

Работа с файлами

Из данной главы вы узнаете о том, как определять тип файлов, создавать, удалять, копировать и перемещать файлы с помощью таких утилит, как **file**, **touch**, **rm**, **cp**, **mv** и **rename**.

Все имена файлов регистрозависимы

Имена файлов в Linux (или в любой системе Unix) являются **регистрозависимыми**. Это означает, что имена **FILE1** и **file1** не являются равноценными, а путь к файлу **/etc/hosts** отличен от пути к файлу **/etc/Hosts** (последнего файла не должно существовать в файловой системе типичного компьютера, работающего под управлением Linux).

В данном примере продемонстрировано различие между двумя файлами, причем имя первого файла начинается с заглавной буквы **W**, а второго - с прописной буквы **w**.

```
paul@laika:~/Linux$ ls
winter.txt  Winter.txt
paul@laika:~/Linux$ cat winter.txt
Холодно.
paul@laika:~/Linux$ cat Winter.txt
Очень холодно!
```

Все является файлом

Любая **директория** на самом деле является **файлом** (с регистрозависмым именем!), хотя этот **файл** и имеет специальный тип. Каждый терминал (например, **/dev/pts/4**), любой жесткий диск или раздел на нем (например, **/dev/sdb1**) и любой процесс представлены где-либо в рамках **файловой системы** с помощью **файла**. После изучения данной главы у вас не останется сомнений в том, что в Linux все является файлом.

Утилита **file**

Утилита **file** предназначена для определения типа файла. В Linux для определения типов файлов не используются их расширения. Для инструментов с интерфейсом командной строки абсолютно безразлично, оканчивается ли имя файла на **.txt** или на **.pdf**. Исполняя обязанности системного администратора, вы должны использовать команду **file** для установления типа интересующего вас файла.

Ниже приведено несколько примеров использования данной утилиты при работе с типичной системой Linux.

```
paul@laika:~$ file pic33.png
pic33.png: PNG image data, 3840 x 1200, 8-bit/color RGBA,
non-interlaced
paul@laika:~$ file /etc/passwd
/etc/passwd: ASCII text
paul@laika:~$ file HelloWorld.c
HelloWorld.c: ASCII C program text
```

Утилита **file** использует файл со списком "магических последовательностей байт", содержащий шаблоны для распознавания типов данных. Файл со списком "магических последовательностей байт" расположен по пути **/usr/share/file/magic**. Используйте команду **man 5 magic** в случае необходимости получения дополнительной информации о нем.

Важно отметить, что для работы с такими специальными файлами, как файлы из директорий **/dev** и **/proc**, следует использовать команду **file -s**.

```
root@debian6~# file /dev/sda
/dev/sda: block special
root@debian6~# file -s /dev/sda
/dev/sda: x86 boot sector; partition 1: ID=0x83, active,
starthead...
root@debian6~# file /proc/cpuinfo
/proc/cpuinfo: empty
root@debian6~# file -s /proc/cpuinfo
/proc/cpuinfo: ASCII C++ program text
```

Утилита **touch**

Создание пустого файла

Один из простых способов создания пустого файла заключается в использовании утилиты **touch**. (Позднее в данной книге мы рассмотрим множество других способов создания файлов.)

Данный пример начинается с открытия пустой директории, в которой с помощью утилиты **touch** создаются два файла, после чего выводится список созданных файлов.

```
paul@debian7:~$ ls -l
```

```
итого 0
paul@debian7:~$ touch file42
paul@debian7:~$ touch file33
paul@debian7:~$ ls -l
итого 0
-rw-r--r-- 1 paul paul 0 окт 15 08:57 file33
-rw-r--r-- 1 paul paul 0 окт 15 08:56 file42
paul@debian7:~$
```

Команда **touch -t**

Утилита **touch** позволяет устанавливать набор значений свойств пустых файлов в процессе их создания. Удастся ли вам определить, какие значения свойств были установлены, ознакомившись с приведенным ниже примером? Если вы не смогли разобраться самостоятельно, обратитесь к странице руководства для утилиты **touch**.

```
paul@debian7:~$ touch -t 200505050000 SinkoDeMayo
paul@debian7:~$ touch -t 130207111630 BigBattle.txt
paul@debian7:~$ ls -l
total 0
-rw-r--r-- 1 paul paul 0 июл 11 1302 BigBattle.txt
-rw-r--r-- 1 paul paul 0 окт 15 08:57 file33
-rw-r--r-- 1 paul paul 0 окт 15 08:56 file42
-rw-r--r-- 1 paul paul 0 май 5 2005 SinkoDeMayo
paul@debian7:~$
```

