

Работа с содержимым файлов

В данной главе мы будем работать с содержимым **текстовых файлов**, используя утилиты **head**, **tail**, **cat**, **tac**, **more**, **less** и **strings**.

Также мы познакомимся с возможностями таких инструментов с интерфейсом командной строки, как **cat**.

Утилита head

Вы можете использовать утилиту **head** для вывода первых десяти строк файла.

```
paul@laika:~$ head /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
paul@laika:~$
```

Утилита **head** также может использоваться для вывода первых **n** строк файла.

```
paul@laika:~$ head -4 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
```

Кроме того, утилита **head** может использоваться для вывода первых **n** байт файла.

```
paul@laika:~$ head -c4 /etc/passwd
rootpaul@laika:~$
```

Утилита **tail**

По аналогии с утилитой **head**, утилита **tail** может использоваться для вывода последних десяти строк файла.

```
paul@laika:~$ tail /etc/services
```

```
vboxd          20012/udp
binkp          24554/tcp      # binkp fidonet protocol
asp            27374/tcp      # Address Search Protocol
asp            27374/udp
csync2         30865/tcp      # cluster synchronization
tool
dircproxy      57000/tcp      # Detachable IRC Proxy
tfido          60177/tcp      # fidonet EMSI over
telnet
fido           60179/tcp      # fidonet EMSI over TCP
```

```
# Local services
```

```
paul@laika:~$
```

Вы можете передать утилите **tail** число, соответствующее количеству строк, которое вы хотите увидеть в выводе.

```
$ tail -3 count.txt
```

```
fido           60179/tcp      # fidonet EMSI over TCP
```

```
# Local services
```

```
paul@laika:~$
```

Утилита **tail** поддерживает и другие полезные параметры, причем некоторые из них мы будем использовать в рамках данного курса.

Утилита **cat**

Утилита **cat** является одним из наиболее универсальных инструментов операционной системы. Все, что она делает - это копирует данные из **стандартного потока ввода** в **стандартный поток вывода**. Данная утилита в комбинации с командной оболочкой позволяет реализовать мощные и разнообразные механизмы обработки данных. Воспользуемся примерами для иллюстрации этих механизмов. Первый пример является достаточно простым и вы можете использовать рассмотренную в нем команду в повседневной работе

для вывода содержимого файла на экран. В том случае, если данные из файла не умещаются на экране, будет осуществлена прокрутка до конца файла.

```
paul@laika:~$ cat /etc/resolv.conf
nameserver 194.7.1.4
paul@laika:~$
```

Объединение файлов

Имя утилиты **cat** является сокращенной формой слова **concatenate** (объединить). Одним из наиболее простых сценариев использования данной утилиты является объединение файлов в рамках файла большего объема (или полного файла).

```
paul@laika:~$ echo один >part1
paul@laika:~$ echo два >part2
paul@laika:~$ echo три >part3
paul@laika:~$ cat part1
один
paul@laika:~$ cat part2
два
paul@laika:~$ cat part3
три
paul@laika:~$ cat part1 part2 part3
один
два
три
paul@laika:~$ cat part1 part2 part3 >all
paul@laika:~$ cat all
один
два
три
paul@laika:~$
```

Создание файлов

Также вы можете использовать утилиту **cat** для создания текстовых файлов. Введите команду **cat > winter.txt**, приведенную в примере ниже. После этого введите одну строку или несколько строк текста, завершая ввод каждой из строк нажатием клавиши **Enter**. По-

сле ввода последней строки текста нажмите и удерживайте клавишу **Ctrl** и одновременно с этим кратковременно нажмите клавишу **d**.

```
paul@laika:~/test$ cat > winter.txt
```

Сегодня очень холодно!

```
paul@laika:~/test$ cat winter.txt
```

Сегодня очень холодно!

```
paul@laika:~/test$
```

Комбинация клавиш **Ctrl+d** позволяет передать исполняющемуся процессу символ окончания файла (**End of File - EOF**), что приведет к завершению исполнения процесса **cat**.

Специальный маркер окончания файла

Вы можете установить специальный маркер окончания файла для утилиты **cat** с помощью параметра **<<** таким образом, как показано в примере. Данная конструкция называется локальной директивой (**here directive**) и позволяет завершать работу процесса **cat**.

```
paul@laika:~/test$ cat > hot.txt <<stop
```

> Сегодня жарко!

> Да, это лето.

> stop

```
paul@laika:~/test$ cat hot.txt
```

Сегодня жарко!

Да, это лето.

```
paul@laika:~/test$
```

Копирование файлов

При рассмотрении третьего примера вы можете обнаружить, что утилита **cat** также может использоваться для копирования файлов. Подробные пояснения относительно происходящих в этом случае процессов будут даны в главе, посвященной командной оболочке **bash**.

```
paul@laika:~/test$ cat winter.txt
```

Сегодня очень холодно!

```
paul@laika:~/test$ cat winter.txt > cold.txt
```

```
paul@laika:~/test$ cat cold.txt
```

Сегодня очень холодно!

```
paul@laika:~/test$
```

Утилита **tac**

Предназначение утилиты **tac** (в отличие от утилиты **cat**) может быть проиллюстрировано с помощью единственного примера.

```
paul@laika:~/test$ cat count
```

один

два

три

четыре

```
paul@laika:~/test$ tac count
```

четыре

три

два

один

```
paul@laika:~/test$
```

Утилиты **more** и **less**

Утилита **more** может оказаться полезной в случае возникновения необходимости вывода содержимого файлов, которое не уместится на экране. Утилита **more** позволяет ознакомиться с содержимым файла, разделенным на страницы. После открытия файла с использованием данной утилиты следует использовать клавишу "пробел" для перехода к следующей странице или клавишу **q** для выхода из режима просмотра содержимого файла. Некоторые пользователи предпочитают использовать утилиту **less** вместо утилиты **more**.

Утилита **strings**

С помощью утилиты **strings** вы можете осуществить вывод читаемых человеком **ascii**-строк, которые обнаруживаются в бинарных файлах. В приведенном ниже примере выясняется путь к бинарному файлу **ls**, после чего осуществляется вывод читаемых пользователем строк из этого бинарного файла (вывод сокращен).

```
paul@laika:~$ which ls
```

/bin/ls

```
paul@laika:~$ strings /bin/ls
```

/lib/ld-linux.so.2

librt.so.1

__gmon_start__

```
_Jv_RegisterClasses  
clock_gettime  
libacl.so.1  
...
```