

## Работа с файловой системой

**Файловая система** — это логическая структура каталогов и файлов. В отличие от Windows, где каждый логический диск хранит отдельное дерево каталогов, во всех UNIX-подобных системах это древовидная структура растет из одного корня: он начинается с корневого каталога (*/*), родительского по отношению ко всем остальным, физических файловых системах разного типа, находящиеся на разных разделах и даже на удаленных машинах, представляются как ветви этого дерева.

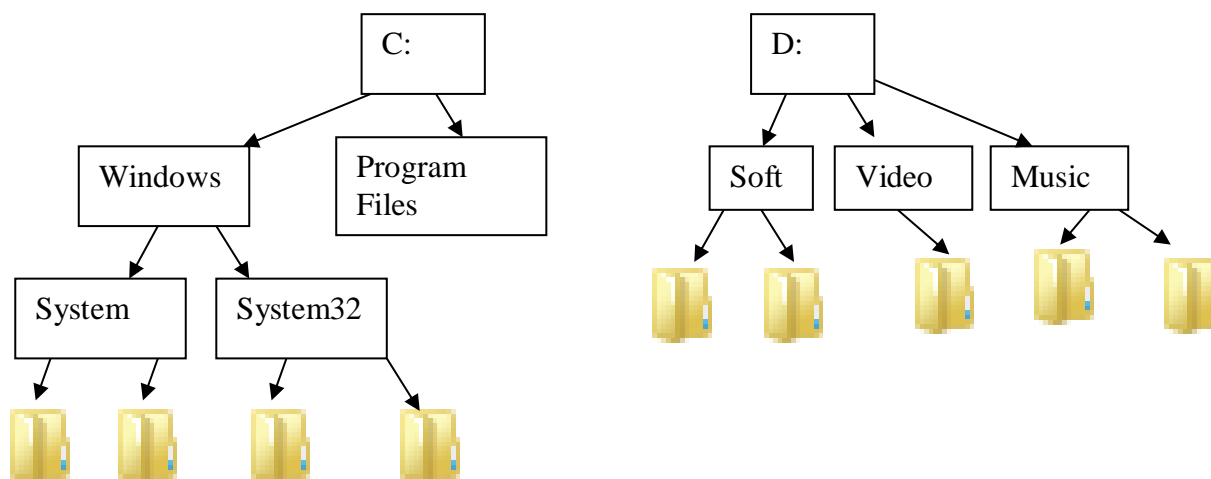


Рис. ФС\_01 Многоуровневая файловая структура Windows

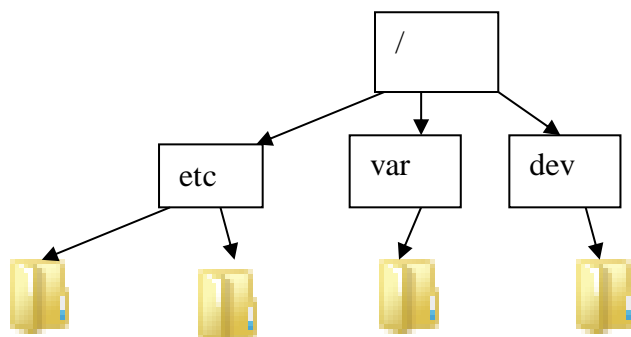


Рис. ФС\_02 Файловая структура Linux

### Имена файлов и каталогов

Имена файлов и каталогов могут иметь длину до 255 символов. Символ «*/*» (слэш) и символ с кодом 0 запрещены. Кроме того, ряд символов имеет специальное значение для командного интерпретатора, и их использование не рекомендуется. Это символы: *~!@#\$%\*{}""\';;> N пробел*

### Команды работы с файлами и каталогами

#### Текущий каталог

**Текущий каталог** — это каталог, от которого отсчитываются относительные пути. В каждый момент времени с каждой работающей программой, в том числе с командной оболочкой, связан единственный такой каталог. Узнать, какой каталог сейчас является текущим, можно с помощью команды **pwd** без аргументов. Команда **pwd** (**p**rint **w**orking **d**irectory)

возвращает полный путь текущего каталога командной оболочки — естественно, именно той командной оболочки, при помощи которой была выполнена команда `pwd`

### *Просмотр содержимого каталога*

Команда **ls**

Справка по работе команды: **ls --help | more**

В дальнейших примерах работать будем с каталогом **/etc**

#### **Пример ФС 1:**

Отобразить содержимое каталога **/etc**, включая скрытые файлы:

```
cd /etc
```

```
ls -al
```

**Ключи :**

**-a** - показать все файлы (включая скрытые)

**-l** - использовать длинный формат

#### **Пример ФС 2:**

Отобразить содержимое каталога **/etc**, включая скрытые файлы, с сортировкой по размеру:

```
cd /etc
```

```
ls -alS
```

**Ключи :**

**-S** - отсортировать по размеру

#### **Пример ФС 3:**

Отобразить содержимое каталога **/etc**, в обратном порядке:

```
ls -r
```

#### **Пример ФС 4:**

Отобразить содержимое каталога **/etc**, включая скрытые файлы, с сортировкой по размеру:

```
cd /etc
```

```
ls -alS
```

