Установка и сброс параметра <b>suid</b>	222
Установка и сброс признака <b>sgid</b>	225
Установка и сброс признака "sticky bit"	228
Выводы	231

<b>Глава 9. Архивирование и сжатие данных .....</b>	<b>233</b>
Архивирование и сжатие файлов посредством программы <code>zip</code>	235
Повышение уровня сжатия с помощью программы <code>zip</code>	237
Архивирование и сжатие файлов конкретного типа в каталогах и подкаталогах	239
Защита <code>zip</code> -архивов паролем	241
Разархивирование файлов	242
Проверка файлов, предназначенных для разархивирования	243
Сжатие файлов посредством программы <code>gzip</code>	244
Рекурсивная обработка файлов посредством программы <code>gzip</code>	246
Распаковка файлов, сжатых с помощью программы <code>gzip</code>	247
Проверка файлов, предназначенных для распаковки с помощью программы <code>gunzip</code>	248
Сжатие файлов посредством программы <code>bzip2</code>	249
Распаковка файлов, сжатых с помощью программы <code>bzip2</code>	250
Проверка файлов, предназначенных для разархивирования с помощью программы <code>bunzip2</code>	251
Архивирование файлов с помощью программы <code>tar</code>	251
Создание архивов и сжатие файлов посредством программ <code>tar</code> и <code>gzip</code>	253
Проверка файлов, предназначенных для распаковки и разархивирования	255
Распаковка и разархивирование файлов	257
Выводы	258
<b>Глава 10. Поиск файлов, каталогов, слов и фраз.....</b>	<b>259</b>
Поиск в базе имен файлов	260
Поиск в базе имен файлов без учета регистра	262
Обновление базы, используемой программой <code>locate</code>	263
Поиск фрагментов текстового файла	265
Общие сведения о шаблонах поиска	266
Рекурсивный поиск фрагментов текста в файлах	270
Поиск слов и выделение результатов	271
Поиск фрагментов текста в файлах без учета регистра	272
Поиск слов в файлах	273
Отображение номеров строк, в которых слово содержится в файле	274
Поиск слов в выходных данных других команд	275
Просмотр контекста для слов, имеющих в файлах	277
Отображение строк, не содержащих указанных слов	280
Отображение списка файлов, содержащих указанное слово	281
Поиск слов среди результатов	281
Выводы	283

<b>Глава 11. Команда find .....</b>	<b>285</b>
Поиск файлов по имени	285
Поиск файлов по имени владельца	287
Поиск файлов по размеру	288
Поиск файлов по типу	291
Поиск файлов по времени	292
Отображение результатов при выполнении всех выражений (AND)	295
Отображение результатов при выполнении любого из выражений (OR)	296
Отображение результатов, если выражение не выполняется (NOT)	299
Выполнение действий над каждым найденным файлом	300
Более эффективный поиск файлов	303
Поиск файлов, имена которых содержат пробелы	306
Выводы	307
<b>Глава 12. Оболочка.....</b>	<b>309</b>
Просмотр списка предыстории	309
Повторное выполнение последней команды	311
Вызов предыдущей команды путем указания ее номера	312
Вызов предыдущей команды путем указания строки символов	313
Поиск и выполнение предыдущей команды	314
Отображение псевдонимов команд	319
Просмотр псевдонима конкретной команды	320
Создание нового временного псевдонима	320
Создание нового постоянно действующего псевдонима	321
Удаление всех псевдонимов	323
Создание новой временной функции	324
Создание новой постоянной функции	326
Вывод на экран списка всех функций	329
Удаление функции	330
Когда следует использовать псевдоним, а когда функцию?	331
Выводы	333
<b>Глава 13. Контроль использования системных ресурсов.....</b>	<b>335</b>
Выяснение того, как долго работает компьютер	336
Вывод информации о процессах, выполняемых в системе	336
Просмотр дерева процессов	339
Отображение процессов, принадлежащих конкретному пользователю	340
Завершение выполняющегося процесса	341

