

# Встраивание и параметры командных оболочек

В данной главе приводится краткий обзор методов использования **дочерних и встраиваемых командных оболочек**, а также

описываются некоторые **параметры командных оболочек**.

## Встраивание командных оболочек

Командные оболочки могут подвергаться **встраиванию** в рам-ках строк команд или, другими словами, в ходе разбора строк команд могут создаваться новые процессы, являющиеся копиями процесса текущей командной оболочки. Вы можете использовать переменные для доказательства факта создания новых командных оболочек. В примере ниже переменная **\$var1** существует исключительно в рам-ках дочерней (временной) командной оболочки.

```
[paul@RHELv4u3 gen]$ echo $var1
```

```
[paul@RHELv4u3 gen]$ echo $(var1=5;echo $var1)
```

```
[paul@RHELv4u3 gen]$ echo $var1
```

```
[paul@RHELv4u3 gen]$
```

Вы можете осуществлять встраивание командной оболочки в рамках уже **встроенной командной** оболочки, причем сам описанный процесс называется **многоуровневым встраиванием** командных оболочек.

В примере ниже демонстрируется работа встроенной командной оболочки, функционирующей в рамках другой встроенной командной оболочки.

```
paul@deb503:~$ A='командная оболочка'
paul@deb503:~$ echo $C$B$A $(B='встроенная ';echo $C$B$A;
echo $(C='встроенная ';echo $C$B$A))
командная оболочка встроенная командная оболочка встроен-
ная встроенная командная оболочка
```

## Обратные кавычки

Однократное встраивание командной оболочки может оказаться полезным в том случае, если необходимо избежать изменения текущей директории. В примере ниже для встраивания командной оболочки используются **обратные кавычки** вместо рассмотренного ранее символа доллара со скобками.

```
[paul@RHELv4u3 ~]$ echo `cd /etc; ls -d * | grep pass`
passwd passwd- passwd.0LD
[paul@RHELv4u3 ~]$
```

Вы можете использовать нотацию **\$()** при необходимости осуществления многоуровневого встраивания командных оболочек, так как эту задачу невозможно решить с помощью **обратных кавычек**.

## Обратные кавычки или одинарные кавычки

Размещение встраиваемых команд между **обратными кавычками** вместо круглых скобок со знаком доллара позволяет отказаться от использования одного дополнительного символа. Однако, следует быть очень осторожным, так как обратные кавычки часто путают с одинарными кавычками. В техническом плане различия между кавычками **'** и **`** являются значительными.

```
[paul@RHELv4u3 gen]$ echo `var1=5;echo $var1`
```

5

```
[paul@RHELv4u3 gen]$ echo 'var1=5;echo $var1'
var1=5;echo $var1
[paul@RHELv4u3 gen]$
```

## Параметры командной оболочки

И команда **set**, и команда **unset** являются встроенными командами командной оболочки. Они могут использоваться для установки значений параметров самой командной оболочки **bash**. Проясним это утверждение, рассмотрев следующий пример. По умолчанию командная оболочка будет рассматривать неустановленные переменные как переменные, не имеющие ассоциированных значений. После установки значения параметра **-u** командная оболочка будет рассматривать любые обращения к неустановленным переменным как ошибки. Обратитесь к странице руководства для командной оболочки **bash** для получения дополнительной информации.

```
[paul@RHEL4b ~]$ echo $var123
```

```
[paul@RHEL4b ~]$ set -u
[paul@RHEL4b ~]$ echo $var123
-bash: var123: unbound variable
[paul@RHEL4b ~]$ set +u
[paul@RHEL4b ~]$ echo $var123
```

```
[paul@RHEL4b ~]$
```

Для вывода списка всех параметров вашей командной оболочки с установленными значениями следует использовать команду **echo \$-**. Параметр **noclobber** (или **-C**) будет описан позднее в рамках данной книги (в главе, посвященной перенаправлению потоков ввода/вывода).

```
[paul@RHEL4b ~]$ echo $-
himBH
[paul@RHEL4b ~]$ set -C ; set -u
[paul@RHEL4b ~]$ echo $-
himuBCH
[paul@RHEL4b ~]$ set +C ; set +u
[paul@RHEL4b ~]$ echo $-
```

himBH

[paul@RHEL4b ~]\$

При исполнении команды **set** без параметров вашему вниманию будет представлен список всех параметров без осуществления каких-либо изменений в том случае, если командная оболочка работает в режиме **posix**. Вы можете перевести командную оболочку в режим **posix**, выполнив команду **set -o posix**.