

Работа с директориями

В данной главе приводится краткий обзор большинства стандартных команд, используемых при работе с директориями: **pwd**, **cd**, **ls**, **mkdir** и **rmdir**. Эти команды доступны при работе с любой системой Linux (или Unix).

Кроме того, в данной главе обсуждаются понятия **абсолютных** и **относительных путей**, а также описывается методика использования механизма **завершения путей** командной оболочки **bash**.

Команда **pwd**

С помощью команды **pwd** (расшифровывается как Print Working Directory - вывести информацию о рабочей директории) может быть получена информация о вашем текущем местонахождении в рамках файловой системы. Попробуйте выполнить эту команду: получите доступ к интерфейсу командной строки системы (воспользовавшись одним из приложений со следующими названиями: terminal, console или xterm) и введите команду **pwd**. Командная оболочка выведет путь к вашей **текущей директории**.

```
paul@debian8:~$ pwd  
/home/paul
```

Команда **cd**

Вы можете изменить вашу текущую директорию с помощью команды **cd** (расшифровывается как Change Directory - изменить директорию).

```
paul@debian8$ cd /etc
paul@debian8$ pwd
/etc
paul@debian8$ cd /bin
paul@debian8$ pwd
/bin
paul@debian8$ cd /home/paul/
paul@debian8$ pwd
/home/paul
```

Команда **cd ~**

Команда **cd** также может использоваться для быстрого перехода назад в вашу домашнюю директорию. Простое исполнение команды **cd** без задания пути к целевой директории приведет к перемещению в домашнюю директорию. Исполнение команды **cd ~** приведет к аналогичному эффекту.

```
paul@debian8$ cd /etc
paul@debian8$ pwd
/etc
paul@debian8$ cd
paul@debian8$ pwd
/home/paul
paul@debian8$ cd ~
paul@debian8$ pwd
/home/paul
```

Команда **cd ..**

Для перехода в **родительскую директорию** (ту директорию, которая находится над вашей текущей директорией в дереве директорий) следует использовать команду **cd ..**

```
paul@debian8$ pwd
/usr/share/games
```

```
paul@debian8$ cd ..  
paul@debian8$ pwd  
/usr/share
```

*Для того, чтобы остаться в текущей директории, просто введите команду **cd .** ;-)* Позднее мы все же познакомимся с практическим примером использования символа **.**, представляющего текущую директорию.

Команда **cd** -

Другой полезный вариант использования команды **cd** заключается в выполнении простой команды **cd** - для перехода в предыдущую директорию.

```
paul@debian8$ pwd  
/home/paul  
paul@debian8$ cd /etc  
paul@debian8$ pwd  
/etc  
paul@debian8$ cd -  
/home/paul  
paul@debian8$ cd -  
/etc
```

Абсолютные и относительные пути

Вы должны иметь представление об **абсолютных и относительных путях** в рамках дерева директорий файловой системы. Если вы вводите путь, начинающийся с символа **слэша (/)**, подразумевается, что путь будет указан относительно корневой директории файловой системы. Если же вы не начинаете ввод пути с символа **слэша**, подразумевается, что точкой отсчета будет текущая директория.

В примере ниже показано, что текущей директорией является директория **/home/paul**. Для перехода из этой директории в директорию **/home** вам придется ввести команду **cd /home** вместо команды **cd home**.

```
paul@debian8$ pwd  
/home/paul  
paul@debian8$ cd home  
bash: cd: home: Нет такого файла или каталога
```

```
paul@debian8$ cd /home
```

```
paul@debian8$ pwd
```

```
/home
```

При нахождении в директории **/home** вам придется ввести команду **cd paul** вместо команды **cd /paul** для перехода в поддиректорию **paul** текущей директории **/home**.

```
paul@debian8$ pwd
```

```
/home
```

```
paul@debian8$ cd /paul
```

```
bash: cd: /paul: Нет такого файла или каталога
```

```
paul@debian8$ cd paul
```

```
paul@debian8$ pwd
```

```
/home/paul
```

В том же случае, если вашей текущей директорией является **корневая директория /**, то и команда **cd /home**, и команда **cd home** позволят вам переместиться в директорию **/home**.

```
paul@debian8$ pwd
```

```
/
```

```
paul@debian8$ cd home
```

```
paul@debian8$ pwd
```

```
/home
```

```
paul@debian8$ cd /
```

```
paul@debian8$ cd /home
```

```
paul@debian8$ pwd
```

```
/home
```

Этот пример был последним примером, в котором используются команды **pwd**. С этого момента информация о текущей директории будет всегда выводиться в рамках приглашения командной оболочки. Далее в данной книге будут даны пояснения относительно того, как следует использовать переменную командной оболочки **\$PS1** для вывода описанной информации.