rm

Удаление файлов навсегда

В том случае, если вам больше не нужен файл, следует использовать утилиту **rm** для его удаления. В отличие от некоторых графических пользовательских интерфейсов, интерфейс командной строки системы не предполагает использования таких хранилищ файлов, как **корзина**, предназначенных для последующего восстановления удаленных файлов. Если вы удалили файл с помощью утилиты **rm**, этот файл будет удален навсегда. Исходя из этого, следует проявлять осторожность при удалении файлов!

```
paul@debian7:~$ ls
BigBattle.txt file33 file42 SinkoDeMayo
```

```
paul@debian7:~$ rm BigBattle.txt
paul@debian7:~$ ls
file33  file42  SinkoDeMayo
paul@debian7:~$
```

Команда `rm -i`

Для того, чтобы застраховаться от случайного удаления файла, вы можете использовать команду `rm -i`.

```
paul@debian7:~$ ls
file33  file42  SinkoDeMayo
paul@debian7:~$ rm -i file33
rm: удалить пустой обычный файл "file33"? yes
paul@debian7:~$ rm -i SinkoDeMayo
rm: удалить пустой обычный файл "SinkoDeMayo"? n
paul@debian7:~$ ls
file42  SinkoDeMayo
paul@debian7:~$
```

Команда `rm -rf`

По умолчанию с помощью команды `rm -r` невозможно удалить непустые директории. Однако, утилита `rm` принимает параметры, которые позволяют вам удалять любую директорию. Команда `rm -rf` известна благодаря тому, что она позволяет удалять все элементы файловой системы (конечно же, в том случае, если вы имеете достаточные права для выполнения этой операции). В том случае, если вы вошли в систему, воспользовавшись учетной записью пользователя `root`, следует проявить особую осторожность при использовании команды `rm -rf` (параметр `f` расшифровывается как **force** (принудительное выполнение операции), а параметр `r` - как **recursive** (рекурсивный обход директорий)), так как в случае работы с учетной записью пользователя `root`, упомянутые ограничения прав не будут распространяться на вас. Вы сможете в буквальном смысле удалить все файлы вашей системы по неосторожности.

```
paul@debian7:~$ mkdir test
paul@debian7:~$ rm test
rm: невозможно удалить "test": Это каталог
paul@debian7:~$ rm -rf test
paul@debian7:~$ ls test
```

ls: невозможно получить доступ к test: Нет такого файла или каталога

paul@debian7:~\$

Утилита **cp**

Копирование отдельных файлов

Для копирования файла следует использовать утилиту **cp** с аргументами, представленными путями к исходному и целевому файлам.

paul@debian7:~\$ **ls**

file42 SinkoDeMayo

paul@debian7:~\$ **cp file42 file42.copy**

paul@debian7:~\$ **ls**

file42 file42.copy SinkoDeMayo

Копирование файлов в другую директорию

В том случае, если в качестве пути к целевому файлу используется путь к директории, исходные файлы будут скопированы в эту целевую директорию.

paul@debian7:~\$ **mkdir dir42**

paul@debian7:~\$ **cp SinkoDeMayo dir42**

paul@debian7:~\$ **ls dir42/**

SinkoDeMayo

Команда **cp -r**

Для копирования директорий целиком следует использовать команду **cp -r** (параметр **-r** позволяет осуществлять рекурсивное копирование всех файлов из всех поддиректорий).

paul@debian7:~\$ **ls**

dir42 file42 file42.copy SinkoDeMayo

paul@debian7:~\$ **cp -r dir42/ dir33**

paul@debian7:~\$ **ls**

dir33 dir42 file42 file42.copy SinkoDeMayo

paul@debian7:~\$ **ls dir33/**

SinkoDeMayo

Копирование множества файлов в директорию

Вы также можете использовать утилиту **cp** для копирования множества файлов в одну директорию. В этом случае последний аргумент (аргумент, указывающий на цель) должен быть представлен путем к директории.

```
paul@debian7:~$ cp file42 file42.copy SinkoDeMayo dir42/
paul@debian7:~$ ls dir42/
file42  file42.copy  SinkoDeMayo
```

Команда **cp -i**

Для предотвращения перезаписи существующих файлов в ходе использования утилиты **cp** следует использовать параметр **-i** (для активации интерактивного режима копирования).

```
paul@debian7:~$ cp SinkoDeMayo file42
paul@debian7:~$ cp SinkoDeMayo file42
paul@debian7:~$ cp -i SinkoDeMayo file42
cp: переписать "file42"? n
paul@debian7:~$
```

Утилита **mv**

Переименование файлов с помощью утилиты **mv**

Утилита **mv** используется для переименования файлов или для перемещения файлов в другие директории.

```
paul@debian7:~$ ls
dir33  dir42  file42  file42.copy  SinkoDeMayo
paul@debian7:~$ mv file42 file33
paul@debian7:~$ ls
dir33  dir42  file33  file42.copy  SinkoDeMayo
paul@debian7:~$
```

В том случае, если вам необходимо переименовать один файл, утилита **mv** является предпочтительным инструментом.