Отображение динамически обновляемого списка выполняющихся процессов	344
Получение списка открытых файлов	347
Отображение файлов, открытых конкретным пользователем	348
Получение списка пользователей для конкретного файла	350
Отображение сведений о процессах, соответствующих конкретной программе	351
Отображение информации об оперативной памяти системы	353
Отображение информации об использовании дискового пространства	354
Определение размера области, занятой содержимым каталога	356
Ограничение вывода общим размером пространства, занятого каталогом	358
Выводы	358
<b>Глава 14. Инсталляция программного обеспечения.....</b>	<b>361</b>
Инсталляция программных пакетов в RPM-системах	362
Удаление программных пакетов из RPM-систем	363
Инсталляция зависимых программных пакетов в RPM-системах	364
Удаление зависимых программных пакетов из RPM-систем	367
Обновление программных пакетов в RPM-системах	369
Поиск пакетов, готовых к копированию на RPM-системы	370
Инсталляция программных пакетов в системе Debian	371
Удаление программных пакетов из системы Debian	373
Инсталляция зависимых пакетов в системе Debian	374
Удаление зависимых пакетов из системы Debian	378
Обновление зависимых пакетов в системе Debian	380
Поиск пакетов, доступных для копирования в систему Debian	382
Удаление ненужных инсталляционных пакетов из системы Debian	384
Устранение проблем с помощью системы APT	385
Выводы	387
<b>Глава 15. Сетевое взаимодействие .....</b>	<b>389</b>
Определение состояния сетевых интерфейсов	390
Проверка способности компьютера принимать запросы	393
Контроль прохождения пакета между двумя узлами	395
Выполнение DNS-преобразования	397
Настройка сетевого интерфейса	400
Получение информации о состоянии сетевого интерфейса беспроводной связи	403
Настройка сетевого интерфейса беспроводной связи	406
Получение адресов средствами DHCP	407
Активизация сетевого соединения	409

Перевод сетевого интерфейса в неактивизированное состояние	411
Отображение таблицы маршрутизации	412
Внесение изменений в таблицу маршрутизации	415
Устранение проблем, связанных с сетевым взаимодействием	419
Выводы	423
<b>Глава 16. Работа в сети .....</b>	<b>425</b>
Организация защищенного взаимодействия с другим компьютером	425
Защищенная регистрация на другой машине без использования пароля	429
Защищенная система FTP	433
Защищенное копирование файлов между узлами сети	435
Защищенная передача файлов и создание резервных копий	437
Копирование файлов из веб	444
Копирование веб-сайтов	450
Указание последовательностей имен копируемых файлов	452
Выводы	454
<b>Предметный указатель .....</b>	<b>455</b>



# Получение информации о командах

В предыдущей главе мы начали изучать основные команды системы Linux. Мы рассмотрели довольно много команд них, но осталось гораздо больше. Команда `ls` — мощный инструмент, поддерживающий огромное количество опций, гораздо больше, чем было рассмотрено в главе 2. Как же получить дополнительную информацию об этой команде и других, интересующих вас? И как разобраться в командах, если неизвестны даже их имена? В этом вам поможет данная глава. Вы узнаете, как глубже изучить те команды, о которых вы успели составить представление, те, о которых кроме имени вы не знаете ничего, и даже полностью неизвестные команды.

Начнем с “гигантов” — команд `man` и `info`, а затем перейдем к менее глобальным командам, которые тем не менее используют ту же информацию, что и команда `man`. Дойдя до конца этой главы, вы будете готовы продолжить изучение самых разнообразных средств, доступных посредством оболочки.

---

## ЗАМЕЧАНИЕ

Пересматривая книгу для второго издания, я удалил информацию о команде `man -u` (которая перестраивает базу данных справочной системы `man`) и команде `man -t` (которая выводит страницы `man` на печать). Кроме того, информация о команде `whatis` была перемещена в раздел о команде `man -f`, а информация о команде `apropos` — в раздел о команде `man -k`. Исходный текст можно найти на моем веб-сайте [www.granneman.com/linux-redactions](http://www.granneman.com/linux-redactions).

---

## Получение информации о командах с помощью команды `man`

### `man`

Хотите больше узнать о командах Linux? Нет ничего проще. Предположим, вам понадобилась информация о команде `ls`. Введите команду `man ls`, и на экране появится страница справочного руководства (сама команда `man` сокращенно означает *manual*, т.е. руководство), на которой подробно описываются все особенности команды `ls`. В качестве параметра `man` можно указать имя любой команды. Все они (или почти все) отражены в справочной системе.

Несмотря на то что справочное руководство чрезвычайно полезно, при работе с командой `man` подчас возникают проблемы. Необходимо знать имя команды (хотя эту трудность можно обойти), кроме того, нередко информация оказывается устаревшей и не отражает средства, появившиеся относительно недавно. Сведения о некоторых командах вообще отсутствуют. Однако хуже всего, когда описание интересующей вас команды существует и еще не устарело, но, ознакомившись с ним, вы обнаруживаете, что оно практически бесполезно.

