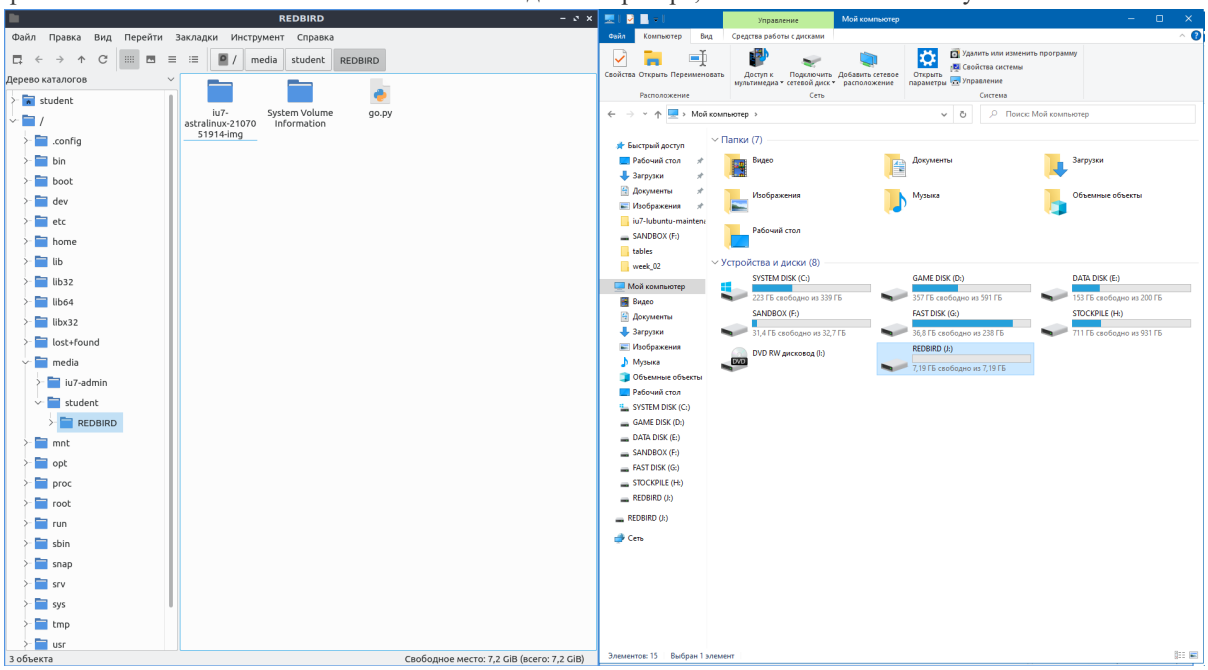
Занятие 2. Командная оболочка, работа с файлами и каталогами

Файловая система Linux

Также как и Windows, Linux организует свои файлы в иерархическую структуру каталогов. Каталоги имеют древовидную организацию и могут содержать другие файлы и каталоги. Первый каталог в файловой системе мы будем называть **корневым каталогом** или просто **корнем**.

В операционной системе Windows каталоги на разных дисках образуют несколько отдельных деревьев. В UNIX-подобных операционных системах, к которым относится Linux, каталоги на разных дисках объединяются в одно дерево. Устройства хранения подключаются (монтируются) к разным точкам дерева файловой системы в зависимости от желания администратора, ответственного за обслуживание системы.



Пример того, как выглядит монтированная флешка в Lubuntu 20.04 и Windows 10.

Linux различает регистр символов в именах файлов, т. е. `myfile.txt` и `MyFile.txt` – это два разных файла, которые могут находиться в одном каталоге.

В Linux не поддерживается понятие “расширения файла”. Расширение рассматривается как часть имени. Тип и/или назначение файла определяется другими средствами.

В именах файлов можно использовать любые символы, кроме прямого слеша (/). Хотя Linux поддерживает длинные имена файлов с пробелами и знаками пунктуации, не следует использовать в именах файлов пробелы и знаки пунктуации, кроме точки, дефиса и подчеркивания. Наличие пробелов в именах файлов осложняет решение многих задач.

Файлы, имена которых начинаются с точки, считаются скрытыми.

Файловая система Linux обычно соответствует стандарту Filesystem Hierarchy Standard, которого придерживаются UNIX-подобные системы.

