

Bash-скрипты

Кузнецов Антон Михайлович

Санкт-Петербургский Академический Университет

10 сентября 2013 г.

- Любой bash-скрипт должен начинаться со строки:
`#!/bin/bash`
(Последовательность `#!` называется Sha-Bang - указание интерпретатора)
- Запуск интерпретатора:
`bash scriptname [arguments]`
- Комментарии начинаются с символа `#`

- Сделать исполняемым:
`chmod 555 scriptname`
а затем просто запустить:
`./scriptname`
- Если поместить сценарий в каталог `/usr/local/bin`, то вызвать его можно просто набрав название файла в командной строке

Ключ `bash -x`

```
bash -x script.sh
+ echo "Hello, World!"
Hello, World!
+ who
antonk  tty1  2011-09-28 16:01
```

`set -x` – включить отладку с этого момента

`set +x` – выключить отладку

Могут быть строками, целыми числами и массивами.
Можно объявлять константы.

- `export foo=93` – экспорт во все дочерние оболочки.
- `foo=93` – установка для текущей оболочки
* вокруг знака `=` пробелов быть не должно!
- `printenv` – вывести глобальные переменные.
Пример: `HOME`, `MAIL`, `PATH`, `LC_ALL`, `PWD`, `UID`.
- `set` – вывести все переменные и функции.

Пример с переменными

```
a=1
b=$a
echo $b # 1
hello="A B C"
echo hello # hello
echo $hello # A B C
echo "$hello" # A B C
echo '$hello' # $hello
var=23
unset var
```

```
arr[0]=0  
arr[5]=45  
arr2=( zero one two three )  
arr3=( [17]=семнадцать [21]=двадцатьОдин )  
echo ${arr[5]}    # нужны скобки!
```

- `$@` – параметры скрипта (все!)
- `$0` – имя скрипта
- `$1, $2, $3, ...` – параметры скрипта по одному
- `$#` – количество переданных аргументов
- `$*`, `$@` – специальная переменная, содержащая все аргументы

- `a=1`
- `let a=16+5 # let` – арифметические действия
- `for a in 7 8 9 11`
 `do ...`
 `done`
- `read a`
- `a=`echo Hello!``
- `echo $a # Hello!`
- `a=`ls -l``
- `a=$(ls) # то же самое`

Вычисляет заданное выражение

!! аргументы должны отделяться пробелами!!

Примеры:

```
expr 3 + 5 #8
```

```
expr 5 3 #15 экранируем *
```

```
y=$(expr $y + 1) #инкремент
```

```
for arg in list
do
command(s)
done
```

```
for arg in $( seq n ) # 1 2 3 ... n
for arg in {1..n} # 1 2 3 ... n
```

```
for arg in "$@"
```

Задание - 1

Написать скрипт, который подсчитывает количество файлов в каталоге с учетом подкаталогов.

Задание - 2. Тесты

Написать скрипт, который принимает на вход 2 числа (n , m). После чего создает N каталогов (с названиями `input1`, `input2...`) в каждом из которых создает файл `input.txt` состоящий из m случайных чисел.

Создайте в `bash` массив и заполните его числами Фибоначчи от 0-го до N-го. Число N передается в качестве параметра. Если параметр не указан, то должна быть выдана ошибка.