Обычно страницы справочного руководства разрабатывают те же специалисты, которые пишут программы (бывают и исключения, но, как правило, это так). Большинство разработчиков, приложения которых были включены в дистрибутив Linux, являются первоклассными программистами, но они не всегда могут доходчиво описать работу программы. Они прекрасно разбираются в предмете, но слишком часто забывают о том, что то, что очевидно для разработчиков, зачастую неизвестно пользователям.

Несмотря на перечисленные выше проблемы, справочное руководство — незаменимый ресурс для пользователей Linux всех уровней квалификации. Если вы хотите работать с командной строкой, вам надо научиться пользоваться руководством. Как было сказано

ранее, использовать команду `man` несложно. Надо ввести `man`, а затем указать имя команды, о которой вы хотите получить дополнительные сведения.

```
$ man ls
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 -
↳cftuSUX nor --sort.
    Mandatory arguments to long options are
↳mandatory for short options too.
    -a, --all
        do not hide entries starting with .
    -A, --almost-all
        do not list implied . and ..
```

Объем данных, предоставляемых `man`, часто бывает очень большим. Для данного примера он составляет более 200 строк. Описание некоторых команд очень короткое, однако есть и такие команды, описание которых гораздо длиннее, чем приведенное в данном примере. Ваша задача — ознакомиться с описанием, которое в большинстве случаев включает в себя следующие разделы.

- **NAME.** Имя команды и краткое описание.
- **SYNOPSIS.** Формат вызова команды.
- **DESCRIPTION.** Подробное описание возможностей, предоставляемых командой.
- **OPTIONS.** Чаще всего именно этот раздел оказывается самым полезным для пользователей. В нем приводятся опции, предусмотренные для команды, и их краткое описание.
- **FILES.** Файлы, используемые командой.

- AUTHOR. Автор, который разработал программу, реализующую эту команду; контактная информация.
- BUGS. Замеченные недостатки и сведения о том, куда сообщить, если вы обнаружите новую ошибку.
- COPYRIGHT. Информация об авторских правах.
- SEE ALSO. Другие команды, имеющие отношение к рассматриваемой.

Просмотр страницы справочной системы происходит следующим образом. Для того чтобы перейти на строку вниз, следует нажать клавишу со стрелкой вниз, чтобы переместиться на одну строку вверх, надо нажать клавишу со стрелкой вверх. Перейти вперед на одну страницу позволяет клавиша пробел или <F> (первая буква в слове *forward*), а на страницу назад — клавиша <B> (первая буква в слове *backward*). Когда вы дойдете до конца страницы, выполнение команды `man` автоматически завершится и вы вернетесь в среду оболочки. В некоторых случаях автоматическое завершение не происходит, и тогда надо нажать клавишу <Q>, чтобы выйти из программы. Завершить команду можно в любой момент с помощью клавиши <Q>; при этом не обязательно находиться в конце страницы.

Нередко бывает нелегко найти требуемые сведения на странице, поэтому пользователю предоставляются минимальные средства поиска. Для того чтобы отыскать данные на открытой странице справочного руководства, надо ввести косую черту, затем термин, который надо найти, и нажать клавишу <Enter>. Если этот термин существует, содержимое страницы сдвинется так, чтобы он отображался на экране. Чтобы продолжить поиск, надо повторно нажимать клавишу <Enter> (или <N>). Чтобы вернуться к предыдущему вхождению термина, надо нажать комбинацию клавиш <Shift+N>.

---

### ЗАМЕЧАНИЕ

По умолчанию для реального вывода страниц справочника на экран команда `man` использует команду `less` (которая рассматривается в главе 6). Приведенные выше комбинации клавиш для

навигации по справочнику относятся к команде `less`; для того чтобы получить больше информации об этих комбинациях клавиш, выполните команду `man less`. Если по каким-то причинам вы не хотите использовать команду `less`, то измените переменные окружения `MANPAGER` или `PAGER`. Правда, для того чтобы понять, как они работают, придется потрудиться!