Каталог	**Назначение**
/	Корневой каталог. Обычно принято не размещать в корне какие-либо файлы.
/bin	Каталог хранит большинство программ, относящихся к категории утилит GNU.
/boot	Загрузочный каталог, в котором хранятся файлы, требующиеся во время процесса загрузки операционной системы.
/dev	Каталог устройств, в котором Linux создает специальные файлы устройств.
/etc	Каталог файлов конфигурации системы.
/home	Каталог, в котором Linux создает пользовательские каталоги.
/lib	Каталог, в котором хранятся файлы библиотек системы и приложения.
/media	Каталог, который используется для монтирования сменных носителей.
/mnt	Каталог, который используется для временного монтирования файловых систем.
/opt	Дополнительный каталог, который часто служит для хранения дополнительных пакетов программ.
/root	Каталог пользователя root.
/sbin	Каталог, в котором хранятся дополнительные программы для администрирования и настройки системы.
/tmp	Каталог, в котором могут создаваться временные работе файлы.
/usr	Каталог программ, установленных пользователем.
/var	Каталог с «изменчивым содержимым», предназначенный для часто изменяющихся файлов, таких как файлы журналов.

Эмулятор терминала

Обычно в каждом дистрибутиве Linux, который используется в качестве настольной системы, используется графический интерфейс. При использовании графического интерфейса для взаимодействия с командной оболочкой нам понадобится эмулятор терминала. Такая программа обычно присутствует в меню графического рабочего стола (в KDE используется konsole, в GNOME – gnome-terminal; часто соответствующий пункт в меню называется просто «терминал» («terminal»)).

Попробуйте вызвать терминал в своей системе. После появления окна терминала вы увидите

```
user@linux-pc:~$
```

Мы будем называть это **приглашением к вводу** (**shell prompt**). В разных дистрибутивах оно выглядит по-разному, но обычно включает в себя строку `имя_пользователя@имя_компьютера`, за которой следует имя текущего каталога и знак доллара. Если Вы находитесь в своём домашнем каталоге, то вместо его полного пути будет выводиться знак тильда (~).

Замечание 1
Если последний символ в приглашении — знак решетки (#), это означает что сеанс в терминале обладает правами суперпользователя (администратора).

В большинстве дистрибутивов Linux по умолчанию установлена командная оболочка bash. Эта командная оболочка, соответствующая стандарту POSIX, будет использоваться в этом курсе.

Замечание 2
В современных системах Linux после запуска автоматически создаются несколько виртуальных консолей. Получить доступ к этим виртуальным консолям в большинстве дистрибутивов Linux можно с помощью комбинаций клавиш CTRL+ALT+FN, где `N (1 <= N <= 12)` – номер необходимой виртуальной консоли. Некоторые из этих консолей являются текстовыми.

Наберите команду `date` (будучи вызванной без аргументов, она выводит дату) и нажмите клавишу ENTER.

```
user@linux-pc:~$ date
Sun Jul 18 20:53:53 MSK 2021
```

Если теперь нажать клавишу со стрелкой вверх, после приглашения к вводу появится предыдущая команда `date`. Это называется *историей команд*. Нажмите клавишу со стрелкой вниз, и предыдущая команда исчезнет.

Вызовите предыдущую команду еще раз. Теперь попробуйте нажимать клавиши со стрелками влево и вправо. Позиция курсора меняется в командной строке. Благодаря этому легко можно редактировать команды.

Завершить сеанс работы с терминалом можно либо закрыв окно эмулятора терминала, либо указав команду `exit`.

Операции с файлами и каталогами

Переход по каталогам

Для перехода в другой каталог используется команда `cd` (*change directory*). Формат команды представлен ниже

```
cd destination
```

Команда может принимать единственный параметр destination – имя каталога, в который необходимо перейти. Если параметр отсутствует, то выполняется переход в домашний каталог пользователя.

Параметр destination может указываться двумя способами:

- в виде абсолютного пути;
- в виде относительного пути.

Абсолютный путь к каталогу (файлу) определяет, где находится каталог (файл), начиная от корневого каталога.

```
user@linux-pc:~$ cd /etc
user@linux-pc:/etc$
```

Приглашение к вводу показывает, что новым каталогом для командной оболочки после выполнения команды cd стал каталог /etc.

Относительный путь к каталогу (файлу) позволяет указывать путь к нужному каталогу (файлу) относительно текущего местоположения, не вынуждая начинать с корня. Относительный путь не начинается с символа «/», который указывает на корневой каталог.

