

## Лицензия

*Данный документ написан Фленовым Михаилом и распространяется только на компакт диске к книге «Linux глазами хакера» или на сайтах автора. Вы можете копировать этот документ куда угодно, но **нельзя** выкладывать в Интернете или других носителях информации без согласия автора.*

*С автором можно связаться через персональный сайт <http://www.flenov.info>*

## Команды ОС Linux

## Содержание

Лицензия .....	1
Содержание .....	2
Введение.....	3
Команды общего назначения.....	7
shutdown.....	7
startx .....	7
echo .....	7
\$HOME или ~ .....	8
\$LOGNAME .....	9
\$MAIL.....	9
\$PWD .....	9
\$OLDPWD .....	9
\$SHELL.....	10
\$USER.....	10
Переменные пользователя .....	10
Файловая система .....	12
PWD .....	12
CD .....	12
LS .....	13
cat .....	15
tac .....	15
cp.....	15
mkdir .....	16
rm .....	16
df .....	16
mount.....	17
umount.....	18
fdformat .....	18
tar .....	18
rpm .....	19
which .....	20
Работа с журналом .....	21
who .....	21
users.....	22
last .....	22
lastlog .....	23
lsof.....	25
Задачи .....	26
Команды общего назначения.....	28
Псевдонимы .....	28
Перенаправление ввода вывода.....	28
Объединение ввода .....	28
Последовательность команд.....	29
Советы по работе с командной строкой.....	31

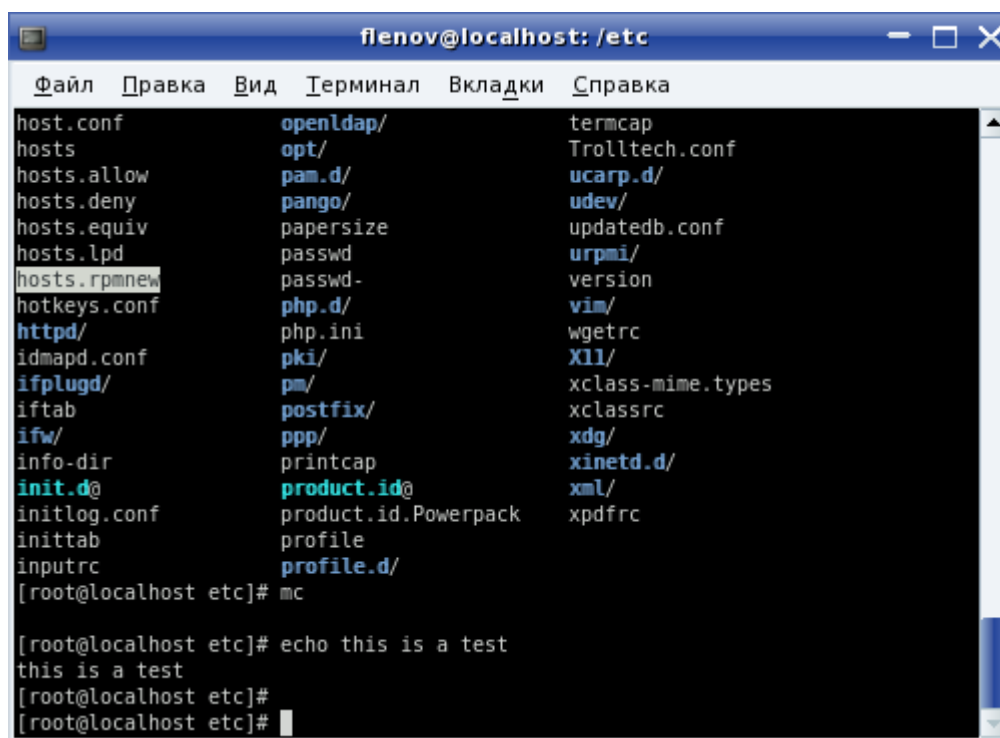
## Введение

Несмотря на то, что на каждом шагу нас преследует графический интерфейс и красивые окна, командная строка не теряет своей актуальности. Даже в ОС Windows командная строка иногда позволяет сделать очень интересные действия. А в Windows 2008 появилась даже отдельная конфигурация, которая позволяет запустить сервер без графического режима и выполнять все действия с помощью текстовых команд.

ОС Linux наоборот, последние годы все больше движется в сторону настольных систем, где пользователи любят красивые окошки и сверкающие бирюлички. Но если вы используете ОС как сервер, то без командной строки никуда не деться. Они реально помогают, даже локально, не говоря уже об удаленном подключении к серверу.

Выполнение текстовой команды происходит намного быстрее, чем движение мышкой, запуск графической программы и последующее управление графическими элементами управления. С другой стороны, команд и их параметров просто очень много. Помнить их все очень сложно, если не сказать невозможно. Я думаю, что никто и не знает абсолютно всех команд. Это просто нереально. Все знают только те команды, которыми пользуются чаще всего. А если нужно сделать редкую задачу или что-то оригинальное, то тут на помощь приходят книги и Интернет.

Я надеюсь, что данный документ станет тем справочником, с помощью которого вы сможете освоить те команды, которые помогут вам в повседневной жизни. А если понадобится выполнить что-то оригинальное, то и в этом случае, данный мануал поможет найти нужно решение.

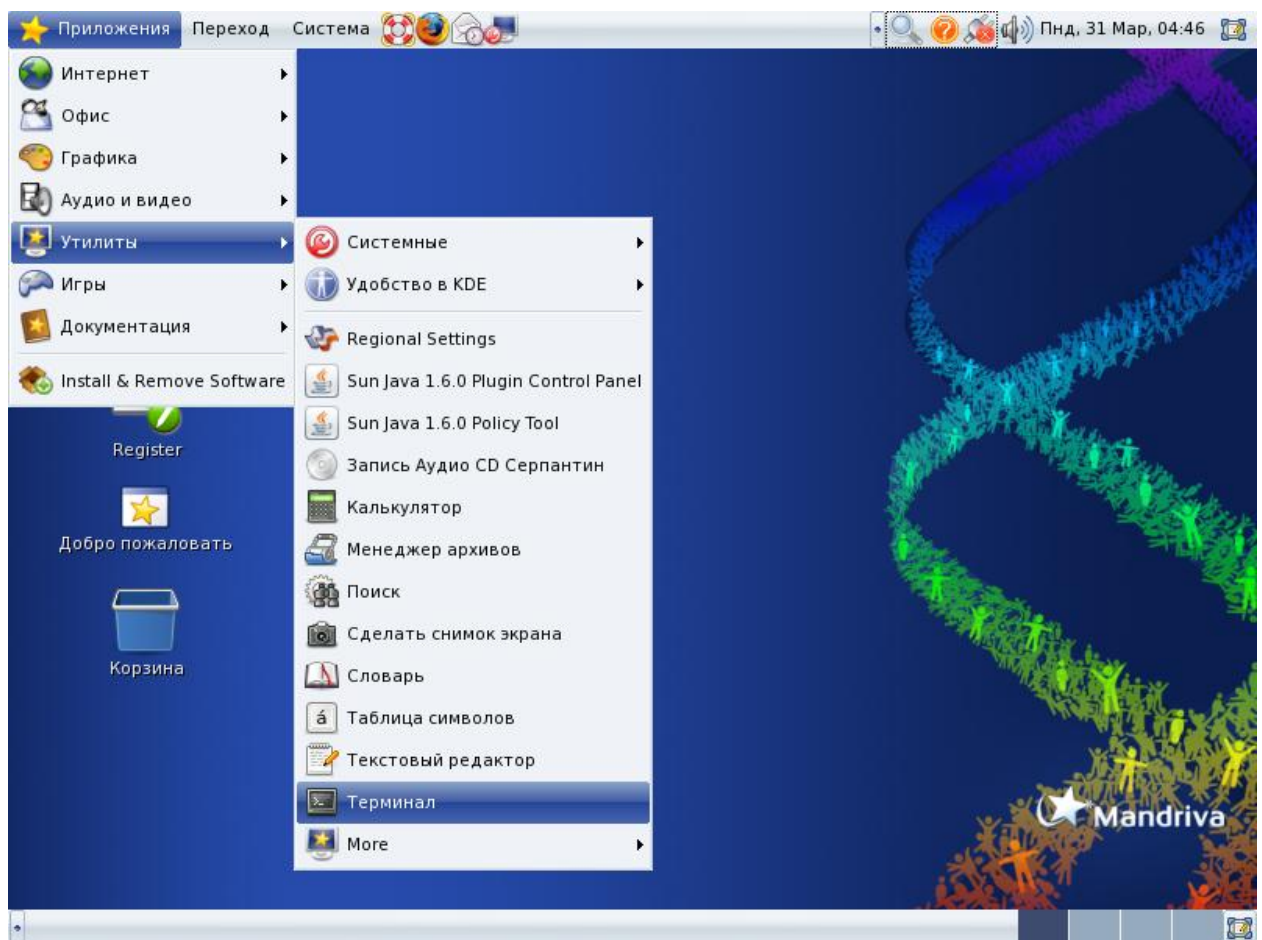


```
flenov@localhost: /etc
Файл  Правка  Вид  Терминал  Вкладки  Справка
host.conf      openldap/      termcap
hosts          opt/           Trolltech.conf
hosts.allow    pam.d/         ucarp.d/
hosts.deny     pango/         udev/
hosts.equiv    papersize      updatedb.conf
hosts.lpd      passwd         urpmi/
hosts.rpmnew   passwd-        version
hotkeys.conf   php.d/         vim/
httpd/         php.ini        wgetrc
idmapd.conf    pki/           X11/
ifplugd/       pm/            xclass-mime.types
iftab         postfix/       xclasssrc
ifw/          ppp/           xdg/
info-dir      printcap       xinetd.d/
init.d@        product.id@    xml/
initlog.conf   product.id.Powerpack
inittab        profile        xpdfrc
inputrc        profile.d/
[root@localhost etc]# mc

[root@localhost etc]# echo this is a test
this is a test
[root@localhost etc]#
[root@localhost etc]#
```

Я не претендую на полноту изложения, потому что не знаю всего и не могу знать, но я постарался максимально полно собрать для вас информацию о командах и их параметрах. Если есть какие-то вопросы или пожелания, я всегда открыт для общения и буду рад увидеть тебя на своем сайте [www.flenov.info](http://www.flenov.info).

