

# Shell-программирование

## *Создание и запуск shell-программ (сценариев)*

Создать текстовый файл (например, с помощью cat)

```
cat > proba
```

```
>echo "Hello, everybody"
```

```
> ^C
```

Изменить права доступа (добавить право запуска)

```
chmod +x proba
```

Запустить программу

```
./proba
```

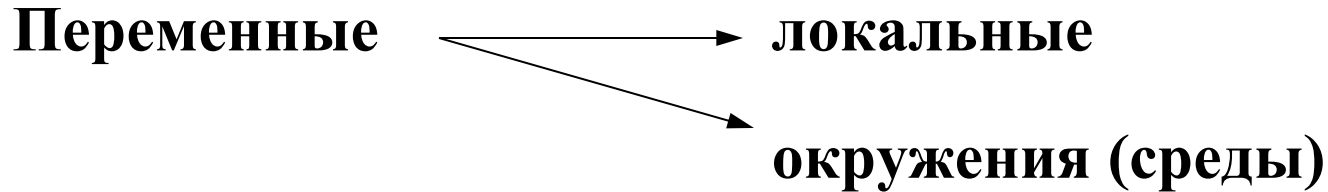
# Комментарий в shell-программах

Комментарий в Shell-процедуре — #

**#!** в первой строке командного файла задает командный интерпретатор для обработки команд сценария, например

**#! /bin/csh**

# Shell-переменные



## *Вывод значений переменных*

echo \$имя\_переменной

env      вывод всех переменных среды

set      вывод всех переменных (локальных и среды) и их значений

unset    имя\_переменной      присвоение    переменной  
значения NULL

# Среда пользователя (окружение)

Среда пользователя описывает сеанс работы с системой для программ, которые он запускает.

## Синтаксис:

**env**

## Пример:

```
$ env
```

```
HOME=/home/gerry
```

```
EDITOR=vi
```

```
TERM=70092
```

```
...
```

```
PATH=/usr/bin:/usr/local/bin:/home/gerry/bin
```

# Переменные среды

HOME

PATH

LOGNAME

SHELL

и др.

Некоторые из этих переменных устанавливаются для всех пользователей, другие устанавливаются явно в файлах инициализации (например, **/etc/profile** или **.profile**)

# Пользовательские файлы инициализации

командный интерпретатор	файл инициализации
Bourne shell (sh)	.profile
C shell (csh)	.login .cshrc
Korn shell (ksh)	.profile .kshrc
Bourne Again shell (bash)	.profile .bashrc

## **Shell: вывод на экран всех переменных, содержащихся в данный момент в среде**

```
$ env
```

```
MANPATH=/usr/share/man:/usr/local/man
```

```
PATH=/usr/bin:/usr/ccs/bin:/usr/local/bin
```

```
LOGNAME=user3
```

```
ERASE=^H
```

```
SHELL=/usr/bin/sh
```

```
HOME=/home/user3
```

```
TERM=hpترم
```

```
PWD=/home/user3
```

```
TZ=PST8PDT
```

```
EDITOR=/usr/bin/vi
```

# Shell: установка значений переменных

## Синтаксис:

переменная=значение

## Пример:

\$ color=lavender

Установка локальной переменной

\$ count=3

Установка локальной переменной

\$ dir\_name=tree/car.models/ford

Установка локальной переменной

\$ PS1=hi\_there\$

Изменение переменной среды



# Управляющие символы команды echo

Символ	Действие при печати
<code>\b</code>	Возврат влево на один символ
<code>\c</code>	Подавление перехода на новую строку
<code>\f</code>	Перевод страницы
<code>\n</code>	Новая строка
<code>\r</code>	Переход в начало строки
<code>\t</code>	Символ табуляции
<code>\\</code>	Обратный слэш
<code>\nnn</code>	Символ, код ASCII которого nnn

# Использование кавычек

Ряд символов имеет «специальный» смысл для shell`а:

пробел

возврат каретки

\$

#

\*

< >

«Закавычивание» позволяет отменить особый смысл специальных символов.

## Символы закавычивания

Обратный слэш

\

Одиночные кавычки

'

Двойные кавычки

“

## «Закавычивание»: символ \

\

Отменяет специальный смысл  
непосредственно следующего за ним  
символа

### Пример:

```
$ echo the \$text
```

```
the $text
```

```
$ color=red\ white\ and\ blue
```

```
$ echo the value of \$color is $color
```

```
the value of $color is red white and blue
```

```
$ echo one two \
```

```
>three four
```

```
one two three four
```

## «Закавычивание»: символ ‘

‘ ’

Отменяют специальный смысл  
всех символов, заключенных между  
одинокими кавычками

### Пример:

```
$ color='red white and blue'
```

```
$ echo 'the value of \${color} is ${color}'
```

```
the value of \${color} is ${color}
```

```
$ echo 'the value of ${color} is' ${color}
```

```
the value of ${color} is red white and blue
```

## «Закавычивание»: символ “

“

Отменяет специальный смысл всех символов, заключенных между двойными кавычками, за исключением \, \$, {имя переменной}, \$(команда) и “

### Пример:

```
$ color="red white and blue"
```

```
$ echo "the value of \${color} is ${color}"
```

```
the value of ${color} is red white and blue
```

```
$ cur_dir="$LOGNAME - your current directory is $(pwd)"
```

```
$ echo $cur_dir
```

```
user3 - your current directory is /home/user3/tree
```

```
$ echo "they're all here, \\, ', \" "
```

```
they're all here, \, ', "
```