---

## Получение кратких сведений о команде

```
man -f  
whatis
```

Если вам известно имя команды, но вы не знаете, какие действия она выполняет, с ее назначением можно быстро ознакомиться, не просматривая всю страницу справочного руководства. Если вы зададите опцию `-f` (или `--whatis`) или выполните команду `whatis`, на экране отобразится раздел с описанием команды.

```
$ man -f ls  
ls (1) - list directory contents  
$ whatis ls  
ls (1) - list directory contents
```

Какой из этих вариантов предпочтительнее? Лично мне нравится использовать команду `whatis`, поскольку ее легче запомнить, а мои друзья предпочитают команду `man -f`, потому что она короче. Другим преимуществом команды `whatis` является то, что она поддерживает обработку регулярных выражений и символов групповых операций. Для поиска по базе данных с помощью групповых операций используйте опцию `-w` (или `--wildcard`).

```
$ whatis -w ls*  
ls (1) - list directory contents  
lsb (8) - Linux Standard Base support for Debian  
lshal (1) - List devices and their properties  
lshw (1) - list hardware  
lskat (6) - Lieutenant Skat card game for KDE
```

Применение символов групповых операций немного замедляет поиск по сравнению с командой `whatis` без опций, но на современных быстрых компьютерах этой разницей можно пренебречь.

Регулярные выражения можно использовать с опцией `-r` (или `--regex`).

```
$ whatis -r ^rm.*
rm (1)      - remove files or directories
rmail (8)   - handle remote mail received via uucp
rmdir (1)   - remove empty directories
rmt (8)     - remote magtape protocol module
```

---

### СОВЕТ

Из-за недостатка места регулярные выражения в этой книге не рассматриваются, но вы можете узнать больше, прочитав книгу Бена Форта (Ben Forta) *Изучаем регулярные выражения* (пер. с англ., ООО “Диалектика”, 2019 г.). Настоятельно советую как можно лучше изучить регулярные выражения — они чрезвычайно полезны. Еще лучше, если вы заглянете на сайт [http://xkcd.com/208/!](http://xkcd.com/208/)

---

## Поиск команды по выполняемым ею действиям

```
man -k
apropos
```

Немного поработав с командой `man`, вы обнаружите, что она предоставит нужную вам информацию... при условии, что вы знаете, на какой странице эта информация находится. Но что делать, если вам известно, какие действия должна выполнять команда, но не знаете ее реального имени? В этой ситуации вам поможет опция `-k` (или `--apropos`). Она позволяет организовать поиск по слову или фразе, описывающим команду. На экран выводится список команд, для которых либо имя совпадает с ключевым словом, либо это слово содержится в разделе `SYNOPSIS`.

```
$ man list
```

```
No manual entry for list
```

```
$ man -k list
```

```
last (1) - show listing of last logged
```

```
↳in users
```

```
ls (1) - list directory contents
```

```
lshw (1) - list hardware
```

```
lsof (8) - list open files
```

```
[Листинг сокращен для экономии места]
```

Опцию `-k` надо применять осторожно, так как объем выходных данных может быть очень большим, и вы не сможете найти в них те сведения, которые вам нужны. Если поиск закончился безрезультатно, попробуйте повторить его, задав другое ключевое слово.

---

### СОВЕТ

Если страница справочника является слишком длинной и вам нелегко найти то, что вы ищете, попытайтесь организовать конвейер из команд `man -k` и `grep`. (Понятие конвейера обсуждается в следующей главе, а команда `grep` — в главе 10.) Пока просто введите команду `man -k list | grep hardware`. В качестве альтернативы можете выполнить команду `apropos`, которая рассматривается в этом разделе!

---

Команда `apropos` делает то же самое, что и команда `man -k`, за одним важным исключением: подобно команде `whatis` (рассмотренной в предыдущем разделе), для поиска можно использовать опцию `-w` (или `--wildcard`) или `-r` (или `--regex`). Впрочем, стоит отметить, что существует возможность использовать опцию `-e` (или `--exact`), которая позволяет сузить поиск на конкретном слове или фразе без каких-либо исключений. Например, в предыдущем листинге поиск слова *list* привел к выдаче информации о команде `last`, потому что в ее описании есть слово *listing*. Посмотрим, что получится, если выполнить тот же самый поиск без опции `-e`.

## \$ **apropos -e list**

ls (1) - list directory contents

lshw (1) - list hardware

lsuf (8) - list open files

На этот раз команда `last` не используется, потому нам нужны только результаты, которые содержат слово *list*, а не *listing*. На самом деле список результатов поиска слова `list` содержит 80 позиций без опции `-e` и 55 позиций с использованием опции `-e`, поэтому опция `-e` делает поиск точнее.

---

### ЗАМЕЧАНИЕ

Слово *apropos* в обыденной речи встречается редко, но на самом деле это литературное слово, означающее “соответствующий” или “уместный.” Синонимом этого слова является *appropriate* (подходящий), но оно имеет другой латинский корень. Если верите, найдите точное толкование этих слов в словаре по адресу [www.dictionary.com](http://www.dictionary.com).