Запуск терминала зависит от графической оболочки и от ее версии. Например, в GNOME нужно выбрать меню **Приложения/Утилиты/Терминал**, как показано на следующем экране:



*Рабочий стол GNOME и запуск консоли в GNOME*

Чтобы выполнить команду в окне терминала, достаточно ввести ее и нажать Enter. Чтобы запустить программу, нужно ввести путь к ней и потом имя самой команды. Например:

```
> /usr/local/program/filename.sh
```

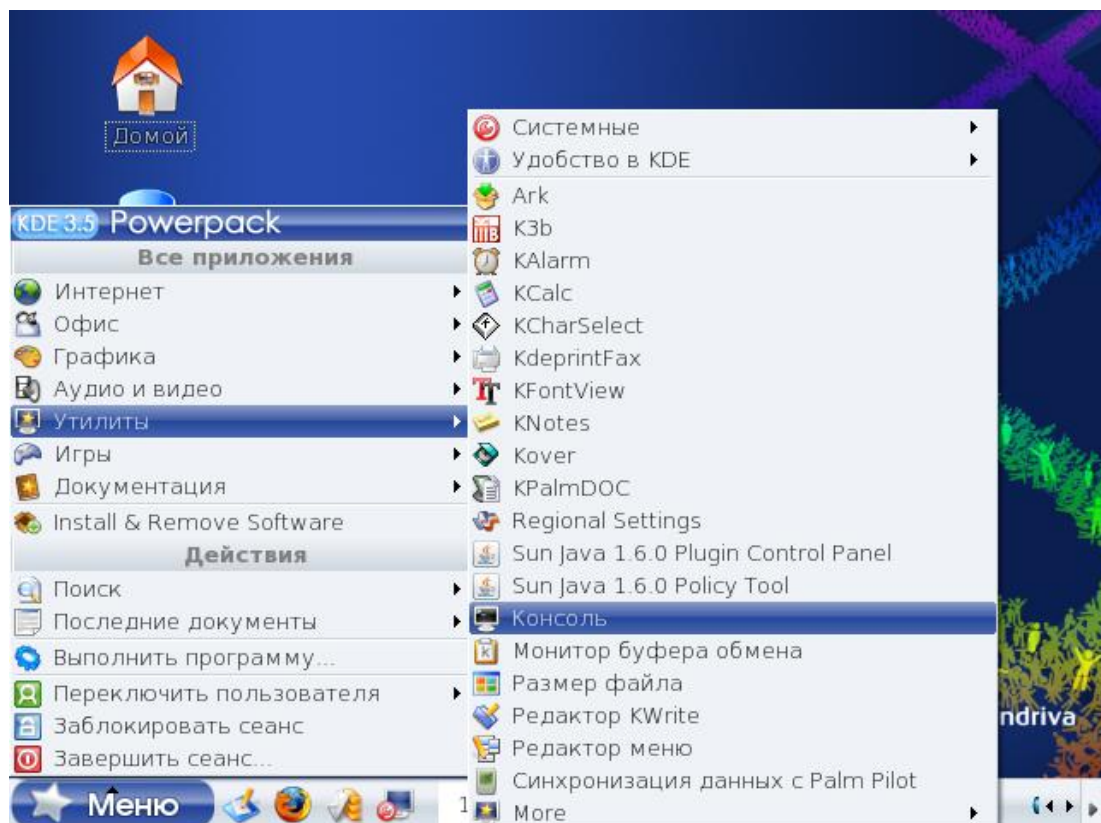
Эта команда запустит на выполнение программу filename.sh из каталога /usr/local/program. Если программа запускается из текущей директории, то достаточно ввести команду так:

```
> ./filename.sh
```

Точка указывает на текущий каталог, а значит, система ищет файл filename.sh в текущей папке.



*Загрузка Mandriva*



*Просто используйте KDE (запуск консоли в KDE)*

В данном мануале, когда нужно показать команду, вводимую в окно терминала, вначале я буду указывать символ `>`. Этот символ означает приглашение для ввода команд. Сам этот символ вводить в окно терминала не нужно. Вы должны вводить все, что находится после него. Например:

```
> ls
```

Из всех этих символов вы должны ввести только два и это `ls`. Символ приглашения вводить не нужно, и пробел тоже указан только для того, чтобы отделить символ приглашения от команды. Надеюсь, что это поможет вам лучше понимать, где находятся команды, вводимые в окно терминала.

### ***Примечание***

*Если нужно ввести в командную строку имя файла, содержащие пробелы, то это имя нужно заключить в двойные кавычки. Иначе, система не сможет понять, какое из слов, разделенных пробелами, является файлом и является ли пробел разделителем команд или это одно имя файла.*

## Команды общего назначения

### **shutdown**

Эта команда используется для завершения работы ОС. В общем виде она выглядит следующим образом:

shutdown опции время сообщение

Рассмотрим опции, которые вы можете указывать:

- -k - реально не перегружать систему, а только направить всем активным приложениям сообщение о завершении;
- -r - перезагрузить систему;
- -h – выключить компьютер вообще.
- -f – пропустить проверку файловой системы при перезагрузке.
- -F – запустить проверку файловой системы при перезагрузке.
- -c – отменить уже запущенную команду завершения работы.

Следующая команда заставит систему перезагрузиться через 10 минут, при этом, все пользователи получают сообщение *шухер, чичас перезагрузимся!*

```
shutdown -r +10 "шухер, чичас перезагрузимся"
```

Можно указать и реальное время. Например, следующая команда выключит компьютер в 6 часов вечера с тем же сообщением:

```
shutdown -h 18:00 "шухер, чичас перезагрузимся"
```

Если нужно выполнить операцию немедленно, то вместо времени укажите now. Следующая команда выключит компьютер немедленно:

```
shutdown -h now
```

### **startx**

Если вы загрузились в текстовом режиме, то переключиться графическую оболочку можно, выполнив команду startx. Никаких параметров указывать не нужно. Если конфигурационные файлы графической оболочки настроены корректно, то она загрузится.

### **echo**

Самая простая команда – echo. Она выглядит следующим образом:

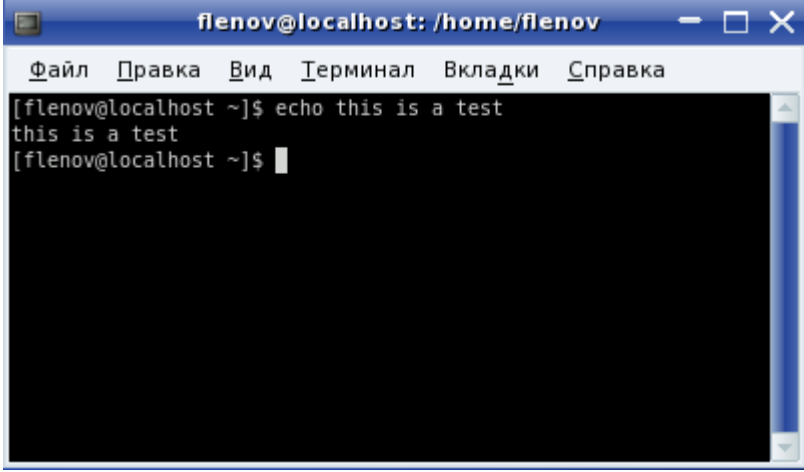
```
echo текст
```

Эта команда банально выводит на экран текст, указанный в качестве параметра. Например, выполните следующую команду:

```
> echo This is a test
```

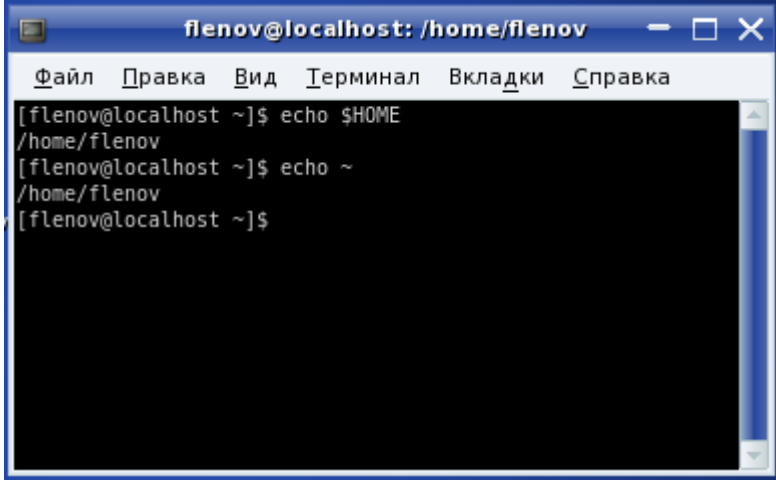
В результате на экране будет просто отображен текст:

```
This is a test
```

A screenshot of a terminal window titled 'flenov@localhost: /home/flenov'. The window has a menu bar with 'Файл', 'Правка', 'Вид', 'Терминал', 'Вкладки', and 'Справка'. The terminal shows the command '[flenov@localhost ~]\$ echo this is a test' followed by the output 'this is a test' on the next line. The prompt '[flenov@localhost ~]\$' is visible on the third line.