Переименование директорий с помощью утилиты **mv**

Эта же утилита **mv** может быть использована и для переименования директорий.

```
paul@debian7:~$ ls -l
```

итого 8

```

drwxr-xr-x 2 paul paul 4096 окт 15 09:36 dir33
drwxr-xr-x 2 paul paul 4096 окт 15 09:36 dir42
-rw-r--r-- 1 paul paul    0 окт 15 09:38 file33
-rw-r--r-- 1 paul paul    0 окт 15 09:16 file42.copy
-rw-r--r-- 1 paul paul    0 май  5  2005 SinkoDeMayo
paul@debian7:~$ mv dir33 backup
paul@debian7:~$ ls -l
итого 8
drwxr-xr-x 2 paul paul 4096 окт 15 09:36 backup
drwxr-xr-x 2 paul paul 4096 окт 15 09:36 dir42
-rw-r--r-- 1 paul paul    0 окт 15 09:38 file33
-rw-r--r-- 1 paul paul    0 окт 15 09:16 file42.copy
-rw-r--r-- 1 paul paul    0 май  5  2005 SinkoDeMayo
paul@debian7:~$

```

Команда **mv -i**

Утилита **mv** поддерживает параметр **-i** по аналогии с утилитами **cp** и **rm**.

В примере ниже показано, как при использовании команды **mv -i** запрашивается подтверждение перезаписи существующего файла.

```

paul@debian7:~$ mv -i file33 SinkoDeMayo
mv: переписать "SinkoDeMayo"? no
paul@debian7:~$

```

Утилита **rename**

Об утилите **rename**

Утилита **rename** является одним из редких случаев, когда в рамках книги "Фундаментальные основы Linux" приходится делать разделение между дистрибутивами Linux. Практически каждая из описанных в данной книге утилит работает практически на каждом компьютере под управлением Linux. Но реализации утилиты **rename** в различных дистрибутивах отличаются.

Следует пытаться использовать утилиту **mv** всегда, когда вам необходимо переименовать несколько файлов.

Утилита **rename** в дистрибутиве Debian/Ubuntu

Утилита **rename** в дистрибутиве Debian имеет сложный синтаксис (используются регулярные выражения) для одновременного переименования множества файлов.

Ниже приведен пример использования утилиты **rename**, с помощью которой у всех файлов с расширением **.txt** производится замена расширения на **.png**.

```
paul@debian7:~/test42$ ls
abc.txt  file33.txt  file42.txt
paul@debian7:~/test42$ rename 's/\.txt/\.png/' *.txt
paul@debian7:~/test42$ ls
abc.png  file33.png  file42.png
```

Во втором примере производится замена всех (первых) вхождений слов "**file**" на "**document**" во всех файлах с расширением **.png**.

```
paul@debian7:~/test42$ ls
abc.png  file33.png  file42.png
paul@debian7:~/test42$ rename 's/file/document/' *.png
paul@debian7:~/test42$ ls
abc.png  document33.png  document42.png
paul@debian7:~/test42$
```

Утилита **rename** в дистрибутиве CentOS/RHEL/Fedora

В Red Hat Enterprise Linux синтаксис утилиты **rename** немного отличается. В первом примере, представленном ниже, осуществляется переименование всех файлов с расширением **.conf** (*.conf) путем замены всех вхождений **.conf** на **.backup**.

```
[paul@centos7 ~]$ touch one.conf two.conf three.conf
[paul@centos7 ~]$ rename .conf .backup *.conf
[paul@centos7 ~]$ ls
one.backup  three.backup  two.backup
[paul@centos7 ~]$
```

Во втором примере производится переименование всех файлов (*) путем замены всех вхождений **one** на **ONE**.

```
[paul@centos7 ~]$ ls
one.backup  three.backup  two.backup
[paul@centos7 ~]$ rename one ONE *
```



```
[paul@centos7 ~]$ ls
ONE.backup  three.backup  two.backup
[paul@centos7 ~]$
```