---

## Просмотр страницы справочной системы, посвященной конкретной команде

**man [1-8]**

В предыдущем листинге вы, вероятно, заметили, что на странице для команды `ls` отображается последовательность символов `LS(1)`. Ранее, когда мы рассматривали опцию `-k`, имя каждой команды сопровождалось числом в скобках. В большинстве случаев после имени отображалось число 1, но после команды `lsuf` было выведено число 8. Что же означают эти цифры?

Страницы справочного руководства распределены по разделам с номерами от 1 до 8. Назначение разделов описано ниже (если какой-то из примеров будет непонятен вам, не беспокойтесь; большинство команд являются узкоспециализированными).



1. Команды общего назначения, например `cd`, `chmod`, `lp`, `mkdir`, `passwd`.
2. Низкоуровневые системные вызовы, поддерживаемые ядром, например `intro`, `chmod`.
3. Функции библиотеки C, например `beep`, `HTML::Parser`, `Mail::Internet`.
4. Специальные файлы, к которым, в частности, относятся устройства в каталоге `/dev`, например `console`, `lp`, `mouse`.
5. Информация о форматах файлов и соглашениях, например `apt.conf`, `dpkg.cfg`, `hosts`, `passwd`.
6. Игры, например `atlantik`, `bouncingcow`, `kmahjongg`, `rubik`.
7. Различная информация, в том числе сведения о макропакетах, например `ascii`, `samba`, `utf-8`.
8. Команды системного администрирования, вызываемые пользователем `root`, например `mount`, `shutdown`.

До сих пор все интересовавшие нас команды располагались в разделе 1. Это неудивительно, так как мы изучаем общие вопросы работы с системой Linux. Заметьте, что некоторые ключевые слова соответствуют нескольким разделам, например, информация о `chmod` находится в разделах 1 и 2, а сведения о `passwd` можно получить в разделе 1 или 5. По умолчанию принимается наименьший из возможных номеров. Так, если вы введете в командной строке `man passwd`, то получите информацию о команде с таким именем из раздела 1. Если вас интересует файл `passwd`, полученная информация вряд ли окажется полезной для вас. Для того чтобы получить страницу для файла `passwd`, укажите после `man` требуемый номер раздела.

```
$ man passwd
```

```
PASSWD(1)
```

```
PASSWD(1)
```

```
NAME
```

```
passwd - change user password
```

```
SYNOPSIS
```

```
passwd [-f|-s] [name]
```

```
passwd [-g] [-r|-R] group
```

```

passwd [-x max] [-n min] [-w warn] [-i inact]
↳login
passwd {-l|-u|-d|-S|-e} login
DESCRIPTION
passwd changes passwords for user and group
↳accounts. A normal user...
[Листинг сокращен для экономии места]
$ man 5 passwd
PASSWD(5)                                PASSWD(5)
NAME
passwd - The password file
DESCRIPTION
passwd contains various pieces of information
↳for each user account.
[Информация сокращена для экономии места]

```

## Получение информации о командах с помощью `info`

### `info`

Команда `man` проста в использовании, а предоставляемые ею страницы справочного руководства удобны, однако их нельзя назвать дружественными по отношению к пользователю. В рамках проекта GNU был разработан альтернативный формат — `info`-страницы, для доступа к которым используется команда `info`.

Многие пользователи считают, что страницы `info` написаны лучше, чем страницы `man`, на них более полно представлена информация, однако страницы `man` проще в работе. Для конкретной команды страница `man` только одна, а содержимое страниц `info` организовано в виде разделов, называемых узлами, которые могут включать подразделы, называемые подузлами. Для того чтобы эффективно работать с системой `info`, надо освоить навигацию не только в пределах одной страницы, но также между узлами и подузлами. Поначалу бывает довольно трудно найти информацию на страницах `info`. Как ни странно, именно система, разработанная специально для начинающих, оказывается более сложной в изучении.

---

## ЗАМЕЧАНИЕ

Лично мне страница `info` очень не нравится, потому, если я вынужден ее использовать (когда в справочнике `man` нет нужной страницы), я обычно перехожу в поисковую систему Google.

---

Система `info` имеет много аспектов (что делает ее интереснее!). Чтобы получить сведения о системе `info`, надо использовать следующую команду:

```
$ info info
```

В результате ее выполнения будет открыта страница `info` для команды `info`. Попробуем научиться ориентироваться в пространстве `Info`.