## ***\$HOME или ~***

Это не команды, это что-то типа переменных, и обе они указывают на домашний каталог текущего пользователя. Например, мой домашний каталог это /home/flenov. Если выполнить команду echo и указать в качестве параметра любую из этих переменных, то в результате я увижу свой каталог, как показано на следующем скрине:

A screenshot of a terminal window titled 'flenov@localhost: /home/flenov'. The window has a menu bar with 'Файл', 'Правка', 'Вид', 'Терминал', 'Вкладки', and 'Справка'. The terminal shows three lines of command and output: '[flenov@localhost ~]\$ echo \$HOME' followed by '/home/flenov' on the next line, '[flenov@localhost ~]\$ echo ~' followed by '/home/flenov' on the next line, and '[flenov@localhost ~]\$' on the third line.

Тут очень важным является регистр. \$HOME нужно писать именно в верхнем регистре.



## **\$LOGNAME**

Эта переменная содержит имя пользователя, под которым вы вошли в систему. Попробуйте выполнить следующую команду и убедитесь, что будет отображено то же имя, которое вы вводили при авторизации:

```
> echo $LOGNAME
```

## **\$MAIL**

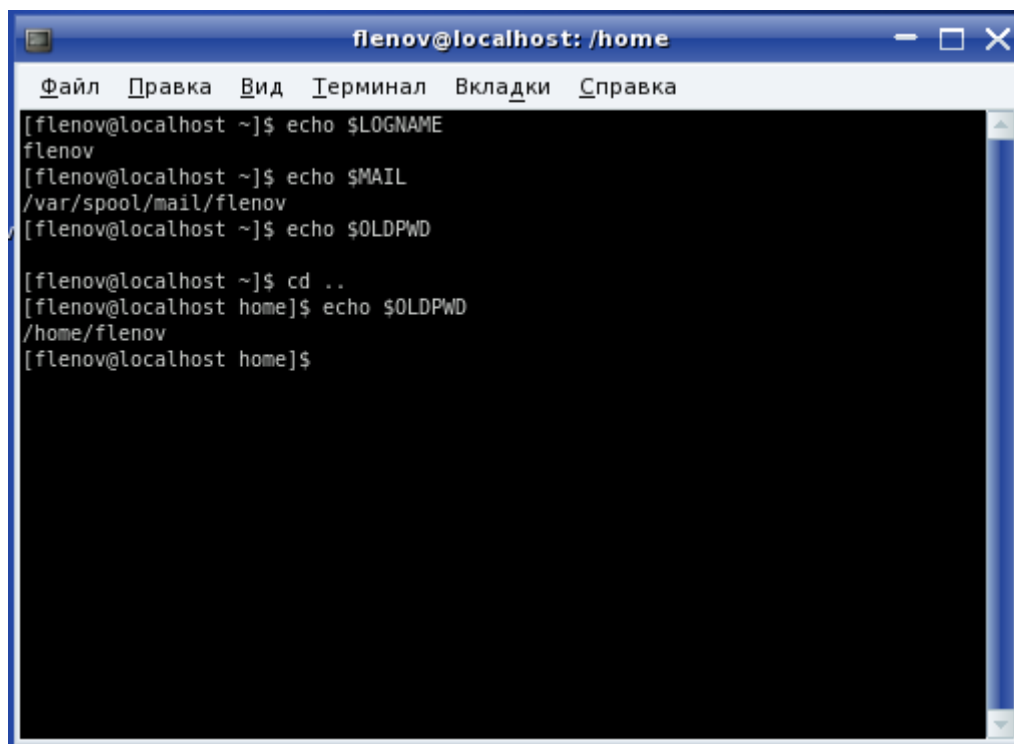
Эта переменная содержит путь к вашей домашней директории.

## **\$PWD**

В этой переменной находится путь к текущей директории. Выполнение команды `echo $PWD` идентично выполнению команды `pwd`.

## **\$OLDPWD**

Путь к директории, в которой вы были только что. Например, если вы были в своей домашней директории, а потом перешли на уровень выше (выполнили команду `cd ..`), то текущей директорией будет `/home`, а в переменной `$OLDPWD` будет домашний каталог (для меня `/home/flenov`).



```
flenov@localhost: /home
Файл  Правка  Вид  Терминал  Вкладки  Справка
[flenov@localhost ~]$ echo $LOGNAME
flenov
[flenov@localhost ~]$ echo $MAIL
/var/spool/mail/flenov
[flenov@localhost ~]$ echo $PWD
/home
[flenov@localhost ~]$ cd ..
[flenov@localhost home]$ echo $OLDPWD
/home/flenov
[flenov@localhost home]$
```

Если вы хотите вернуться в директорию, в которой были ранее (как бы нажать кнопку «назад»), выполните команду

```
> cd $OLDPWD
```

## ***\$SHELL***

В этой переменной храниться путь к вашему командному процессору. В Linux есть несколько командных процессоров, но самый популярный bash, а в переменной вы увидите путь /bin/bash.

## ***\$USER***

Переменная содержит имя текущего пользователя.

## ***Переменные пользователя***

Помимо системных переменных, вы можете использовать и собственные переменные, определенные пользователем. Допустим, что мы хотим создать переменную, которая будет равна значению числа Пи. Я точно не помню это число, но допустим, что оно равно 3.14. Мы не математики, поэтому погрешность в трамвайную остановку не повлияет на создание самой переменной. Короче, создадим переменную с именем TESTVAR и со значением 3.14. Для этого просто выполняем команду:

```
> TESTVAR=3.14
```

Обратите внимание, что при объявлении переменной не нужно перед именем ставить знак доллара. Выполните Следующую команду, дабы убедиться, что все на месте и корректно:

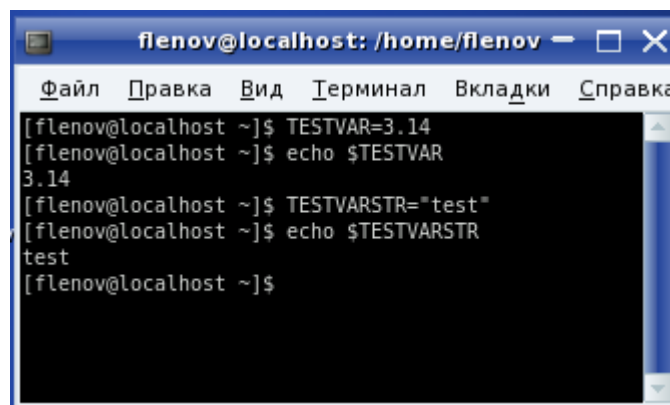
```
> echo $TESTVAR
```

Теперь уже символ доллара в начале имени необходим.

При создании строковой переменной, строку нужно заключать в двойные кавычки:

```
> TESTVARSTR="Test"
```

```
> echo $TESTVARSTR
```



```
flenov@localhost: /home/flenov
Файл  Правка  Вид  Терминал  Вкладки  Справка
[flenov@localhost ~]$ TESTVAR=3.14
[flenov@localhost ~]$ echo $TESTVAR
3.14
[flenov@localhost ~]$ TESTVARSTR="test"
[flenov@localhost ~]$ echo $TESTVARSTR
test
[flenov@localhost ~]$
```

Переменные пользователей видны только в командном процессоре, в котором их объявили. Чтобы их увидели и другие программу, необходимо экспортировать переменную с помощью команды export:

```
> export TESTVARSTR
```

Вы можете одновременно создавать и тут же экспортировать переменную:

```
> export TESTVARSTR=777
```

## Файловая система

Команды работы с файловой системой наиболее часто используемые не только администратором, но и простым пользователем.

### PWD

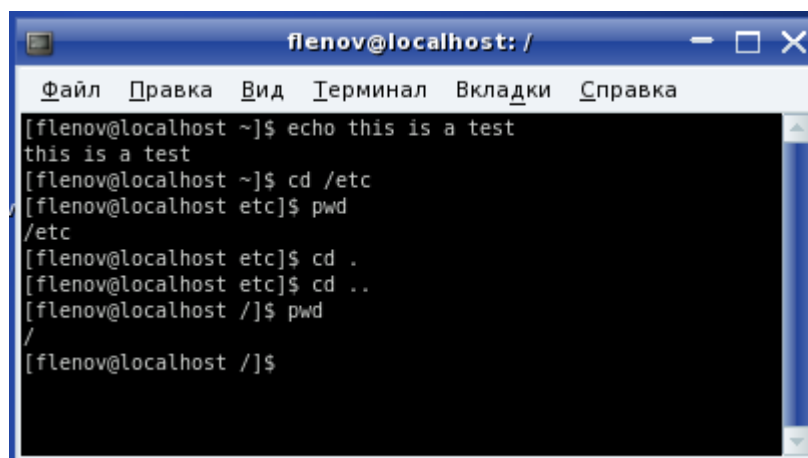
Эта команда возвращает текущую директорию. Например, если вы находитесь в директории /home/root, то результат работы команды будет:

```
> pwd  
/home/root
```

### CD

Первая команда наверно самая популярная – cd. Она позволяет перемещаться по файловой системе, а именно, изменять текущий каталог. Команда расшифровывается как change directory (наверно ☺, я точно не уверен, но мне так кажется). Итак, если вы хотите изменить директорию и перейти в каталог /etc, то нужно набрать:

```
> cd /etc
```



Судя по снимку, после выполнения команды cd /etc мы оказываемся в соответствующей директории. Это подтверждает и команда pwd.

Имя точка, указывает на текущий каталог. Если выполнить команду cd и указать точку, то ничего не произойдет, потому что переход в текущий каталог не имеет смысла. Имя из двух точек указывает на каталог уровня выше. Это значит, что если выполнить следующую команду, то вы перейдете в родительский каталог:

```
> cd ..
```

На рисунке выше, находясь в директории /etc, мы выполняем переход в родительскую папку. Для /etc родительским каталогом является корень /. Именно в него мы и попадаем и это тоже подтверждает команда pwd.

