

Занятие 12. Сценарии командной оболочки, чтение ввода с клавиатуры

Занятие 12. Сценарии командной оболочки, чтение ввода с клавиатуры

Литература

Недостаток сценария, который демонстрирует работу с целыми числами, заключается в том, что приходится изменять сам сценарий для обработки нового числа. Пользоваться этим сценарием было бы удобнее, если бы сценарий предлагал пользователю ввести значение.

Для чтения одной строки со стандартного ввода используется команда `read`:

```
1 | read [параметр...] [переменная]
```

где параметр — это один или несколько параметров команды, с назначением которых можно ознакомиться в документации, переменная — имя одной или нескольких переменных для сохранения введенного значения. Если имя переменной не указано, строка с данными сохраняется в переменной `REPLY`.

```
1 |#!/bin/bash
2 |# Проверка целочисленного значения
3 |
4 |echo -n "Введите целое число: "
5 |read int
6 |
7 |if ! echo "$int" | grep -Eq "[+-]?[0-9][0-9]*" ; then
8 |    echo "Введено не число ($int)"
9 |    exit 1
10|fi
11|
12|if [ $int -eq 0 ]; then
13|    echo "Число равно нулю"
14|else
15|    if [ $int -lt 0 ]; then
16|        echo "Число отрицательное"
17|    else
18|        echo "Число положительное"
19|    fi
20|
```

```

21  if [  $((int \% 2))$  -eq 0 ]; then
22      echo "Число четное"
23  else
24      echo "Число нечетное"
25  fi
26  fi

```

Обратите внимание на несколько моментов. Во-первых, в первой команде `echo` используется параметр `-n`, который подавляет переход на новую строку после вывода сообщения. Во-вторых, для проверки правильности ввода целого числа используется команда `grep`. Входные данные (строка) передаются этой команде с помощью конвейера, а сама команда вызывается с двумя параметрами: `-E` — задает использование расширенных регулярных выражений, `-q` — подавляет выдачу сообщений на экран, если строка соответствует шаблону, возвращается значение 0.

Расширенные регулярные выражения понадобились из-за квантификатора `?`, который обозначает ни одного или одно вхождение символа, к которому он применяется.

Команда `read` может сохранять ввод в несколько переменных:

```

1  #!/bin/bash
2  # Чтение нескольких значений
3  echo -n "Введите несколько слов: "
4  read var_1 var_2 var_3
5  echo "var_1 = '$var_1'"
6  echo "var_2 = '$var_2'"
7  echo "var_3 = '$var_3'"

```

Обратите внимание, как действует команда `read`, когда получает разное количество параметров:

```

1  igor1@DESKTOP-0IDSOVA:~/bin$ read_2
2  Введите несколько слов: one two three
3  var_1 = 'one'
4  var_2 = 'two'
5  var_3 = 'three'
6  igor1@DESKTOP-0IDSOVA:~/bin$ read_2
7  Введите несколько слов: one two
8  var_1 = 'one'
9  var_2 = 'two'
10 var_3 = ''
11 igor1@DESKTOP-0IDSOVA:~/bin$ read_2
12 Введите несколько слов: one two three four
13 var_1 = 'one'
14 var_2 = 'two'
15 var_3 = 'three four'
16 igor1@DESKTOP-0IDSOVA:~/bin$

```

Следующий сценарий показывает использование различных параметров команды `read`. Попробуйте догадаться, что делает тот или иной параметр. Если у вас это не получилось, обратитесь к документации.

```

1  #!/bin/bash
2  if read -t 10 -sp "Введите пароль: " ; then
3      echo -e "\nПароль = '$REPLY'"
4  else
5      echo -e "\nИстек тайм-аут для ввода" >&2
6      exit 1
7  fi

```

Часто для организации интерактивной работы используется меню. Рассмотрим соответствующий пример:

```

1  #!/bin/bash
2  # Реализация меню
3  clear
4  echo "
5  Выберите:
6  1. Системная информация
7  2. Использование дискового пространства
8  0. Выход
9  "
10 read -p "Введите [0-3]: "
11 if echo "$REPLY" | grep -q "^[0-3]$" ; then
12     if [ $REPLY -eq 0 ]; then
13         echo "Bye"
14         exit
15     fi
16     if [ $REPLY -eq 1 ]; then
17         echo "Hostname: $HOSTNAME"
18         uptime
19         exit
20     fi
21     if [ $REPLY -eq 2 ]; then
22         df -h
23         exit
24     fi
25 else
26     echo "Неверный ввод" >&2
27     exit 1
28 fi

```

Литература

1. Шоттс У. “Командная строка Linux.”, глава 28
2. Блум Р., Бреснахэн К. “Командная строка Linux и сценарии оболочки.”, глава 13, подраздел “Получение ввода данных от пользователя”