## Навигация в системе `Info`

В пределах одного раздела действуют следующие правила. Чтобы перейти вниз или вперед на одну строку, надо нажать клавишу со стрелкой вниз. Аналогично клавиша со стрелкой вверх позволяет перейти на одну строку вверх или назад. Когда вы достигнете конца раздела, курсор не будет реагировать на попытки сдвинуться вниз.

Если вы хотите переместиться вниз на один экран, надо использовать клавишу `<PageDown>`, а если на один экран вверх — клавишу `<PageUp>`. Описанные здесь клавиши не позволят покинуть текущий раздел.

Если вы достигли конца раздела и хотите перейти в его начало, нажмите клавишу `<B>` (первая буква в слове `beginning`).

Если, перемещаясь по разделу, вы заметили, что информация выглядит странно, например, символы или слова искажены, нажмите комбинацию клавиш `<Ctrl+L>`, чтобы обновить содержимое экрана.

Теперь, когда вы знаете, как перемещаться в пределах раздела или узла, рассмотрим навигацию между узлами. Если вы не хотите перемещаться вперед и назад с помощью клавиш `<PageDown>` и

<PageUp>, используйте для перемещения вниз клавишу пробела, а для перемещения вверх — клавишу <Backspace> или <Delete>. Действие этих клавиш несколько отличается от действия клавиш <PageDown> и <PageUp>: если вы достигнете конца раздела, то перейдете к следующему разделу (или подразделу, если таковой существует). Перемещаясь назад и достигнув начала раздела, вы перейдете к предыдущему разделу или подразделу. Используя пробел или клавиши <Backspace> и <Delete>, вы можете просмотреть весь набор info-страниц для конкретной команды.

Чтобы перейти к следующему разделу несколько быстрее и избежать многократных нажатий клавиш, воспользуйтесь клавишей <N> (next). Если раздел, который вы просматриваете, содержит подразделы, то, нажав клавишу <N>, вы пропустите подразделы и сразу перейдете к следующему разделу того же уровня, что и текущий. Если вы просматриваете подраздел и нажмете клавишу <N>, то перейдете к следующему подразделу. Подобно клавише <N>, которая перемещает к следующему разделу того же уровня, клавиша <P> (previous) переместит вас к предыдущему разделу.

Если надо переместиться вперед к следующему элементу, независимо от того, является ли он узлом или подузлом, надо использовать клавишу <]>. Если вы просматриваете узел, в котором есть подузлы, то после нажатия клавиши <]> перейдете к первому из них, если же подузлов нет, то эта клавиша переместит вас к следующему узлу того же уровня. Чтобы переместиться назад подобным образом, надо использовать клавишу <[>.

Переместиться к вышестоящему или родительскому узлу позволяет клавиша <U> (up). При этом будьте осторожны: нажав данную клавишу лишний раз, вы подниметесь выше корневого узла для текущей команды и очутитесь в узле Directory — настоящем корневом узле, из которого можно перейти к любому другому узлу info. (Достичь узла Directory можно также, нажав клавишу <D> (directory), причем сделать это можно в любой момент.)

Узел `Directory` представляет один из типов страниц, доступных посредством системы `info`, а именно, страницу `Menu`, в которой перечисляются подтемы текущей темы. Если вы окажетесь на странице `Menu`, то сможете переместиться к одному из указанных в ней подузлов одним из двух способов. Введите `m` (`menu`), а затем имя подузла, в который вам необходимо перейти. Например, ниже показана первая страница, которую вы увидите, выполнив команду `info info`.

```
File: dir, Node: Top This is the top of the INFO tree
```

```
This (the Directory node) gives a menu of major topics.  
Typing "q" exits, "?" lists all Info commands, "d" returns here,  
"h" gives a primer for first-timers,  
⤴ "mEmacs<Return>" visits the Emacs manual, etc.
```

```
* Menu:
```

```
Basics
```

```
* Common options: (coreutils)Common options.  
* Coreutils: (coreutils). Core GNU (file, text, shell) utilities.  
* Date input formats: (coreutils)Date input formats.  
* Ed: (ed). The GNU line editor
```

Для того чтобы перейти в пакет `Coreutils`, надо ввести букву `m` и за ней слово `Core`. После этого можно либо заканчивать ввод, т.е. ввести слово `utils`, либо нажать клавишу `<Tab>`, и система `Info` автоматически сформирует имя пункта, соответствующего уже введенным символам. Если система `info` отказывается формировать имя пункта, значит, вы допустили ошибку либо введенным вам символам соответствуют несколько пунктов меню. Исправьте ошибку или продолжайте ввод до тех пор, пока `info` сможет однозначно идентифицировать интересующий вас пункт. Если в этот момент окажется, что вам не надо покидать данную страницу, нажмите комбинацию клавиш `<Ctrl+G>`, чтобы отменить команду, и продолжайте просмотр текущего узла.