**Ключи :**

**-S** - отсортировать по размеру

### *Создание каталога*

Команда **mkdir**

#### **Пример ФС 5:**

Перейти в домашний каталог и создать там каталог **temp** :

```
cd ~
```

```
mkdir temp
```

```
ls -al
```

```
[root@localhost ~]# ls -al
total 36
dr-xr-x---.  4 root root    187 Jun  4 09:46 .
dr-xr-xr-x. 17 root root    224 Feb 26 06:33 ..
-rw-----.  1 root root   1261 Nov  3 2017 anaconda-ks.cfg
-rw-----.  1 root root    867 Mar  9 02:10 .bash_history
-rw-r--r--.  1 root root    18 Dec 28 2013 .bash_logout
-rw-r--r--.  1 root root   176 Dec 28 2013 .bash_profile
-rw-r--r--.  1 root root   176 Dec 28 2013 .bashrc
-rw-r--r--.  1 root root   100 Dec 28 2013 .cshrc
-rw-----.  1 root root    40 Jun  4 09:33 .lessht
drwxr-----. 3 root root    19 Mar  9 01:43 .pki
-rw-----.  1 root radiusd 1024 Mar  9 01:46 .rnd
-rw-r--r--.  1 root root   129 Dec 28 2013 .tcshrc
drwxr-xr-x.  2 root root     6 Jun  4 09:46 temp
```

### Создание пустого файла (touch)

**touch** - установка времени последнего изменения [файла](#) или доступа в текущее время. Также используется для создания пустых файлов.

!!! Самостоятельно ознакомиться с синтаксисом команды (**man touch**)

### Пример ФС 6:

Создать в каталоге **temp** файл **linux.txt**:

```
cd temp
touch linux.txt
ls -al
```

```
[root@localhost ~]# cd temp
[root@localhost temp]# touch linux.txt
[root@localhost temp]# ls -al
total 0
drwxr-xr-x. 2 root root 23 Jun  4 09:50 .
dr-xr-x---. 4 root root 187 Jun  4 09:46 ..
-rw-r--r--. 1 root root 0 Jun  4 09:50 linux.txt
[root@localhost temp]#
```

### Пример ФС 7:

Записать в файл **linux.txt** свою фамилию:

```
cd temp
echo Novikov > linux.txt
```

“>” называется символом перенаправления ввода. Другими словами команда **echo** направляет текст (Novikov) не на экран (консоль), а в файл. При этом содержимое файла переписывается.

Более подробно ознакомиться с возможностями команды **echo** необходимо по ссылке <https://osst.ru/komanda-echo-v-linux> , а также с помощью **man echo**

### Пример ФС 8:

!!! Самостоятельно изучить назначение и синтаксис команды **cat** (с использованием мануала **man** и сведений и интернета)

Вывести содержимое файла **linux.txt**:

**cat linux.txt**

```
[root@localhost temp]# echo Novikov > linux.txt
[root@localhost temp]# cat linux.txt
Novikov
[root@localhost temp]# _
```

### Пример ФС 9:

Записать в файл **linux.txt** свое имя:

**echo Dmitry > linux.txt**

**cat linux.txt**

```
[root@localhost temp]# echo Dmitry > linux.txt
[root@localhost temp]# cat linux.txt
Dmitry
```

Если необходимо добавить текст в конец существующего файла, то необходимо использовать символ ">>"

### Пример ФС 10:

Дописать в файл в файл **linux.txt** свое отчество:

**echo Nikolaevich >> linux.txt**

**cat linux.txt**

```
[root@localhost temp]# echo Nikolaevich>>linux.tx
[root@localhost temp]# cat
^C
[root@localhost temp]# cat linux.txt
Dmitry
Nikolaevich
```

Если необходимо отменить выполнение команды, то необходимо использовать сочетание клавиш **CTRL+C**

### *Копирование и перемещение файлов*

#### Перемещение (mv)

Для **перемещения** файлов и каталогов предназначена утилита **mv**

У **mv** два обязательных параметра: первый — перемещаемый файл или каталог, второй — файл или каталог назначения. Имена файлов и каталогов могут быть заданы в любом допустимом виде: при помощи полного или относительного пути. Кроме того, **mv** позволяет перемещать не только один файл или каталог, а сразу несколько.