Относительный путь начинается либо с имени каталога, если выполняется переход к каталогу, расположенному ниже текущего каталога, либо со специального символа, указывающего положение относительно текущего каталога. Для этого используются
точка «.», которая обозначает текущий каталог;
двойная точка «..», которая представляет родительский каталог.

Например, требуется перейти в каталог Desktop

```
/home
  /user # You are here
    /Видео
    /Документы
    /Загрузки
    /Изображения
    /Музыка
    /Общедоступные
    /Шаблоны
    /Desktop
```

Сделать это можно следующим образом

```
cd Desktop
```

или

```
cd /home/user/Desktop
```

или

```
cd ../user/Desktop
```

Обратите внимание, что папки на самом деле носят кириллические имена, а не являются ссылками на папки Downloads, Videos, Templates, как в Windows.

Символ “двоеточие” обеспечивает переход вверх на один уровень, а затем часть относительного пути `/user/Desktop` позволяет снова перейти на более низкий уровень, но уже в каталог `Desktop`. Символ двойная точка можно использовать сколько потребуется.

```
user@linux-pc:~$ cd ../../etc/
user@linux-pc:/etc$
```

Название текущего каталога выводится командой `pwd` (*print working directory*).

```
user@linux-pc:~$ pwd
/home/user
user@linux-pc:~$
```

Перечисление содержимого каталога

Чтобы вывести список каталогов и файлов в текущем рабочем каталоге воспользуйтесь командой `ls` (*list*).

```
user@linux-pc:~$ ls
repos  work
user@linux-pc:~$

```

Команде можно указать каталог, содержимое которого требуется вывести (или даже несколько):

```
user@linux-pc:~$ ls /usr
bin    games   lib   lib64   libx32 sbin  src
config include lib32 libexec local  share app checkdirs cleanall
user@linux-pc:~$
```

Команда ls формирует вывод в алфавитном порядке (по столбцам).

Если для работы используется эмулятор терминала, который поддерживает цвет, то с помощью команды ls можно сформировать вывод, в котором разные типы файлов обозначены разным цветом. Скорее всего, у Вас даже обычный вызов ls будет цветным – это связано с алиасами, о которых мы поговорим в другой раз.

Если же эмулятор терминала не поддерживает цвет, то команде ls можно задать параметр -F, в котором проще отличить файлы от каталогов

```
user@linux-pc:~$ ls /usr -F
bin/   games/   lib/   lib64/   libx32/ sbin/  src/
config include/ lib32/ libexec/ local/  share/ app* checkdirs* cleanall*
user@linux-pc:~$
```

В выводе, сформированном с помощью параметра -F, каталоги обозначены символом «/». Кроме того, исполняемые файлы обозначаются символом «*».

Примечание: параметры и аргументы

Команды часто сопровождаются одним или несколькими параметрами, изменяющими их поведение, и одним или несколькими аргументами, на которые воздействует команда. Большинство команд выглядит вот так

```
команда -параметр аргумент
```

Большинство команд используют параметры, состоящие из одного символа, которому предшествует дефис, например, -F. Многие команды поддерживают параметры с длинными именами, состоящими из слова, которому предшествуют два дефиса. Кроме того, многие команды позволяют объединять вместе параметры с короткими именами.

```
user@linux-pc:~$ ls -Fa --reverse
work/    .sudo_as_admin_successful .local/     .bashrc       ./
repos/   .subversion/              .landscape/ .bash_logout
.viminfo .profile                  .config/    .bash_history
.vim/    .motd_shown               .cache/     ../
```

В примере команде ls передаются два коротких параметра: параметр F, добавляющий в конец каждого имени символ-индикатор, параметр a, выводящий информацию о скрытых файлах, и параметр с длинным именем reverse, чтобы изменить порядок сортировки на обратный.

В Linux часто используется *скрытые файлы* для хранения информации о конфигурации. Скрытые файла в Linux — это файлы, имена которых начинаются с точки. Такие файлы не появляются в выводе команды ls по умолчанию (именно поэтому они называются скрытыми).

