

Основные команды для работы с файлами

Команды для работы с файловой системой

Просмотр файловой системы - ls

Команда ls (list files) предназначена для просмотра содержимого каталогов и получения информации о файлах.

Примеры:

```
$ ls
$ ls -a
$ ls /bin
$ ls -l /
$ ls -l /dev
$ ls -l ..
$ ls -a .
```

Определение текущего каталога — pwd

pwd (print working directory) показывает текущий рабочий каталог.

Смена текущего каталога — cd

cd (change directory) меняет текущий рабочий каталог. Если команда cd вызвана без параметров, происходит переход в домашний каталог пользователя. Команда cd позволяет вернуться в предыдущий каталог (каталог, откуда был осуществлен переход в текущий).

Примеры:

```
$ cd /usr
$ cd bin
$ cd ..
```

```
$ cd  
$ cd ~  
$ cd ~u1
```

Создание файла с помощью команды touch

Команда touch меняет временные метки файла и имеет побочное действие, которое используется гораздо чаще, чем основное — если файла с указанным именем нет, создается пустой файл.

Создание каталога — mkdir

Команда mkdir (make directory) создает каталоги, в том числе и промежуточные (если указана опция -p).

Примеры:

```
$ mkdir /tmp/test  
$ mkdir ~p /tmp/a/b/c
```

Удаление файла — rm

Команда rm удаляет файлы и каталоги, в т.ч. каталоги с содержимым (опция r или R). Опция f подавляет вывод запросов (например, при удалении файлов, доступных только для чтения) и ошибок при удалении, а опция i выводит запрос при удалении каждого файла.

Примеры:

```
$ touch /tmp/test  
$ rm -r /tmp/test  
# rm -fr /*
```

Удаление каталога — rmdir

Команда `rmdir` удаляет только пустые каталоги. При использовании опции `r` удаляются и родительские каталоги целевого каталога, если они пусты.

Примеры:

```
$ rmdir /tmp/test  
$ rmdir -p /tmp/a/b/c
```

Копирование файлов и каталогов — `cp`

Команда `cp` позволяет копировать файлы и каталоги (опция `r` или `R`). При её использовании часто применяются шаблоны шелла. В общем случае, команда `cp` требует не менее двух параметров: что копировать и куда копировать. Команда `cp` имеет большое количество опций, подробно о которых можно узнать на странице [man](#).

Примеры:

```
$ touch file1  
$ mkdir dir1  
$ cp file1 file2  
$ cp file1 incorrectdirname  
$ cp file1 dir1/  
$ cp -r dir1/ dir2/
```

Перемещение и переименование файлов и каталогов — `mv`

Команда `mv` предназначена для перемещения и переименования файлов и каталогов. При перемещении внутри одного каталога имя исходного файла/каталога меняется на новое, что эквивалентно переименованию. При перемещении внутри одного раздела (одной файловой системы) меняется только жесткая ссылка на объект и процесс перемещения происходит очень быстро. При перемещении данных между различными файловыми системами происходит копирование с последующим удалением источника, так что время выполнения команды зависит от объема данных.

Примеры:

```
$ mv file2 file3
$ mv dir2 dir3
$ mv file3 incorrectdirname
$ mv file3 dir1/
$ mv dir3 dir1/
```

Создание линков/ссылок на файлы и каталоги - ln

Команда ln позволяет создавать символичные (с опцией s) и жесткие (без опции s) ссылки.

Примеры:

```
$ ln -s /etc/rc.conf file4

$ mkdir -p ~/var/db/mysql
$ touch ~/var/db/mysql/file.db

$ mkdir ~/disk2
$ mv ~/var/db/mysql ~/disk2/

$ ln -s ~/disk2/mysql/ ~/var/db/mysql
$ ls ~/var/db/mysql/
```

Команды для работы с содержимым файлов Определение типа файла — file

Команда file представляет собой интерфейс к системе magic, который доступен пользователю в виде обычной команды.

Примеры:

```
$ file /usr/sbin/adduser
$ file /bin/sh
$ file /usr/share/man/man1/cat.1.gz
```

Просмотр файлов — more/less

Пейджеры (pager) more или less используются для просмотра больших объемов текстовой информации страницами. Они позволяют осуществлять навигацию, поиск по тексту и некоторые другие действия с помощью команд. Получить помощь по этим командам всегда можно, нажав клавишу h. Выход из пейджера осуществляется с помощью команды q. Поиск информации осуществляется с помощью команд / (вперед) и ? (назад), после которых указывается шаблон для поиска. Для получения подробной справки обратитесь к соответствующей странице справочного руководства.