Путь команде `cd` можно указывать как полностью, так и относительно текущего каталога. Например, если вы находитесь в `/home`, то для перехода в каталог `/home/root` можно выполнить одну из двух команд:

```
> cd /home/root
> cd root
```

Обе команды идентичны, но только если текущий каталог это `/home`. В первом случае система видит полный путь. Поэтому, где бы вы не находились в данный момент, следующим вашим местоположением будет `/home/root`. Конечно, если этот путь существует.

Во второй команде указано только имя папки без слешей. Такое имя система будет искать внутри текущего каталога. Если он есть, то вы окажетесь в нем, иначе увидите сообщение об ошибке.

Ранее я вам уже показал переменные `$HOME` и `~`. Обе они указывают на домашний каталог текущего пользователя. Если вы хотите перейти в свой каталог, то нет смысла писать его полный путь. Достаточно просто выполнить одну из команд:

```
> cd $HOME
> cd ~
```

## LS

Команда `ls` выводит список файлов и подкаталогов указанной директории. Если имя каталога (файла) отсутствует в параметрах команды, то отображается содержимое текущего каталога. По умолчанию все настроечные файлы (имена начинаются с точки) являются скрытыми. Чтобы их вывести, нужно указать ключ `-a`:

```
ls -a
```

Если мы кроме этого хотим увидеть не только имена (сжатый формат), но и полную информацию о каталоге, нужно добавить ключ `-l`. В результате мы должны выполнить команду:

```
ls -al
```

Но такая команда отобразит файлы текущего каталога, и не факт, что мы сейчас находимся, например, в каталоге `/etc`, который надо просмотреть. Чтобы увидеть именно его, после ключей (можно и до них) нужно указать требуемую папку:

```
ls -al /etc
```

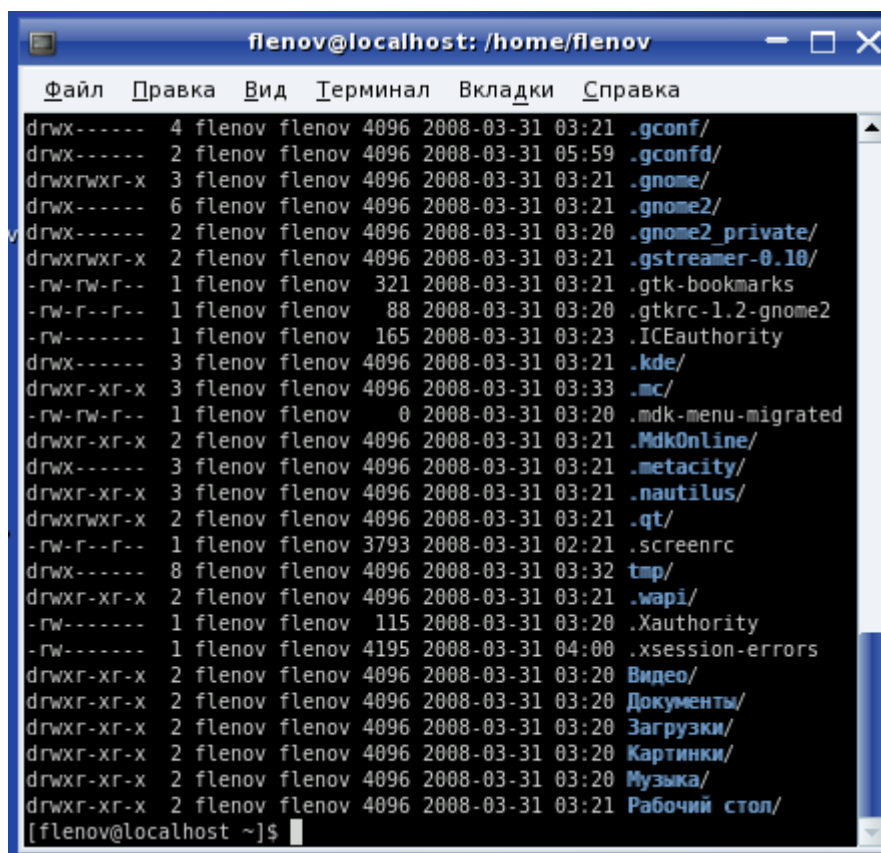
### *Примечание*

Более подробную информацию о команде `ls` можно получить из справочной системы. Для этого выполните команду `man ls`.

Рассмотрим результат вывода команды `ls -al`:

```
drwx----- 3 Flenov FlenovG 4096 Nov 26 16:10 .
drwxr-xr-x  5 root    root    4096 Nov 26 16:21 ..
-rw-r--r--  1 Flenov FlenovG  124 Nov 26 16:10 .bashrc
-rw-r--r--  1 Flenov FlenovG 2247 Nov 26 16:10 .emacs
-rw-r--r--  1 Flenov FlenovG  118 Nov 26 16:10 .gtkrc
```

```
drwxr-xr-x  4 Flenov  FlenovG  4096 Nov 26 16:10 .kde
```



По умолчанию список файлов выводится в несколько колонок. Разберем них на примере первой строки:

- ☐ `drwx-----` — права доступа. Их мы подробно разберем в главе 4. Сейчас вам главное знать, что если первая буква "d", то это директория;
- ☐ цифра 3 — указывает количество жестких ссылок;
- ☐ `Flenov` — имя пользователя, являющегося владельцем файла;
- ☐ `FlenovG` — группа, которой принадлежит файл;
- ☐ 4096 — размер файла.

#### *Примечание*

На первый взгляд, директория не файл и не имеет размера, но не стоит удивляться, что размер директории не равен нулю. На самом деле, директории похожи на файлы, в которых находится список файлов директории. Размер тоже не случаен — четыре кило (4\*1024), что равно блоку памяти (странице), выделяемому для работы с данными. Это лирическое отступление, которое может и не пригодиться в реальной жизни.

- ☐ дата и время последних изменений файла;
- ☐ имя файла.

## **cat**

Команда позволяет вывести на экран содержимое указанного в качестве аргумента файла. Например, вы хотите просмотреть текстовый файл `need.txt`. Для этого нужно выполнить команду:

```
cat need.txt
```

Но это справедливо, если файл находится в текущей директории. А если нет? В этом случае можно указать полный путь:

```
cat /home/root/need.txt
```

## **tac**

Эта команда обратная для `cat` (даже название команды — это слово `cat` наоборот), т. е. выводит на экран файл в обратном порядке, начиная с последней строки до первой.

## **cp**

Команда копирования файла. С ее помощью можно выполнять несколько различных действий:

1. Копирования содержимого файла в другой документ той же папки:

```
cp /home/root/need.txt /home/root/need22.txt
```

Здесь содержимое файла **/home/root/need.txt** (источник) будет скопировано в файл **/home/root/need22.txt** (назначение).

2. Копирования файла в другой каталог:

```
cp /home/root/need.txt /home/flenov/need.txt
```

или

```
cp /home/root/need.txt /home/flenov/need22.txt
```

Обратите внимание, что в этом случае в папке назначения файл может быть как с новым, так и со старым именем.

3. Копирование несколько файлов в новый каталог. Для этого нужно перечислить все файлы в источнике и последним параметром указать папку:

```
cp /home/root/need.txt /home/root/need22.txt /home/new/
```

В этом примере файлы **/home/root/need.txt** и **/home/root/need22.txt** будут скопированы в директорию **/home/new**. Можно копировать файлы и из разных каталогов в один:

```
cp /home/root/need.txt /home/flenov/need22.txt /home/new/
```

В этом примере файлы **/home/root/need.txt** и **/home/flenov/need22.txt** будут скопированы в директорию **/home/new**.

4. Копирование группы (всех) файлов каталога.

А что если надо скопировать все файлы, начинающиеся на букву "n" из одной директории в другую? Неужели придется их все перечислять? Нет, достаточно указать маску `n*`, где звездочка заменяет любые символы, начиная со второго:

```
cp /home/root/n* /home/new/
```

Если нужно скопировать все файлы, имена которых начинаются символами "ra" и заканчиваются буквой "t", то маска будет выглядеть как `ra*t`.

Это далеко не полный список возможностей команды копирования. Она достаточно сложная и мощная, что позволяет выполнить в один присест любую задачу копирования.

## ***mkdir***

Создание новой директории. Например, если вы хотите создать подкаталог **newdir** в текущей директории, то нужно выполнить команду:

```
mkdir newdir
```

## ***rm***

Команда позволяет удалить файл или директорию (должна быть пустая):

```
rm /home/flenov/need22.txt
```

В качестве имен файлов можно использовать и маски, как в команде `cp`. Для удаления директории может понадобиться указание следующих ключей:

- d — удалить директорию.
- r — рекурсивно удалять содержимое директорий.
- f — не запрашивать подтверждение удаляемых файлов. Будьте внимательны при использовании этого параметра, потому что файлы будут удаляться без каких-либо предупреждений. Вы должны быть уверены, что команда написана правильно, иначе можно удалить что-то лишнее, особенно, если вы работаете под учетной записью root.

Пример удаления директории рекурсивно и без запроса на подтверждение:

```
rm -rf /home/flenov/dir
```

## ***df***

Эта команда позволяет узнать свободное место на жестком диске или разделе. Если устройство не указано, то на экран выводится информация о смонтированных файловых системах.