Существует и альтернативный способ. Используя клавиши со стрелкой вверх или вниз, вы можете разместить курсор на требуемом пункте меню и нажать клавишу <Enter>.

Если вам надо осуществить не навигацию, а поиск, у вас также есть два способа сделать это. Вы можете искать по заголовкам узлов либо по всему тексту. Для того чтобы искать в заголовках, введите `i` (index), затем ключевое слово и нажмите клавишу <Enter>. Если слово присутствует в заголовке, вы можете перейти к соответствующему узлу. Можно также повторить поиск и перейти к следующему результату, для этого надо ввести запятую.

Если вы хотите выполнить поиск во всем тексте, введите `s` (search), затем ключевое слово или фразу и нажмите клавишу <Enter>. Для того чтобы повторить поиск, непосредственно после нажатия клавиши <Enter> нажмите клавишу <S>. Сделать это сложнее, чем ввести запятую, но такой подход позволяет добиться нужных результатов.

Если вы “заблудились” в системе info и вам нужна помощь, нажмите клавишу <?>. В ответ в нижней части окна отобразятся команды системы info. Перемещаться в этом разделе позволяют клавиши со стрелками. Чтобы отменить подсказку, нажмите комбинацию клавиш <Ctrl+X><0>, т.е. сначала нажмите комбинацию клавиш <Ctrl+X>, затем отпустите эти клавиши и нажмите клавишу <0>.

И наконец, чтобы закончить работу с системой info, нажмите клавишу <Q>, и вы вернетесь в оболочку.

## Определение путей к исполняемым, исходным файлам и страницам справочного руководства

### `whereis`

Команда `whereis` чрезвычайно полезна: она сообщает путь к исполняемому файлу программы, ее исходным файлам (если они существуют) и соответствующим страницам справочного руководства.

Например, для `kword` (текстового процессора из набора `Koffice`) можно получить следующую информацию (она будет предоставлена при условии, что двоичные, исходные и справочные файлы были установлены):

```
$ whereis kword
kword: /usr/src/koffice-1.4.1/kword /usr/bin/kword
↳/usr/bin/X11/kword usr/share/man/man1/kword.1.gz
```

Сначала команда `whereis` сообщает расположение исходных файлов, `/usr/src/koffice-1.4.1/kword`, затем информирует о местонахождении исполняемых файлов: `/usr/bin/kword` и `/usr/bin/X11/kword`. Как видите, программа `kword` обнаружена в двух позициях файловой системы. Это несколько необычно, но, тем не менее, в этом нет ничего странного. И наконец, вы получаете информацию о том, где находятся страницы справочного руководства: `/usr/share/man/man1/kword.1.gz`. Теперь вы знаете, что программа действительно инсталлирована на вашем компьютере, и можете запустить ее.

Если вас интересуют только исполняемые файлы, при вызове `whereis` укажите опцию `-b`.

```
$ whereis -b kword
kword: /usr/bin/kword /usr/bin/X11/kword
```

Опция `-m` нужна тем, кому необходимы лишь страницы справочного руководства.

```
$ whereis -m kword
kword: /usr/share/man/man1/kword.1.gz
```

И наконец, если вы хотите получать только сведения об исходных файлах, укажите при вызове `whereis` опцию `-s`.

```
$ whereis -s kword
kword: /usr/src/koffice-1.4.1/kword
```

Команда `whereis` позволяет быстро получить чрезвычайно важную информацию о программах. Со временем вы начнете ее использовать гораздо чаще, чем предполагаете сейчас.

## Сведения об экземпляре программы для запуска

### which

Вспомните команду `whereis` и результаты ее применения к команде `kword` с опцией `-b`.

```
$ whereis -b kword  
kword: /usr/bin/kword /usr/bin/X11/kword
```

В файловой системе есть два исполняемых файла `kword`. Но какой из них будет запущен при указании имени `kword` в командной строке? Выяснить это можно с помощью команды `which`.

```
$ which kword  
/usr/bin/kword
```

Команда `which` сообщает о том, какой вариант команды будет выполнен, если вы зададите ее имя. Если вы введете слово `kword`, а затем нажмете клавишу `<Enter>`, ваша оболочка выполнит тот файл, который находится в каталоге `/usr/bin`. Если вы захотите выполнить тот файл, который расположен в каталоге `/usr/bin/X11`, вам надо сделать этот каталог текущим посредством команды `cd`, а затем ввести `./kword`. Можно также указать полный путь к файлу: `/usr/bin/X11/kword`.