Для отображения скрытых файлов используется параметр a (all) (а в графическом файловом менеджере обычно сочетание клавиш Ctrl+H).

```
user@linux-pc:~$ ls

user@linux-pc:~$ ls -a
./  ../  .bash_history  .bashrc  .config/  .dbus-keyrings/  .gitconfig  .inputrc
.lesshst  .local/  .minttyrc  .profile

```

Для получения дополнительных сведений о каталогах и файлах используется параметр -l

```
user@linux-pc:~$ ls -al
итого 31K
drwxr-xr-x 1 user user    0 сен  8 22:13 ./
```

```
drwxr-xr-x 1 user user    0 ноя 15  2020 ../
-rw-r--r-- 1 user user 6,8K сен  8 21:43 .bash_history
-rw-r--r-- 1 user user 5,7K май  6 08:38 .bashrc
drwxr-xr-x 1 user user    0 июл 29 21:51 .config/
drwxr-xr-x 1 user user    0 авг 29 13:50 .dbus-keyrings/
-rw-r--r-- 1 user user  113 июн  7 17:39 .gitconfig
-rw-r--r-- 1 user user 3,2K дек 15  2018 .inputrc
-rw-r--r-- 1 user user   20 авг 23 11:58 .lesshst
drwxr-xr-x 1 user user    0 июл 29 21:51 .local/
-rw-r--r-- 1 user user  149 июл  6 17:57 .minttyrc
-rw-r--r-- 1 user user 1,6K дек 15  2018 .profile
```

В первой строке вывода содержится информация об общем количестве блоков данных, относящихся к текущему каталогу. Остальные строки содержат следующую информацию:

- тип файла (каталог (d), файл (-), символьное устройство (c), блочное устройство (b));
- разрешения для данного файла
- количество жестких ссылок на файл;
- имя владельца файла;
- имя группы файлов, к которой принадлежит этот файл;
- размер файла в байтах;
- время последнего изменения файла;
- имя каталога или файла.

Команда ls предоставляет возможность определить *маску*, которая используется для определения какие каталоги и файлы должны быть отображены в выводе. Маска - это шаблон поиска, в котором могут использоваться так называемые групповые символы.

Групповые символы

Командная оболочка поддерживает специальные символы, помогающие быстро определять группы имен файлов. Эти специальные символы называются *групповыми символами*. Эти символы позволяют выбирать имена файлов по шаблону. Если Вы знакомы с масками, то будьте аккуратны - это хоть и похожие понятия, но разные.

Групповой символ	**Соответствует**
*	Любая последовательность любых символов
?	Любой один символ
[символы]	Любой один символ из указанного множества символов
[!символы]	Любой один символ, не принадлежащий указанному множеству символов
[[:alnum:]]	Любой алфавитно-цифровой символ
[[:alpha:]]	Любой алфавитный символ
[[:digit:]]	Любой цифровой символ
[[:lower:]]	Любая буква в нижнем регистре

Групповой символ	Соответствует
[[:upper:]]	Любая буква в верхнем регистре

Примеры использования групповых символов

Шаблон	Соответствует
*	Все имена файлов
g*	Все имена файлов, которые начинаются на букву g
b*.txt	Все имена файлов, которые начинаются на букву b, за которой следует любое число символов, и заканчиваются на «.txt»
Data???	Все имена файлов, которые начинаются на строку Data, за которой следуют любые три символа
[abc]*	Все имена файлов, которые начинаются на букву a, b или c
[[:upper:]]*	Все имена файлов, начинающиеся на букву в верхнем регистре
[![:digit:]]*	Все имена файлов, которые не начинаются с цифры
***[[:lower:]123]**	Все имена файлов, которые заканчиваются буквой в нижнем регистре или цифрой 1, 2 или 3

Групповые символы можно использовать с любыми командами, принимающими имена файлов или каталогов в виде аргументов.

Определение типов файлов

Полезно иметь возможность определять тип содержимого файлов. Для этого можно использовать команду **file**. При вызове команды в качестве параметра необходимо указать имя файла.

```
user@linux-pc:~$ file .bash_history
.bash_history: UTF-8 Unicode text

user@linux-pc:~$ file .config
.config: directory
```

Просмотр содержимого файлов

Команда less – это программа для просмотре текстовых файлов. Например, можно посмотреть историю команд

```
user@linux-pc:~$ less .bash_history
ls -al
sh go.sh
cat packages.list
cat packages.list
grep -v '^#' packages.list
.bash_history
:
```

После запуска программа less выведет содержимое файла. Если файл занимает больше одной страницы, его можно прокручивать вверх и вниз. Чтобы выйти из программы less, нажмите клавишу q.