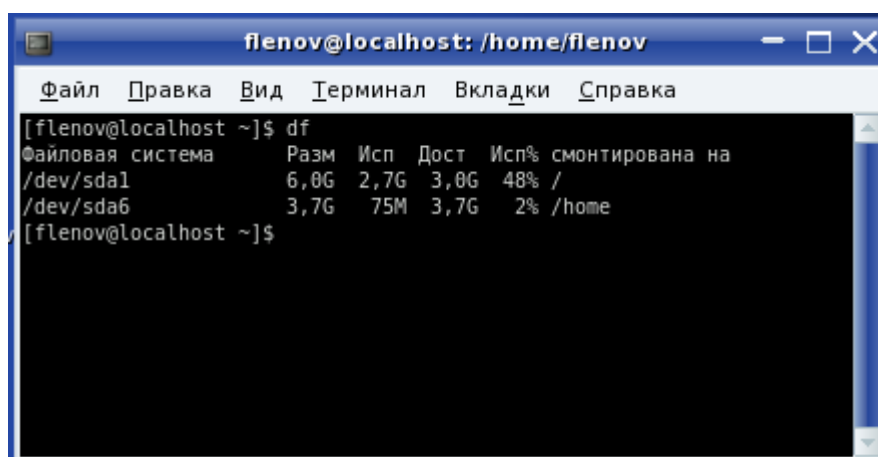
Пример результата выполнения команды:

Filesystem	1k-blocks	Used	Available	Use%	Mounted on
/dev/hda2	16002200	2275552	12913760	15%	/
none	127940	0	127940	0%	/dev/shm



Результирующая таблица состоит из следующих колонок:

- ❑ `Filesystem` — диск, файловая система которого смонтирована;
- ❑ `1k-blocks` — количество логических блоков;
- ❑ `Used` — количество использованных блоков;
- ❑ `Available` — количество доступных блоков;
- ❑ `Use%` — процент использованного дискового пространства;
- ❑ `Mounted on` — как смонтирована файловая система.



```
flenov@localhost: /home/flenov
Файл  Правка  Вид  Терминал  Вкладки  Справка
[flenov@localhost ~]$ df
Файловая система    Разм  Исп  Дост  Исп% смонтирована на
/dev/sda1            6,0G  2,7G  3,0G  48% /
/dev/sda6            3,7G   75M  3,7G   2% /home
[flenov@localhost ~]$
```

## **mount**

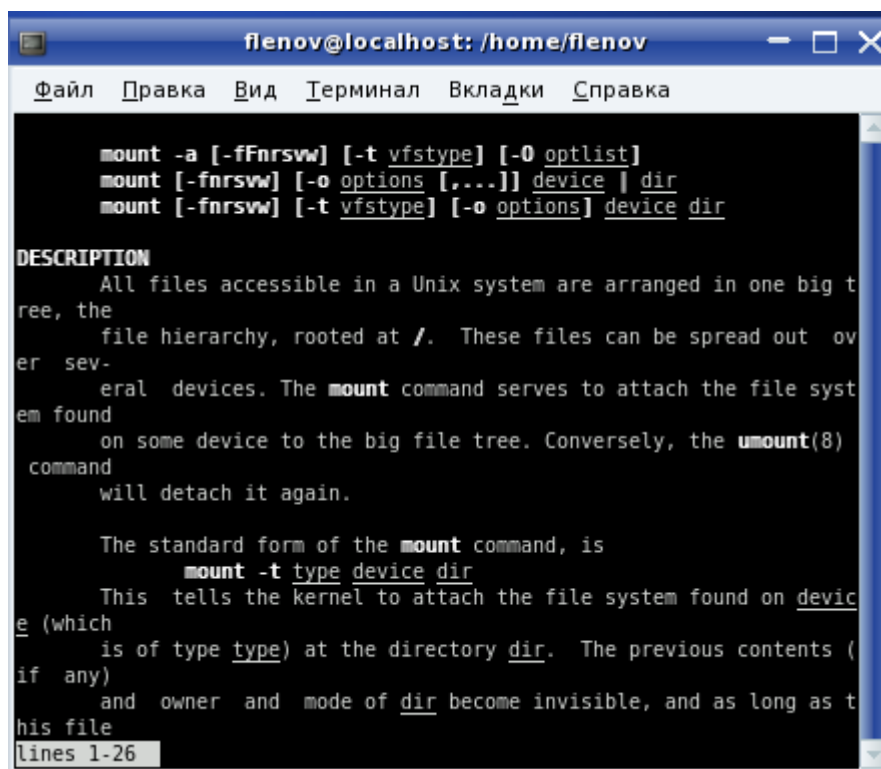
Команда предназначена для монтирования файловых систем. Она достаточно сложна, и ее используют системные администраторы.

Если вы работали с ОС Windows, то скорее всего привыкли к тому, что дискеты, CD-диски и другие съемные носители становятся доступными сразу же, как только вы поместили их в устройство чтения. В Linux это не так, по крайней мере в текстовом режиме, и многие не могут сжиться с этой особенностью. Графические оболочки прекрасно научились монтировать диски автоматом, но все же желательно помнить и хоть немного уметь пользоваться утилитой.

Итак, чтобы CD-ROM стал доступным, надо выполнить команду `mount`, указав в качестве параметра устройство **/dev/cdrom**:

```
mount /dev/cdrom
```

После этого содержимое CD можно посмотреть в директории **/mnt/cdrom**. Получается, что файлы и директории диска как бы сливаются с файловой системой.



## **umount**

Когда вы подключили к файловой системе CD-ROM, то это устройство блокируется, и диск нельзя вытащить, пока он не будет размонтирован. Для этого используется команда `umount`.

Например, следующая команда позволяет размонтировать CD-ROM:

```
umount /dev/cdrom
```

Конечно, вытащить нельзя CD-ROM диск, но если была смонтирована флешка, то ее всегда можно выдернуть физически, и тут ничего не спасет.

## **fdformat**

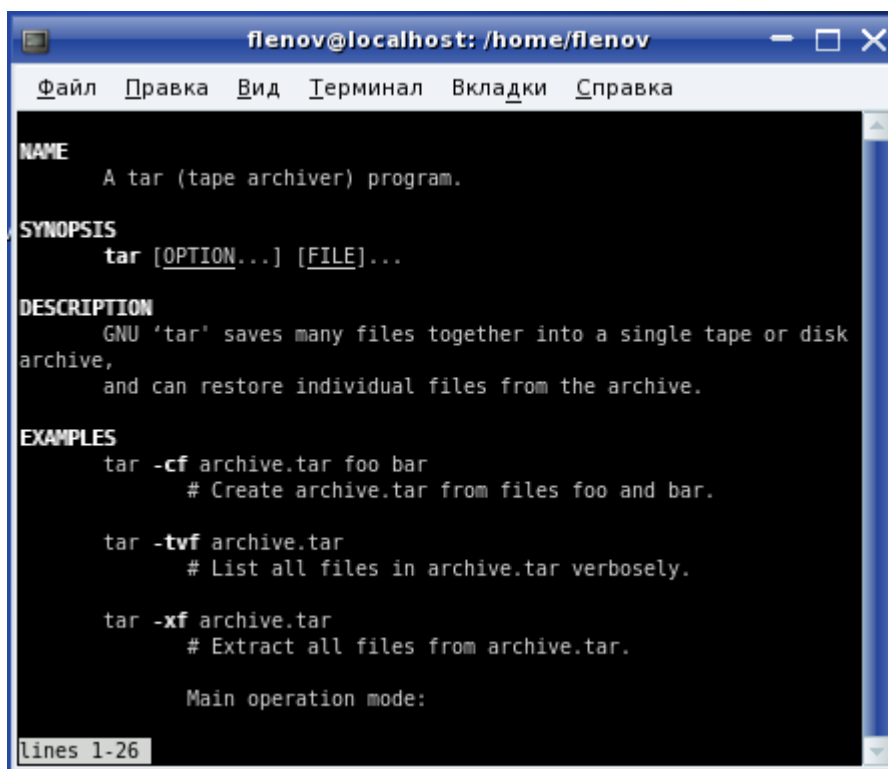
Перед использованием дискет их нужно отформатировать. В ОС Linux для этого используется команда `fdformat`. Конечно, дискеты уже почти не используются, но вдруг вам не повезло с работой и техникой.

## **tar**

По ходу изложения данной книги мы иногда будем устанавливать различные программы, часть из них поставляется в виде архивов `tar.gz`. Чаще всего это программы, хранимые в исходных кодах. Для развертывания такого файла нужно выполнить команду:

```
tar xzvf имяфайла.tar.gz
```

Как правило, после выполнения команды в текущей директории будет создан каталог с таким же именем, как у архива (только без расширения). В нем вы сможете найти все распакованные файлы.



```
flenov@localhost: /home/flenov
Файл  Правка  Вид  Терминал  Вкладки  Справка

NAME
  A tar (tape archiver) program.

SYNOPSIS
  tar [OPTION...] [FILE]...

DESCRIPTION
  GNU 'tar' saves many files together into a single tape or disk
  archive,
  and can restore individual files from the archive.

EXAMPLES
  tar -cf archive.tar foo bar
      # Create archive.tar from files foo and bar.

  tar -tvf archive.tar
      # List all files in archive.tar verbosely.

  tar -xf archive.tar
      # Extract all files from archive.tar.

  Main operation mode:

lines 1-26
```

## ***rpm***

В настоящее время большинство программ поставляются уже не в исходных кодах, а в виде rpm-пакетов. Их установка намного проще, т.к. программы в них уже скомпилированы. Если вы используете МС, то выберите rpm-пакет и нажмите клавишу <Enter>. Таким образом, вы войдете в него как в директорию и увидите содержимое.

Каждый пакет обязательно содержит исполняемый файл install. Запустите его для установки пакета. Или запустите upgrade для обновления уже установленного пакета.

Если вы не используете МС, то для установки нового пакета можно выполнить команду:

```
rpm -i пакет
```

Для обновления уже установленного пакета можно выполнить команду с параметром -U:

```
rpm -U пакет
```

Для того чтобы видеть ход инсталляции, можно указать еще и ключ -v. Таким образом, команда установки будет выглядеть следующим образом:

```
rpm -iv пакет
```

## ***which***

Иногда необходимо знать каталог, в котором расположена программа. Для этого используется команда `which` с именем программы в качестве параметра, которая проверит основные каталоги, содержащие исполняемые файлы. Например, чтобы определить, где находится программа просмотра содержимого каталогов `ls`, выполните следующую команду:

```
which ls
```

В результате вы увидите путь **/bin/ls**. Если ваша ОС поддерживает псевдонимы (alias) команд, то можно будет увидеть и его. Таким образом, после выполнения команды на экране выведется:

```
alias ls='ls -color=tty'
      /bin/ls
```

## Работа с журналом

Журнал позволяет администратору узнать об активности внутри системы, и в случае непредвиденной ситуации узнать причину и как хакер проник в систему. Эта информация поможет исправить ошибку в конфигурации или найти программу/сервис, требующую обновления.