Команда `which` также представляет собой быстрый способ выяснить, поддерживается ли та или другая команда в вашей системе. Если команда может быть выполнена и путь к соответствующему каталогу указан в переменной окружения `PATH`, вы получите информацию о том, где найти ее. В противном случае отобразится лишь приглашение для ввода очередной команды.

```
$ which arglebargle  
$
```

Если вы хотите определить местоположение всех экземпляров файлов, поддерживающих команду (как в случае, если вы зададите выражение `whereis -b`), используйте опцию `-a` (первая буква слова “all”).



```
$ which -a kword
/usr/bin/kword
/usr/bin/X11/kword
```

## Выяснение интерпретации команды

### type

Команда `type` показывает, как оболочка `bash` будет интерпретировать команды, которые вы вводите. Рассмотрим несколько примеров.

Для начала посмотрим, что произойдет, если мы выполним команду `type`.

```
$ type ls
ls is aliased to '/bin/ls -F --color'
$ type cd
cd is a shell builtin
$ type whereis
whereis is /usr/bin/whereis
$ type mark
mark is a function
mark ()
{
    mkdir -p "$MARKPATH";
    ln -s "$(pwd)" "$MARKPATH/$1"
}
$ type do
do is a shell keyword
```

Вывод команды `type` показывает, как команда `bash` интерпретирует каждую команду. Есть пять возможных ответов.

- `alias`. Созданная вами мягкая ссылка, включая цель ссылки. К жестким ссылкам этот вариант не относится, потому что они вообще не выводятся на экран. (Детали см. в главе 3.)
- `builtin`. Команда встраивается в оболочку `bash` и выполняется ею непосредственно, в отличие от вызова внешней программы.

Примерами встроенных команд являются `alias`, `cd`, `echo`, `history`, `source` и `type`.

- `file`. Команда, которая не встроена в оболочку, и поэтому называется внешней. Ее примерами являются команды `whereis` и `whatis` из этой главы.
- `function`. Функция (см. главу 12), созданная системой Linux, программой или вами. Как и команда `alias`, команда `type` выводит реальное содержание этой функции.
- `keyword`. Зарезервированное слово, которое используется только оболочкой `bash`, например `do`, `for`, `if`, `then`, `while` и `!`.

Если вы хотите лишь выяснить, к какой из этих пяти категорий относится ваша команда, используйте опцию `-t`.

```
$ type -t ls
alias
$ type -t cd
builtin
$ type -t mark
function
```

Коротко и точно! Опция `-a` приводит к более длинному листингу, потому что в нем указываются не только разновидность команды, но и все места, где ее можно найти.

```
$ type -a ls
ls is aliased to '/bin/ls -F --color'
ls is /bin/ls
```

Итак, теперь вам известно, что команда `ls` относится к категории `alias`, и вы знаете оригинал, к которому относятся псевдонимы, и путь к исходной команде.

Команда `type` не из тех команд, которые используются каждый день, но если вам интересно, почему та или иная команда не работает так, как вы ожидали, команда `type` поможет вам понять причину.

## Выводы

Название данной главы говорит о том, что в ней будет продолжен разговор о командах. Теперь вы знаете, что существуют самые разнообразные способы получить дополнительную информацию о тех средствах, которые доступны из командной строки. Основными из них являются команды `man` и `info`. Они используют огромные объемы данных, описывающих практически все команды, доступные на компьютере под управлением Linux. Нельзя также забывать о командах `whereis`, `whatis`, `apropos` и `which`. Они очень полезны, особенно тогда, когда вам не хочется вникать в детали использования команд `man` и `info`. Но далеко не всегда они могут предоставить необходимые сведения. Время от времени приходится читать справочное руководство. Это так же, как и с овощами. Не все любят их, но они полезны.

Многие команды, описанные в данной главе, частично дублируют друг друга. Например, `man -k` — это то же самое, что `apropos`, `man -f` можно заменить `whatis`, а `whereis -b` функционально эквивалентна `which -a`. За вами выбор, какую из них использовать в каждой конкретной ситуации. Несмотря на то что разные команды выполняют одни и те же действия, желательно знать их все. Это поможет вам читать сценарии, написанные другими. Система Linux отличается тем, что предоставляет пользователю богатый выбор возможностей, даже тогда, когда это касается таких простых вещей, как команды, запускаемые посредством оболочки.