Завершение путей

Клавиша Tab может помочь избежать ошибок при вводе путей. Нажатие **клавиши Tab** после ввода части команды **cd /et** приведет к завершению команды до вида **cd /etc/**. При этом нажатие **кла-**

виши Tab после ввода команды **cd /Et** не приведет ни к чему, так как была допущена ошибка при вводе части **пути** (использована буква **E** в верхнем регистре).

Вам придется нажимать меньше клавиш в случае использования **клавиши Tab**, к тому же, при использовании этой клавиши вы можете быть уверены в том, что введенный **путь** является корректным!

Утилита **ls**

Вы можете вывести список содержимого директории с помощью утилиты **ls**.

```
paul@debian8:~$ ls
```

```
allfiles.txt  dmesg.txt  services  stuff  summer.txt
```

```
paul@debian8:~$
```

Команда **ls -a**

Часто используемым параметром утилиты **ls** является параметр **-a**, который предназначен для вывода информации обо всех файлах. Под выводом информации обо всех файлах подразумевается вывод информации в том числе и о **скрытых файлах**. В том случае, если имя файла в рамках файловой системы Linux начинается с символа точки, он считается **скрытым файлом** и не включается в обычные списки содержимого директорий.

```
paul@debian8:~$ ls
```

```
allfiles.txt  dmesg.txt  services  stuff  summer.txt
```

```
paul@debian8:~$ ls -a
```

```
.  allfiles.txt  .bash_profile  dmesg.txt  .lessht  
stuff
```

```
..  .bash_history  .bashrc  services  .ssh  
summer.txt
```

```
paul@debian8:~$
```

Команда **ls -l**

Вам придется многократно использовать параметры утилиты **ls** для вывода информации о содержимом директории в различных форматах или для вывода информации о различных файлах из директории. Команда **ls** без параметров позволяет получить список

файлов, расположенных в директории. Команда **ls -l** (в качестве параметра использована строчная буква L, а не число 1) позволяет получить более подробный список файлов.

```
paul@debian8:~$ ls -l
```

```
итого 17296
```

```
-rw-r--r-- 1 paul paul 17584442 сен 17 00:03 allfiles.txt
-rw-r--r-- 1 paul paul    96650 сен 17 00:03 dmesg.txt
-rw-r--r-- 1 paul paul    19558 сен 17 00:04 services
drwxr-xr-x 2 paul paul    4096 сен 17 00:04 stuff
-rw-r--r-- 1 paul paul      0 сен 17 00:04 summer.txt
```

Команда **ls -lh**

Другим периодически используемым параметром утилиты **ls** является параметр **-h**. Он позволяет выводить числовые значения (соответствующие размерам файлов) в формате, лучше читаемом человеком. Также в примере ниже показаны варианты передачи параметров утилите **ls**. Позднее в данной книге будут даны подробные пояснения относительно выводимых данных.

Обратите внимание на то, что мы используем строчную букву L, а не число 1 в качестве параметра утилиты в данном примере.

```
paul@debian8:~$ ls -l -h
```

```
итого 17M
```

```
-rw-r--r-- 1 paul paul 17M сен 17 00:03 allfiles.txt
-rw-r--r-- 1 paul paul 95K сен 17 00:03 dmesg.txt
-rw-r--r-- 1 paul paul 20K сен 17 00:04 services
drwxr-xr-x 2 paul paul 4.0K сен 17 00:04 stuff
-rw-r--r-- 1 paul paul 0 сен 17 00:04 summer.txt
```

```
paul@debian8:~$ ls -lh
```

```
итого 17M
```

```
-rw-r--r-- 1 paul paul 17M сен 17 00:03 allfiles.txt
-rw-r--r-- 1 paul paul 95K сен 17 00:03 dmesg.txt
-rw-r--r-- 1 paul paul 20K сен 17 00:04 services
drwxr-xr-x 2 paul paul 4.0K сен 17 00:04 stuff
-rw-r--r-- 1 paul paul 0 сен 17 00:04 summer.txt
```

```
paul@debian8:~$ ls -hl
```

```
итого 17M
-rw-r--r-- 1 paul paul 17M сен 17 00:03 allfiles.txt
-rw-r--r-- 1 paul paul 95K сен 17 00:03 dmesg.txt
-rw-r--r-- 1 paul paul 20K сен 17 00:04 services
drwxr-xr-x 2 paul paul 4.0K сен 17 00:04 stuff
-rw-r--r-- 1 paul paul 0 сен 17 00:04 summer.txt
paul@debian8:~$ ls -h -l
```

```
итого 17M
-rw-r--r-- 1 paul paul 17M сен 17 00:03 allfiles.txt
-rw-r--r-- 1 paul paul 95K сен 17 00:03 dmesg.txt
-rw-r--r-- 1 paul paul 20K сен 17 00:04 services
drwxr-xr-x 2 paul paul 4.0K сен 17 00:04 stuff
-rw-r--r-- 1 paul paul 0 Sep 17 00:04 summer.txt
paul@debian8:~$
```