Помимо этого, журнал может помочь вам узнать, кто именно сейчас подключен к серверу и что делает. Это так же может быть полезным при мониторинге сервера.

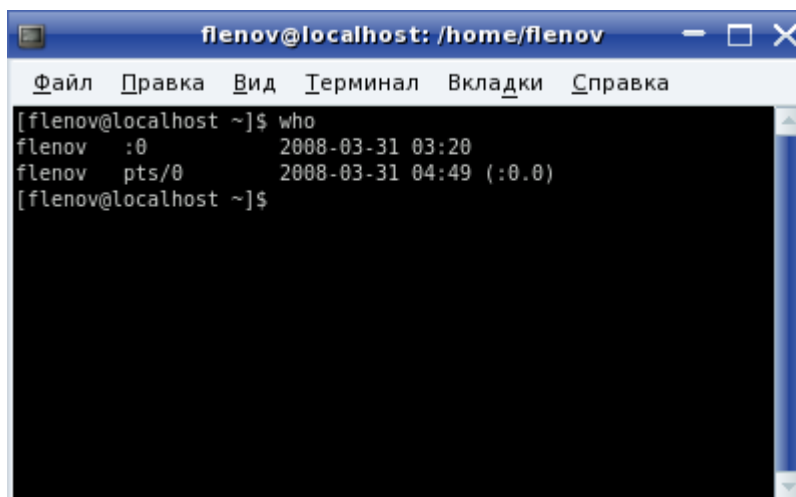
В этом разделе мы рассмотрим команды, которые помогут вам в нелегкой работе с журналом.

### *who*

Команда `who` позволяет узнать, кто сейчас зарегистрирован в системе и сколько времени в ней находится. Информация достается из файла `/var/run/utmp`. Введите эту команду, и на экране появится список примерно следующего вида:

```
robert      tty1      Dec  8 10:15
root        tty2      Dec  8 11:07
```

Из этого списка становится ясно, что пользователь `robert` работает за первым терминалом (`tty1`) и вошел в систему 8 декабря в 10 часов 15 минут.



Большинство хакеров при входе в систему выполняют эту команду, чтобы выяснить, есть ли сейчас в системе администратор. Если пользователь `root` присутствует, то начинающие хакеры стараются уйти, т. к. опасаются, что их знаний не хватит, чтобы остаться незамеченными.

Это еще одна причина, по которой администратор не должен входить в систему под учетной записью `root`. Лучше всего работать как простой пользователь, а когда не хватает прав, то переключаться на привилегированного. На такой случай я создал учетную запись, для которой установил UID равный нулю. Она позволяет получить доступ ко всей

системе, и при этом имеет имя отличное от root, и не вызовет подозрений, когда я буду работать. Так что, в моем случае никогда нельзя увидеть пользователя root.

## users

Эта команда позволяет вытащить из журнала **/var/run/utmp** список всех пользователей, которые сейчас зарегистрированы в системе.

В журнале **/var/run/utmp** информация хранится временно, только на момент присутствия пользователя. Когда он выходит из системы, соответствующая запись удаляется. После этого выяснить, кто и когда работал можно только по журналу **/var/log/wtmp**. Это также бинарный файл, поэтому его содержимое можно увидеть с помощью специализированных программ.

## last

Команда позволяет выяснить, когда и сколько времени определенный пользователь находился в системе. В качестве параметра передается интересующее имя. Например, следующая директива отображает время входа и продолжительность нахождения в системе пользователя robert:

```
last robert
```

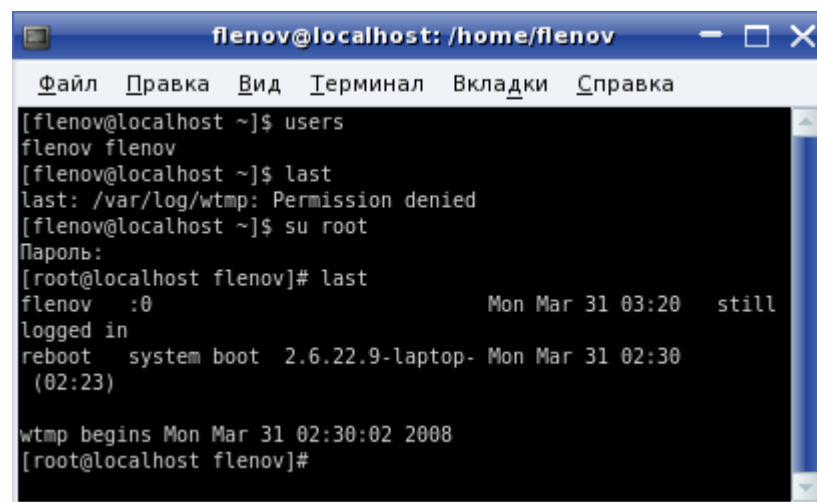
Выполнив команду, вы увидите на экране примерно следующий список:

```
robert          tty1          Thu Dec 2 12:17 — 12:50 (00:33)
```

По этой записи можно понять, что robert находился за терминалом (tty1), зашел в систему 2 декабря на 33 минуты (с 12:17 до 12:50). Если пользователь работал не локально, а через сеть, то будет отображена информация о хосте, с которого входили в систему.

Если выполнить эту команду для себя, то может вывалиться такой список, что читать его будет невозможно, потому что вы достаточно часто работаете в системе. Чтобы ограничить выводимые данные, можно указать ключ **-n** и количество отображаемых строк. Например, следующая команда выдаст информацию о последних пяти входах:

```
last -n 5 robert
```



```
flenov@localhost: /home/flenov
Файл  Правка  Вид  Терминал  Вкладки  Справка
[flenov@localhost ~]$ users
flenov flenov
[flenov@localhost ~]$ last
last: /var/log/wtmp: Permission denied
[flenov@localhost ~]$ su root
Пароль:
[root@localhost flenov]# last
flenov  :0                               Mon Mar 31 03:20   still
logged in
reboot  system boot  2.6.22.9-laptop- Mon Mar 31 02:30
(02:23)

wtmp begins Mon Mar 31 02:30:02 2008
[root@localhost flenov]#
```

На рисунке выше я попытался выполнить команду `last` под правами простого смертного, на что система на меня неплохо выругалась. Пришлось переключиться на `root`, после чего команда выполнилась без проблем.

## ***lastlog***

Если выполнить команду `lastlog`, то она выведет на экран перечень всех пользователей с датами их последнего подключения к системе. Пример списка:

Username	Port	From	Latest
root	ftpd2022	192.168.77.10	Mon Feb 21 12:05:06 +0300 2005
bin			**Never logged in**
daemon			**Never logged in**
adm			**Never logged in**
lp			**Never logged in**
sync			**Never logged in**
shutdown			**Never logged in**
halt			**Never logged in**
mail			**Never logged in**
news			**Never logged in**
uucp			**Never logged in**
operator			**Never logged in**
games			**Never logged in**
gopher			**Never logged in**
ftp			**Never logged in**
nobody			**Never logged in**
vcsa			**Never logged in**
mailnull			**Never logged in**
rpm			**Never logged in**
xfs			**Never logged in**
apache			**Never logged in**
ntp			**Never logged in**
rpc			**Never logged in**
gdm			**Never logged in**
rpcuser			**Never logged in**
nscd			**Never logged in**
ident			**Never logged in**
radvd			**Never logged in**
squid			**Never logged in**

```
mysql                                **Never logged in**
flenov          ftpd2022 192.168.77.10  Mon Feb 21 12:05:06 +0300 2005
named                                **Never logged in**
robert          tty1                Mon Feb 21 12:10:47 +0300 2005
```

```
flenov@localhost ~]$ lastlog
/var/log/lastlog: Отказано в доступе
flenov@localhost ~]$ su root
Пароль:
[root@localhost flenov]# lastlog
Пользователь  Порт  С  Последний раз
root          **Никогда не входил в систему**
bin           **Никогда не входил в систему**
daemon        **Никогда не входил в систему**
adm           **Никогда не входил в систему**
lp            **Никогда не входил в систему**
sync          **Никогда не входил в систему**
shutdown      **Никогда не входил в систему**
halt          **Никогда не входил в систему**
mail          **Никогда не входил в систему**
news          **Никогда не входил в систему**
uucp          **Никогда не входил в систему**
operator      **Никогда не входил в систему**
games         **Никогда не входил в систему**
rpm           **Никогда не входил в систему**
messagebus    **Никогда не входил в систему**
avahi         **Никогда не входил в систему**
haldaemon     **Никогда не входил в систему**
vcsa          **Никогда не входил в систему**
apache        **Никогда не входил в систему**
postfix       **Никогда не входил в систему**
rpc           **Никогда не входил в систему**
gdm           **Никогда не входил в систему**
rpcuser       **Никогда не входил в систему**
nscd          **Никогда не входил в систему**
sshd          **Никогда не входил в систему**
ftp           **Никогда не входил в систему**
flenov        :0  Пнд Мар 31 03:20:47 +0400 2008
root          **Никогда не входил в систему**
bin           **Никогда не входил в систему**
daemon        **Никогда не входил в систему**
adm           **Никогда не входил в систему**

```

Список состоит из четырех колонок:

- ☐ имя пользователя из файла `/etc/passwd`;
- ☐ порт или терминал, на который происходило подключение;
- ☐ адрес компьютера, если вход был по сети;
- ☐ время входа.