Существует множество ситуативных аналогов команды less: cat, more, tail, head.

Создание каталогов

Команда `mkdir` (make directory) используется для создания каталогов.

```
mkdir каталог ...
```

Многоточие в конце аргумента означает, что аргументов может быть несколько.

Создание файлов

Иногда требуется создать пустой файл. Для этого можно воспользоваться командой touch.

```
touch имя_файла
```

Обычно команда `touch` используется для обновления времени последнего изменения файла. Но если ей передать имя несуществующего файла, она создаст пустой файл.

Копирование файлов и каталогов

Команда `cp` (copy) копирует файлы и каталоги. Ее можно использовать двумя разными способами:

```
cp имя_1 имя_2
```

чтобы скопировать один файл или каталог *имя_1* в файл или каталог *имя_2*, и

```
cp имя ... каталог
```

чтобы скопировать несколько файлов или каталогов в указанный каталог.

Параметры команды cp

Параметр	Значение	
-a,	--archive	Скопировать файлы и каталоги со всеми атрибутами.
-i,	--interactive	Запрашивать у пользователя подтверждение перед перезаписью существующего файла. Если этот параметр отсутствует, команда просто перезапишет существующие файлы.
-r,	--recursive	Рекурсивно копировать каталоги и их содержимое. Это обязательный параметр при копировании каталогов.
-u,	--update	При копировании файлов из одного каталога в другой копировать только файлы, отсутствующие в каталоге назначения или более новые.
-v,	--verbose	Выводить информационные сообщения в процессе копирования.

Перемещение и переименование файлов

Команда mv (move) выполняет перемещение или переименование файлов и каталогов в зависимости от особенностей использования. В любом случае исходный файл (или каталог) исчезает.

```
mv имя_1 имя_2
```

Перемещает или переименовывает файл или каталог *имя_1* в *имя_2*.

```
mv имя... каталог
```

Перемещает один или несколько файлов или каталогов в указанный каталог.

Команда mv поддерживает множество тех же параметров, что и команда cp.

Удаление файлов и каталогов

Команда rm (remove) используется для удаления файлов и каталогов.

```
rm имя...
```

Параметры команды rm

Параметр	Значение
-i, –interactive	Запрашивать у пользователя подтверждение перед удалением существующего файла. Если этот параметр отсутствует, команда просто удалит существующие файлы.
-r, –recursive	Рекурсивно удалить каталоги и их содержимое. Это обязательный параметр при удалении каталогов.
-f, –force	Игнорировать отсутствующие фалы и не запрашивать подтверждения.
-v, –verbose	Выводить информационные сообщения в процессе удаления.

ВНИМАНИЕ
Unix-подобные системы, такие как Linux, не имеют команды, отменяющей rm-удаление. Если вы что-то удалите командой rm, это исчезнет навсегда, а не будет перемещено в корзину.

```
rm /home/user -rf # полное удаление папки пользователя со всеми подкаталогами
rm /home/user/testdir/file1.txt # удаление одного файла, если файл не существует, будет выведено сообщение
rm /home/user/testdir/file1.txt # удаление одного файла, если файла не существует, сообщение не будет выведено
```

Создание ссылок

Команда `ln` используется для создания жесткой или символической ссылки. Ее можно использовать одним из двух способов.

```
ln файл ссылка
```

Создает жесткую ссылку.

```
ln -s имя ссылка
```

Создает символическую ссылку. В качестве *имени* можно указать как имя файла, так и имя каталога.

Жесткие ссылки

При создании жесткой ссылки создается дополнительная запись в каталоге для файла, на который указывает ссылка. Жесткая ссылка неотличима от самого файла. При удалении жесткой ссылки удаляется только сама ссылка, файл остается на месте, пока не будут удалены все жесткие ссылки на файл.

Жесткие ссылки имеют два ограничения

- жесткая ссылка не может указывать на каталог;
- жесткая ссылка не может указывать на файл за пределами собственной файловой системы.

Символические ссылки

Символическая ссылка — это файл особого типа, содержащий текстовый указатель на файл или каталог.

Получение справки

Команды

Команда может быть

- выполняемой программой;
- встроенной командой, реализованной внутри командной оболочки;
- функцией командной оболочки;
- псевдонимом (команда, которую можно создать самому на основе других команд).