### Пример ФС 11:

Создать в домашнем каталоге каталог **personal**, переместить туда файл **linux.txt** и переименовать его в файл **family.txt**:

Переходим в домашний каталог:

```
cd ~
```

Создаем каталог **personal**

```
mkdir personal
```

Перемещаем файл **linux.txt** из каталога **temp** в **personal**

```
mv temp/linux.txt personal
```

Смотрим содержимое каталога

```
ls -al personal
```

```
[root@localhost temp]# cd ~
[root@localhost ~]# mkdir personal
[root@localhost ~]# mv temp/linux.txt personal
[root@localhost ~]# ls -al personal
total 4
drwxr-xr-x. 2 root root 23 Jun 5 04:37 .
dr-xr-x---. 5 root root 203 Jun 5 04:37 ..
-rw-r--r--. 1 root root 19 Jun 5 04:11 linux.txt
[root@localhost ~]#
```

Переименовываем файл

```
mv personal/linux.txt personal/family.txt
```

Смотрим содержимое каталога

```
ls -al personal
```

```
[root@localhost ~]# mv personal/linux.txt personal/family.txt
[root@localhost ~]# ls -al personal/
total 4
drwxr-xr-x. 2 root root 24 Jun 5 05:08 .
dr-xr-x---. 5 root root 203 Jun 5 04:37 ..
-rw-r--r--. 1 root root 19 Jun 5 04:11 family.txt
[root@localhost ~]#
```

## Копирование (cp)

В Linux Linux для этого предназначена утилита **cp**. Утилита **cp** требует присутствия двух обязательных параметров: первый — копируемый файл или каталог, второй — файл или каталог назначения. Как обычно, в именах файлов и каталогов можно использовать полные и относительные пути. Существует несколько вариантов комбинации файлов и каталогов в параметрах **cp** — о них можно прочесть в руководстве (**man cp**):

### Пример ФС 12:

Создать в каталоге **personal** копию файла **family.txt** присвоив ей имя **text**:

Переходим в каталог **personal**:

```
cd ~/personal
```

Создаем копию файла. Ускорить ввод можно нажав клавишу **Tab**, при этом если в каталоге имеются имена, начинающиеся с введенных символов, то они будут показаны:

```
cp family.txt text
```

Смотрим содержимое

**ls**

```
[root@localhost ~]# cd ~/personal
[root@localhost personal]# cp family.txt text
[root@localhost personal]# ls
family.txt  text
[root@localhost personal]#
```

Проверяем файл на наличие в нем текста

**cat text**

```
[root@localhost personal]# cat text
Dmitry
Nikolaevich
```

### Пример ФС 13:

Создать копию каталога **personal** присвоив имя **personal\_backup**:

Переходим в родительский каталог:

**cd ../**

Попытаемся создать копию каталога таким же способом, как копировали файл:

**cp personal personal\_backup**

Выводится ошибка

```
[root@localhost ~]# cp personal personal_backup
cp: omitting directory 'personal'
```

Выведем справку (помощь) по работе с данной программой. Команда **more** указывает осуществлять вывод постранично. Для чтения следующей страницы нажать «пробел»

**cp --help | more**

```
-R, -r, --recursive      copy directories recursively
--reflink[=WHEN]         control clone/CoW copies. See below
--remove-destination     remove each existing destination file before
```

Среди прочего видим однобуквенный ключ **-r**, который указывает на возможность рекурсивного копирования содержимого каталога (включая его подкаталоги)

**cp -r personal personal\_backup**

Выводим содержимое текущего каталога

**ls -al**

```
[root@localhost ~]# ls -al
total 36
dr-xr-x---.  6 root root    226 Jun  5 06:01 .
dr-xr-xr-x. 17 root root    224 Feb 26 06:33 ..
-rw-----.  1 root root  1261 Nov  3  2017 anaconda-ks.cfg
-rw-----.  1 root root   1523 Jun  4 10:02 .bash_history
-rw-r--r--.  1 root root    18 Dec 28  2013 .bash_logout
-rw-r--r--.  1 root root   176 Dec 28  2013 .bash_profile
-rw-r--r--.  1 root root   176 Dec 28  2013 .bashrc
-rw-r--r--.  1 root root   100 Dec 28  2013 .cshrc
-rw-----.  1 root root    40 Jun  4 09:33 .lessht
drwxr-xr-x.  2 root root    52 Jun  5 05:38 personal
drwxr-xr-x.  2 root root    52 Jun  5 06:01 personal_backup
```

## Удаление файлов (rm)

Команда **rm** используется для удаления файлов и каталогов. Важно помнить, что удаленные файлы и каталоги не попадают в «мусорную корзину», как в операционных системах, ориентированных на десктоп. Когда файл удаляется с помощью команды **rm**, он навсегда исчезает.

Без каких-либо опций команда `rm` обычно используется для удаления обычных файлов:

```
cp /var/log/messages ~/messages
ls ~
rm messages
ls ~
```

Чтобы избежать случайного удаления файлов при использовании символов-заглушек, используйте параметр `-i`. Эта опция заставляет команду `rm` «интерактивно» подтверждать каждый файл, который вы удаляете:

```
rm -i a*
```

Команда `rm` будет игнорировать каталоги, которые просят удалить; чтобы удалить каталог, используйте опции `-r` или `-R`. Просто будьте осторожны, так как это удалит все файлы и все подкаталоги:

```
rm -r test
```

Для принудительного удаления файлов и каталогов необходимо использовать ключ `-f`:

```
mkdir ~/log
cp -R /var/log/*.log ~/log
ls ~/log
rm -fR ~/log
ls ~/log
```