# Вывод значений переменных - пример

```
$ echo $HOME  
/home/user3
```

```
$ env  
HOME=/home/user3  
PATH=/usr/bin:/usr/contrib/bin:/usr/local/bin  
SHELL=/usr/bin/sh
```

```
$ set  
HOME=/home/user3  
PATH=/usr/bin:/usr/contrib/bin:/usr/local/bin  
SHELL=/usr/bin/sh  
color=lavender  
count=3  
dir_name=/home/user3/tree
```

```
$ unset dir_name
```

# Экспорт локальных переменных в среду

## Синтаксис:

**export [переменная[=значение]]**

Экспортирование  
переменной в среду

## Пример:

```
$ color=lavender
```

```
$ export color
```

```
$ export count=3
```

```
$ export
```

```
export PATH=/usr/bin:/usr/ccs/bin:/usr/contrib/bin:/usr/local/bin
```

```
export color=lavender
```

```
export count=3
```



# Экспорт локальных переменных в среду

## Пример:

```
$ color=blue
$ export color
$ echo $color
blue
$ ./proga
    my color is blue
    my color is red
$ echo $color
blue
```

```
#proga
echo my color is $color
color=red
export color
echo my color is $color
```

## Передача данных shell-процедуре

- Задание среды (переменные окружения)
- Аргументы командной строки
- Использование ввода

# Аргументы shell-процедур

## Командная строка:

\$ sh_program	arg1	arg2	...	arg23	....	argX
\$0	\$1	\$2		\${23}		\$X

## Пример:

```
$ cat color3
echo You are now running program: $0
echo The value of command line argument \#1 is: $1
echo The value of command line argument \#2 is: $2
$ chmod +x color3
$ ./color3 red green
You are now running program: color3
The value of command line argument #1 is: red
The value of command line argument #2 is: green
```

# Аргументы shell-процедур

## Пример:

Данная shell-процедура устанавливает программу, специфицированную как аргумент командной строки в каталог `bin` пользователя.

```
$ cat > my_install
echo $0 will install $1 to your bin directory
chmod +x $1
mv $1 $HOME/bin
echo Installation of $1 is complete

$ chmod +x my_install
$ ./my_install color3
my_install will install color3 to your bin directory
Installation of color3 is complete
$
```

# Специальные переменные shell: # и \*

- # Количество аргументов в командной строке
- \* Строка, содержащая все аргументы командной строки (за исключением \$0)

## Пример:

```
$ cat color4
echo There are $# command line arguments
echo They are $*
echo The first command line argument is $1

$ chmod +x color4

$ ./color4 red green yellow blue
The are 4 command line arguments
They are red green yellow blue
The first command line argument is red
$
```

# Специальные переменные shell: # и \*

## Пример:

```
$ cat > my_install2
echo $0 will install  $# files to your bin directory
echo The files to be installed are: $*
chmod  +x  $*
mv  $* $HOME/bin
echo Installation is complete

$ chmod +x my_install2

$ my_install2 color1 color2
my_install2 will install  2 files to your bin directory
The files to be installed are: color1 color2
Installation is complete
```

# Специальные переменные shell: # и \*

## Пример:

```
$ cat list_dir
  cd $*
  echo You are in the $(pwd) directory
  echo The contents of this directory are:
  ls -F
```

```
$ list_dir dir1 dir2
sh: cd: bad arguments count
```

# Команда shift

- Сдвигает все строку аргументов влево на n позиций
- Уменьшает переменную # на n (по умолчанию n=1)

## Синтаксис:

shift [n]

Команду shift полезно использовать для:

- доступа к позиционным параметрам в группах, например, совокупности координат x и y
- удаления опций из командной строки



# Команда shift

## Пример:

```
$ cat color5
orig_args="$*"
echo There are $# command line arguments
echo They are $*
echo Shifting two arguments
shift 2
echo There are $# command line arguments
echo They are $*
echo Shifting two arguments
shift 2; final_args=$*
echo Original arguments are: $orig_args
echo Final arguments are: $final_args
```

# Команда shift

```
$ color5    red    green  yellow  blue  orange  black
```

There are 6 command line arguments

They are red green yellow blue orange black

Shifting two arguments

There are 4 command line arguments

They are yellow blue orange black

Shifting two arguments

Original arguments are: red green yellow blue orange  
black

Final arguments are: orange black

```
$
```

# Команда read

## Синтаксис:

read переменная [переменная...]

## Пример:

```
$ cat color6
#This program prompts for user input
echo "Please enter your favorite two colors -> \c"
read color_a color_b
echo The colors you entered are: $color_b $color_a
$ chmod +x color6
$ color6
Please enter your favorite two colors -> red blue
The colors you entered are: blue red
$ color6
Please enter your favorite two colors -> red blue tan
The colors you entered are: blue tan red
```

# Команда read

## Пример:

```
$ cat > my_install3
echo $0 will install  files to your bin directory
echo "Enter the names of the files -> \c"
read filenames
chmod +x $filenames
mv $filenames $HOME/bin
echo Installation is complete
```

```
$ chmod +x my_install3
```

```
$ my_install3
my_install3 will install  files to your bin directory
Enter the names of the files -> f1 f2
Installation is complete
```