Примеры запуска:

```
$ more /etc/defaults/rc.conf
$ less /etc/defaults/rc.conf
```

Работа с архивами

UNIX обладает богатым арсеналом средств резервного копирования и восстановления данных: программы dump/restore, cpio, tar и пр. Для работы архивами наиболее широко используется программа tar. Несмотря на то, что в разных системах UNIX используются различные реализации этой программы, получающиеся в результате архивы являются кроссплатформенными, т. е. могут быть обработаны в разных ОС (в т.ч. и Windows). Ниже приведены примеры работы с программой tar:

Создать архив:

```
$ tar -c -v -f имяфайлаархива.tar каталогилифайл ...
```

Посмотреть содержимое архива:

```
$ tar -t -f имяфайлаархива.tar
```

Раскрыть архив целиком:

```
$ tar -x -v -f имяфайлаархива.tar
```

Раскрыть отдельные файлы:

```
$ tar -xf имяфайлаархива.tar 'etc/fstab' $ tar -xOf имяфайлаархива.tar 'etc/fstab' - вывести  
на экран(STDOUT) $ tar -xf имяфайлаархива.tar 'etc/' $ tar -xf имяфайлаархива.tar '**fstab**'
```

Дополнительные ключи:

- z использовать gzip сжатие
- j использовать bzip2 сжатие

Команды оценки использования дискового пространства

Статистика использования разделов — df

Для получения статистики использования разделов дисков (смонтированных файловых систем) используется команда `df`. Наиболее полезные опции здесь — `h` (human-readable, выводит числовые данные в виде, удобном для восприятия пользователем) и `t` (выводит информацию только о файловых системах указанного типа, не принимая во внимание остальные — например, виртуальные файловые системы).

Пример запуска программы:

```
$ df -h -t ufs
Filesystem Size Used Avail Capacity Mounted on
/dev/ad0s1a 434M 143M 256M 36% /
/dev/ad0s1e 403M 12K 371M 0% /tmp
/dev/ad0s1f 6.4G 427M 5.5G 7% /usr
/dev/ad0s1d 771M 304K 709M 0% /var
```

Размер каталога — du

Программа du (disk usage) позволяет получить статистику использования дискового пространства не по разделам целиком, а для конкретных указанных каталогов. Опция h здесь аналогична этой опции программы df, а опции s (summary) и d число (depth) позволяют указать необходимую степень подробности (глубину) выводимой информации. Опция s эквивалента опции d 0 (нулевая глубина погружения), причем эти опции нельзя указывать вместе.

Примеры использования:

```
$ du -s -h /usr/share/
$ du -d 1 /usr/share/
```

Команды поиска файлов

Метоположение программ — which и whereis

Для поиска программ (исполняемых файлов) в UNIX используется команда which, которая ищет указанные файлы в каталогах, перечисленных в переменной окружения PATH. Команда whereis аналогична по действию, но ищет также среди man страниц и в каталогах с исходными текстами программ.

Примеры:

```
$ which ls
/bin/ls
```

```
$ whereis ls
```

```
ls: /bin/ls /usr/share/man/man1/ls.1.gz
```

Поиск файлов по индексированной базе — locate

Поиск файлов по имени с помощью заранее созданной индексной базы данных используется программа `locate`. Для создания индексной базы используется программа `/usr/libexec/locate.updatedb`

Полный поиск файлов - find

Программа `find` обладает большими возможностями для указания атрибутов искомых объектов и позволяет осуществлять поиск с самыми разными параметрами. Подробную информацию можно получить на странице справки.

Примеры использования `find`:

Поиск по имени и по шаблону имени

```
$ find /usr/share -name index.html
```

```
$ find /usr/share -name '*.html'
```

Поиск файлов, которые модифицировались за последние 2 дня и вывод полной информации про них

```
$ find /var/log -ctime -2 -type f -ls
```

Поиск файлов более новых чем некоторый


```
$ touch -t 200901051230 /tmp/xtime  
$ find /etc/ -newer /tmp/xtime -type f
```

Пример выполнения команд над найденными файлами

```
# find /usr/ports/ -name '*.tbz' -exec mv {} /usr/ports/packages/All/ ;
```