

Занятие 9. Сценарии командной оболочки, циклы

Занятие 9. Сценарии командной оболочки, циклы

Команда `for`

Команда `while`

Команда `until`

Операторы `break`, `continue`

Литература

Довольно часто возникает необходимость повторять ряд команд до тех пор, пока не будет выполнено какое-то условие. Командная оболочка `bash` предоставляет несколько операторов для организации циклов.

Команда `for`

Синтаксис команды `for` имеет следующий вид

```
1  for переменная [in список]; do  
2      команды  
3  done
```

Рассмотрим пример:

```
1  user@linux-pc:~/bin$ cat for_1  
2  #!/bin/bash  
3  # Цикл for, простой вариант  
4  
5  for city in Moscow "Saint Petersburg" Vladivostok; do  
6      echo $city  
7  done  
8  user@linux-pc:~/bin$ for_1  
9  Moscow  
10 Saint Petersburg  
11 Vladivostok  
12 user@linux-pc:~/bin$
```

После последней итерации переменная `city` остается действительной до конца сценария.

Команда `for` может «читать» данные из переменной:

```

1 user@linux-pc:~/bin$ cat for_2
2 #!/bin/bash
3 # Цикл for, чтение данных из переменной
4
5 LIST="We dont\t need no education"
6 for item in $LIST; do
7     echo $item
8 done
9 user@linux-pc:~/bin$ for_2
10 We
11 dont\t
12 need
13 no
14 education
15 user@linux-pc:~/bin$

```

Специальная переменная среды IFS (Internal Field Separator), которую называют внутренним разделителем полей, определяет список переменных, который используется командной оболочкой в качестве разделителей. По умолчанию эта переменная содержит следующие значения:

- пробел;
- табуляция;
- символ перехода на новую строку.

Типичное использование этой переменной выглядит следующим образом:

```

1 #!/bin/bash
2 # Цикл for, чтение данных из переменной, использование IFS
3
4 LIST="one,two,three,four"
5
6 IFS_OLD="$IFS"
7 IFS=","
8
9 for item in $LIST; do
10     echo $item
11 done
12
13 IFS="$IFS_OLD"

```

Такой подход позволяет гарантировать восстановление значения по умолчанию для переменной IFS.

Команду for часто используют для обработки файлов в каталоге:

```

1 #!/bin/bash
2 # Итерации по всем файлам в каталоге
3 for file in ~/*; do
4     if [ -d "$file" ]; then
5         echo "$file - каталог"
6     elif [ -f "$file" ]; then
7         echo "$file - файл"
8     fi
9 done

```

Команда while

Синтаксис команды while имеет следующий вид

```
1 while команды; do
2     команды
3 done
```

Как и команды if, команда while проверяет код завершения команд, указанных в заголовке цикла. Пока код завершения равен 0, команда while выполняет команды, указанные в теле цикла.

```
1 user@linux-pc:~/bin$ cat while_1
2 #!/bin/bash
3 # Пример цикла while
4 count=1
5 while [ $count -le 5 ]; do
6     echo $count
7     count=$((count + 1))
8 done
9
10 user@linux-pc:~/bin$ while_1
11 1
12 2
13 3
14 4
15 5
16 user@linux-pc:~/bin$
```

Команда until

Синтаксис команды until имеет следующий вид

```
1 until команды; do
2     команды
3 done
```

Команда until очень похожа на команду while, но в отличие от while завершает цикл когда обнаруживает нулевой код возврата.

```
1 user@linux-pc:~/bin$ cat until_1
2 #!/bin/bash
3 # Пример цикла until
4 count=1
5 until [ $count -gt 5 ]; do
6     echo $count
7     count=$((count + 1))
8 done
9
10 user@linux-pc:~/bin$ until_1
11 1
```

```
12 2
13 3
14 4
15 5
16 user@linux-pc:~/bin$
```

Операторы `break`, `continue`

Команда `break` представляет простой способ досрочного выхода из любого цикла.

```
1 user@linux-pc:~/bin$ cat break_1
2 #!/bin/bash
3 # Использование оператора break
4 for i in 1 2 3 4 5 6 7 8 9 10; do
5     if [ $i -eq 5 ]; then
6         break
7     fi
8     echo "Итерация #$i"
9 done
10 echo "Цикл for закончился"
11
12 user@linux-pc:~/bin$ break_1
13 Итерация #1
14 Итерация #2
15 Итерация #3
16 Итерация #4
17 Цикл for закончился
18 user@linux-pc:~/bin$
```

По умолчанию команда `break` завершает самый вложенный цикл. Однако у команды `break` есть целочисленный параметр, который показывает уровень цикла, из которого должен быть выполнен выход.

Команда `continue` позволяет передать управление в конец тела цикла, тем самым пропустив его часть.

```
1 user@linux-pc:~/bin$ cat continue_1
2 #!/bin/bash
3 # Использование оператора continue
4 for i in 1 2 3 4 5 6 7 8 9 10; do
5     if [ $i -gt 2 ] && [ $i -lt 9 ]; then
6         continue
7     fi
8     echo "Итерация #$i"
9 done
10
11 user@linux-pc:~/bin$ continue_1
12 Итерация #1
13 Итерация #2
14 Итерация #9
15 Итерация #10
16 user@linux-pc:~/bin$
```

Организация циклов на основе данных файла

```
1  #!/bin/bash
2  # Обработка файла с помощью циклов
3  IFS_OLD="$IFS"
4  IFS=$'\n'
5  for entry in `cat /etc/passwd`; do
6      echo "Строка: $entry"
7      IFS=:
8      for item in $entry; do
9          echo $'\t' "$item"
10     done
11     echo
12 done
13 IFS="$IFS_OLD"
```

Литература

1. Шотес У. “Командная строка Linux.”, главы 29, 33
2. Блум Р., Бреснахэн К. “Командная строка Linux и сценарии оболочки.”, глава 12