Утилита **mkdir**

Обход дерева директорий файловой системы Unix является интересным занятием, но еще больший интерес представляет создание ваших собственных директорий с помощью утилиты **mkdir**. Вам придется передавать как минимум один параметр утилите **mkdir**, а именно, имя новой директории, которая должна быть создана. При этом следует серьезно задумываться перед использованием начального символа / в именах директорий.

```
paul@debian8:~$ mkdir mydir
paul@debian8:~$ cd mydir
paul@debian8:~/mydir$ ls -al
итого 8
drwxr-xr-x 2 paul paul 4096 сен 17 00:07 .
drwxr-xr-x 48 paul paul 4096 сен 17 00:07 ..
paul@debian8:~/mydir$ mkdir stuff
paul@debian8:~/mydir$ mkdir otherstuff
paul@debian8:~/mydir$ ls -l
итого 8
drwxr-xr-x 2 paul paul 4096 сен 17 00:08 otherstuff
drwxr-xr-x 2 paul paul 4096 сен 17 00:08 stuff
paul@debian8:~/mydir$
```

Команда `mkdir -p`

Исполнение следующей команды закончится неудачей, так как **родительской директории** для директории **threedirsdeep** не существует.

```
paul@debian8:~$ mkdir mydir2/mysubdir2/threedirsdeep
mkdir: невозможно создать каталог
"mydir2/mysubdir2/threedirsdeep": Нет такого файла или каталога
```

В случае использования параметра **-p** утилиты **mkdir** при необходимости будут создаваться **родительские директории**.

```
paul@debian8:~$ mkdir -p mydir2/mysubdir2/threedirsdeep
paul@debian8:~$ cd mydir2
paul@debian8:~/mydir2$ ls -l
итого 4
drwxr-xr-x 3 paul paul 4096 сен 17 00:11 mysubdir2
paul@debian8:~/mydir2$ cd mysubdir2
paul@debian8:~/mydir2/mysubdir2$ ls -l
итого 4
drwxr-xr-x 2 paul paul 4096 сен 17 00:11 threedirsdeep
paul@debian8:~/mydir2/mysubdir2$ cd threedirsdeep/
paul@debian8:~/mydir2/mysubdir2/threedirsdeep$ pwd
/home/paul/mydir2/mysubdir2/threedirsdeep
```

Утилита `rmdir`

В том случае, если директория пуста, вы можете использовать утилиту `rmdir` для удаления этой директории.

```
paul@debian8:~/mydir$ ls -l
итого 8
drwxr-xr-x 2 paul paul 4096 сен 17 00:08 otherstuff
drwxr-xr-x 2 paul paul 4096 сен 17 00:08 stuff
paul@debian8:~/mydir$ rmdir otherstuff
paul@debian8:~/mydir$ cd ..
paul@debian8:~$ rmdir mydir
rmdir: не удалось удалить "mydir": Каталог не пуст
paul@debian8:~$ rmdir mydir/stuff
```



```
paul@debian8:~$ rmdir mydir
```

```
paul@debian8:~$
```

Команда `rmdir -p`

И по аналогии с параметром **`mkdir -p`**, вы также можете использовать утилиту **`rmdir`** для рекурсивного удаления директорий.

```
paul@debian8:~$ mkdir -p test42/subdir
```

```
paul@debian8:~$ rmdir -p test42/subdir
```

```
paul@debian8:~$
```