С помощью `lastlog` удобно контролировать системные записи. У них дата последнего входа должна быть `**Never logged in**`, потому что под ними нельзя войти в систему (в качестве командной оболочки установлены `/bin/false`, `/dev/null`, `/sbin/nologin` и др.). Если



вы заметили, что кто-либо проник в систему через одну из этих учетных записей, то это значит, что хакер использует ее, изменив настройки.

Простая замена командной оболочки в файле **/etc/passwd** может открыть хакеру потайную дверь, и администратор не заметит этой трансформации. Но после выполнения команды `lastlog` все неявное становится явным.

Обращайте внимание на тип подключения и адрес. Если что-то вызывает подозрение, то можно выявить атаку на этапе ее созревания.

## ***lsuf***

С помощью этой команды можно определить, какие файлы и какими пользователями открыты в данный момент. Результат выполнения команды:

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE	NAME
init	1	root	cwd	DIR	3,2	4096	2	/
init	1	root	rtd	DIR	3,2	4096	2	/
init	1	root	txt	REG	3,2	26920	635256	/sbin/init
init	1	root	mem	REG	3,2	89547	553856	/lib/ld-2.2.5.so
init	1	root	10u	FIFO	3,2		195499	/dev/initctl
keventd	2	root	cwd	DIR	3,2	4096	2	/
keventd	2	root	rtd	DIR	3,2	4096	2	/
kapmd	3	root	10u	FIFO	3,2		195499	/dev/initctl

Это далеко не полный результат. Даже если в данный момент вы один работаете с системой, количество открытых файлов может исчисляться парой десятков, и число их заметно растет, если в системе несколько пользователей, ведь один файл может открываться несколько раз каждым из них. Это касается в основном системных конфигурационных файлов.

## Задачи

ОС Linux многозадачная система, а это значит, что в ней одновременно может выполняться сразу несколько задач. Несколько задач может работать даже в одном терминале. Как запустить команду в фоне?

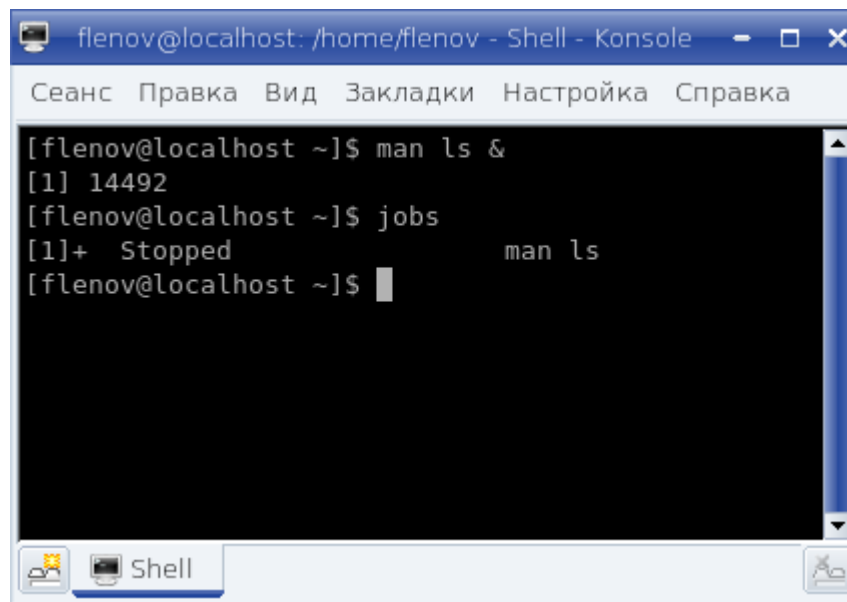
Если команда работает очень долго, то вы можете запустить ее на выполнение в фоновом режиме. Для этого используется символ `&` который нужно поставить в конце команды. Например, у меня на работе есть программа сервер OLAP отчетности, которая написана на Java и при запуске захватывает консоль. Это значит, что консоль блокируется программой, и пока программа не будет завершена, я не могу закрыть консоль или выполнить в ней какую-либо команду. Если таким образом запустить несколько серверов, то на рабочем столе количество консолей начинает расти. Чтобы избавиться от этого, достаточно после команды поставить символ `&`:

```
> start-olap-server.sh &
```

Теперь программа запускается на выполнение, но не блокирует консоль. Да, вы видите вывод на экране, но в любой момент может забрать консоль себе, нажав Enter. Вы можете закрывать консоль, и стартующая программа не прервет свою работу.

В книге Linux глазами хакера я подробно рассказывал о работе с командами, выполняемыми в фоновом режиме. Здесь же мы немного пробежимся по этой теме еще раз, но более быстрыми шагами.

Чтобы просмотреть список программ, уже работающих в фоне, выполните команду `jobs`.



```
flenov@localhost: /home/flenov - Shell - Konsole
Сеанс  Правка  Вид  Закладки  Настройка  Справка

[flenov@localhost ~]$ man ls &
[1] 14492
[flenov@localhost ~]$ jobs
[1]+  Stopped                  man ls
[flenov@localhost ~]$
```

Число, которое вы видите в квадратных скобках – это номер выполняемой задачи. В фон можно отправить команду нажатием `Ctrl+Z`. В этом случае команда уходит в фон, но перестает выполняться.

Знак плюса слева от имени программы – показывает активную запись. Если выполнить команду `fg`, то именно эта команда станет активной. Да, если команду нужно перевести из фона на передний план, нужно выполнить `fg`. Если нужно вывести на передний план не активную, а определенную задачу, то после команды `fg` укажите номер нужной программы.

## Команды общего назначения

В этом разделе мы поговорим об общих командах и просто командной оболочке. Эта информация поможет вам лучше понять и эффективнее использовать командную строку в ОС Linux.

### Псевдонимы

С помощью команды `alias` можно создавать псевдонимы для команд. Например, если вы часто выполняете какое-то действие с длинной командой и кучей параметров, то вы можете создать для нее псевдоним. Например, команда для просмотра текущего каталога используется команда `ls`, но она показывает короткое содержимое, без прав доступа, что иногда бывает очень нужным. Каждый раз писать `ls` и параметр `-l` нудно, поэтому можно создать псевдоним `ll`:

```
alias ll="ls -l"
```

Теперь, выполняя команду `ll` будет идентично выполнению `ls -l`. Но не торопитесь делать именно этот псевдоним в своей системе. Вполне возможно, что она уже есть у вас, по крайней мере, у меня в дистрибутиве есть.

### Перенаправление ввода вывода

Теперь поговорим о том, как можно направлять выходные данные команды не на экран, а, например, в файл. Для этого используем символ `>`. Например, выполните команду:

```
> ls > outfile.txt
```

В результате выполнения этой команды, результат (т.е. содержимое каталога) не будет выведено на экран, а будет сохранено в файл.

Если отобразить содержимое файла:

```
> cat outfile.txt
```

То вы увидите то, что должно было быть на экране еще после выполнения той команды.

Мы можем не только выводить результат в файл, но и получать параметры из файла. Например, можно выполнить команду:

```
> cat < infile.txt
```

Команде `cat` в качестве параметров будут передано содержимое `infile.txt`.