Чтобы узнать к какому типу относится команда можно воспользоваться командой type.

```
user@linux-pc:~$ type ls
ls is aliased to \`ls --color=auto\`
user@linux-pc:~$ type cd
cd is a shell builtin
user@linux-pc:~$ type less
less is /usr/bin/less
```

Иногда требуется узнать точное местоположение выполняемой программы. Это можно сделать с помощью команды which.

```
user@linux-pc:~$ which ls
/usr/bin/ls
```

which не способна различать встроенные команды или псевдонимы.

```
user@linux-pc:~$ which cd
user@linux-pc:~$
```

help

bash имеет встроенную справку для каждой встроенной команды. Чтобы получить ее, введите help и имя команды.

```
user@linux-pc:~/repos/pkcs11$ help cd
cd: cd [-L|[-P [-e]] [-@]] [dir]
Change the shell working directory.

Change the current directory to DIR. The default DIR is the value of the HOME shell
variable.

The variable CDPATH defines the search path for the directory containing DIR.
Alternative directory names ...
```

Замечание

Квадратные скобки в описании команды указывают на необязательный параметр. Вертикальная черта используется для указания взаимоисключающих вариантов.

Замечание

Запуск команду help указать без параметров, она выведет список всех встроенных команд.

–help

Многие программы поддерживают параметр `--help` для вывода описания параметров, поддерживаемых команд.

```
user@linux-pc:~/repos/pkcs11$ ls --help
usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
...
```

man

Для большинства программ существует документация, которая называется man-страница (man – от слова manual). Для просмотра этой документации используется специальная программа man.

```
user@linux-pc:~/repos/pkcs11$ man ls
```

В большинстве систем Linux man использует `less` для вывода страницы

man-страницы могут отличаться друг от друга оформлением, но в общем случае содержат

- заголовок (NAME);
- краткий обзор синтаксиса команды (SYNOPSIS);
- описание назначения команды, список всех параметров (DESCRIPTION).

Обратите внимание в первой строке man-страницы пишется имя программы и в скобках число. Это число обозначает раздел документации, к которому относится данная страница.

Раздел	Описание
1	Пользовательские команды
2	Программные интерфейсы системных вызовов в ядре
3	Программные интерфейсы в библиотеке C
4	Специальные файлы, такие как узлы устройств и драйверы
5	Формат файлов
6	Игры и развлечения
7	Прочее
8	Команды системного администрирования

Иногда нужно заглянуть в конкретный раздел (например, из-за того названия команд часто совпадают с именами файлов, описание формата которого вы ищите). Сделать это можно следующим образом

```
man раздел термин
```

Программа whatis выводит имя и однострочное описание из man-страницы.

```
user@linux-pc:~$ whatis ls
ls (1)        - list directory contents
```

info

В проекте GNU существует альтернативное руководство Info, которое часто называют info-страницами. Info-страницы отображаются с помощью команды info.

Info-страницы организованы в древовидную структуру. Каждая страница содержит отдельную тему. При этом страницы включают гиперссылки, с помощью которых можно перемещаться от узлу к узлу. Описание гиперссылки начинается с символа звездочка. Чтобы перейти по гиперссылке необходимо установить текстовый курсор на звездочку и нажать клавишу ENTER.

Чтобы вывести info-страницу, введите команду info и укажите необязательное имя интересующей программы.

Команда	Действие
?	Вывести справку
Page Up или Backspace	Вывести предыдущую страницу
Page Down или Пробел	Вывести следующую страницу
n	Вперед — вывести следующий узел
p	Назад — вывести предыдущий узел
u	Вверх — вывести узел, родительский по отношению к текущему
ENTER	Перейти по гиперссылке
q	Завершить

apropos

Иногда бывает непонятно какую именно из команд нужно использовать для решения задачи. В такой ситуации можно воспользоваться командой apropos, которая осуществляет поиск заданной строки в заголовках страниц руководств.

```
user@linux-pc:~/repos/pkcs11$ apropos usb
lsusb (8)      - list USB devices
usb-devices (1)   - print USB device details
usbhid-dump (8)   - dump USB HID device report descriptors and streams
```

Замечание

Команда man -k действует как apropos.

Литература

1. Блум Р., Бреснахэн К. “Командная строка Linux и сценарии оболочки.”, глава 3.
2. Шотсс У. “Командная строка Linux.”, главы 1 — 5.
3. Filesystem Hierarchy Standard, version 3.0.