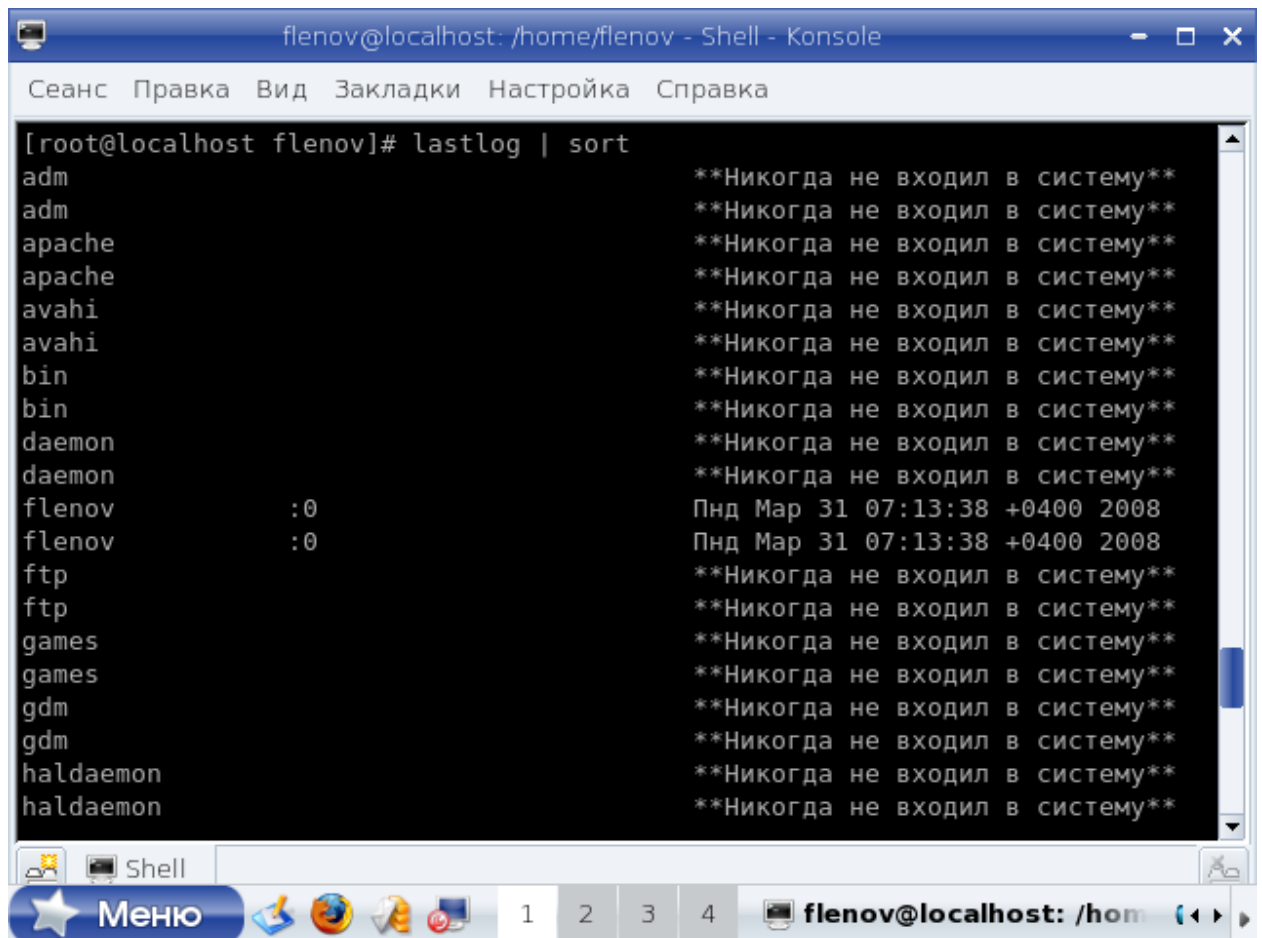
### Объединение ввода

С помощью символа вертикальной черты `|` вы можете изменять стандартный поток ввода вывода. Допустим, что вы хотите узнать, когда в систему последний раз входил пользователь по имени `flenov`. Для этого можно выполнить команду `lastlog`. Но если в системе зарегистрировано 1000 пользователей, то найти нужного будет проблематично. Он может находиться в любом месте, ведь результат не отсортирован. Представьте себе,

если бы телефонная книга не содержала всех абонентов в отсортированном виде! Это была бы катастрофа.

Как отсортирован вывод команды? Да очень просто, нужно направить список пользователей команде `sort`, которая сортирует входящие данные и выводит их в отсортированном виде. А это можно сделать с помощью символа вертикальной черты:

```
> lastlog | sort
```



```
flenov@localhost: /home/flenov - Shell - Konsole
Сеанс  Правка  Вид  Закладки  Настройка  Справка

[root@localhost flenov]# lastlog | sort
adm                **Никогда не входил в систему**
adm                **Никогда не входил в систему**
apache            **Никогда не входил в систему**
apache            **Никогда не входил в систему**
avahi             **Никогда не входил в систему**
avahi             **Никогда не входил в систему**
bin               **Никогда не входил в систему**
bin               **Никогда не входил в систему**
daemon            **Никогда не входил в систему**
daemon            **Никогда не входил в систему**
flenov             :0                Пнд Мар 31 07:13:38 +0400 2008
flenov             :0                Пнд Мар 31 07:13:38 +0400 2008
ftp               **Никогда не входил в систему**
ftp               **Никогда не входил в систему**
games             **Никогда не входил в систему**
games             **Никогда не входил в систему**
gdm               **Никогда не входил в систему**
gdm               **Никогда не входил в систему**
haldaemon         **Никогда не входил в систему**
haldaemon         **Никогда не входил в систему**

Меню  1  2  3  4  flenov@localhost: /home
```

## Последовательность команд

А что если вам нужно просто выполнить последовательность команд? В этом случае вы пишете эти команды через точку с запятой. Например:

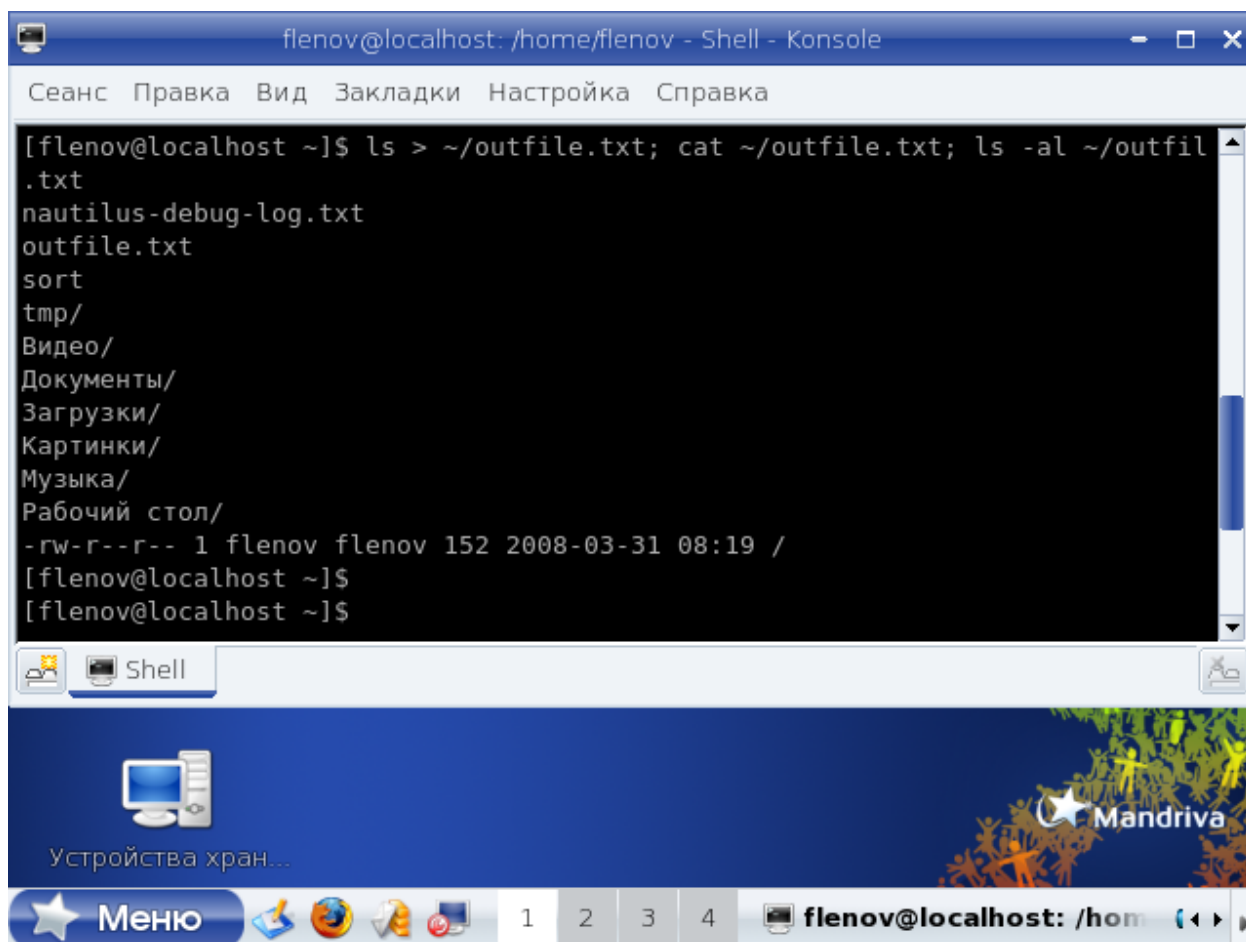
```
> ls > ~/outfile.txt; cat ~/outfile.txt; ls -al ~/outfile.txt
```

Эта команда идентична вводу трех отдельных команд:

```
> ls > ~/outfile.txt
```

```
> cat ~/outfile.txt
```

```
> ls -al ~/outfile.txt
```



```
[flenov@localhost ~]$ ls > ~/outfile.txt; cat ~/outfile.txt; ls -al ~/outfil
.txt
nautilus-debug-log.txt
outfile.txt
sort
tmp/
Видео/
Документы/
Загрузки/
Картинки/
Музыка/
Рабочий стол/
-rw-r--r-- 1 flenov flenov 152 2008-03-31 08:19 /
[flenov@localhost ~]$
[flenov@localhost ~]$
```

В первой строке я запрашиваю отображение файлов текущей директории и сохранение результата в файл outfile.txt в домашней директории пользователя (на домашнюю директорию указывает символ ~).

Вторая команда отображает содержимое созданного файла. Так как эта команда будет выполнена после завершения первой, файл уже будет существовать. Третья команда отображает параметры файла, его права доступа и время создания.

Если команды разделить с помощью символов && то вы как бы указываете, что обе команды должны быть выполнены. Если первая команда не выполниться, то вторая уже не будет выполняться. Например, если файл outfile2.txt не существует, то вторая команда не будет выполнена.

```
> ls ~/outfile2.txt && cat ~/outfile.txt
```

Если же нужно выполнить одну из команд, то разделите их символами двух вертикальных черточек: ||. Например:

```
> ls ~/outfile2.txt || cat ~/outfile.txt
```

Если файл outfile2.txt не существует, то будет выполнена вторая команда. Если же он существует, значит, первая команда выполниться корректно, а вторая не выполниться никогда.

## ***Советы по работе с командной строкой***

Небольшой секрет – когда вы набираете команды, то можно экономить время с помощью кнопки Tab. Нужно всего лишь начать набирать команду и нажать Tab. Например, если вы находитесь в корне, то там только одна директория на букву h и это home. Наберите букву h и нажмите Tab, оболочка сама допишет полное имя директории. То же самое касается и файлов.

Если вам нужно повторить выполнение введенной ранее команды, то можете использовать клавиши вверх или вниз. Нажав кнопку вверх, в командной строке появится последняя введенная команда. Нажмите еще раз, и увидите предпоследнюю. Таким образом, можно перемещаться по истории введенных команд.

*Данный документ написан Фленовым Михаилом и распространяется только на компакт диске к книге «Linux глазами хакера» или на сайтах автора. Вы можете копировать этот документ куда угодно, но **нельзя** выкладывать в Интернете или других носителях информации без согласия автора.*

*С автором можно связаться через персональный сайт <http://www.flenov.info>*