

# Лабораторная работа 10

Программирование в командном процессоре ОС UNIX. Командные  
файлы

Гарут Александр Игоревич

# Содержание

Цель работы	3
Ход работы	4
Вывод	7
Контрольные вопросы	8

## Цель работы

Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы.

## Ход работы

1. Напишем скрипт, который при запуске делает резервную копию себя, в backup в домашнем каталоге, архивируя себя.

```
tar -cf script01.tar script01
mv script01.tar /home/garutalek/backup/script01.tar
```

Изображение1.1: Код скрипта

Удостовермся что всё работает

```
[garutalek@fedora ~]$ chmod +x script01
[garutalek@fedora ~]$ ./script01
[garutalek@fedora ~]$ cd backup/
[garutalek@fedora backup]$ ls
script01.tar
```

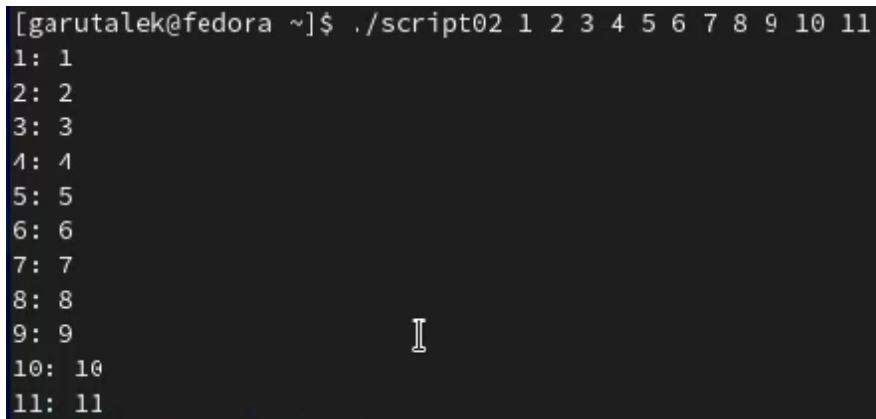
Изображение1.2: Запуск скрипта

2. Напишем скрипт, принимающий произвольное число аргументов, выводящий их же.

```
count=1
for param in "$@"
do
echo "$count: $param"
count=$((count + 1))
done
```

Изображение2.1: Код скрипта

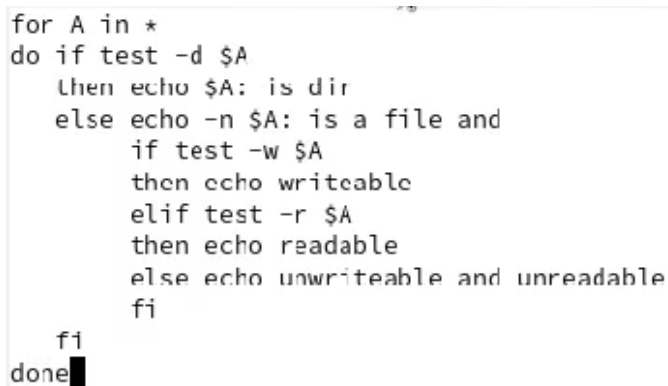
Удостовермся что всё работает

A terminal window with a black background and white text. The prompt is [garutalek@fedora ~]\$ and the command is ./script02 1 2 3 4 5 6 7 8 9 10 11. The output consists of 11 lines, each showing a number followed by a colon and the same number (e.g., 1: 1, 2: 2, ..., 11: 11).

```
[garutalek@fedora ~]$ ./script02 1 2 3 4 5 6 7 8 9 10 11
1: 1
2: 2
3: 3
4: 4
5: 5
6: 6
7: 7
8: 8
9: 9
10: 10
11: 11
```

Изображение2.2: Запуск скрипта

3. Напишем скрипт, аналог ls, который выводит файлы из директории и их права доступа.

A code editor window with a light gray background. It contains a shell script that iterates over files in the current directory and prints their permissions. The script uses 'for' loops, 'if' statements, and 'test' commands to check for directories, writeability, and readability.

```
for A in *
do if test -d $A
then echo $A: is dir
else echo -n $A: is a file and
if test -w $A
then echo writeable
elif test -r $A
then echo readable
else echo unwriteable and unreadable
fi
fi
done
```

Изображение3.1: Код скрипта

Удостовермся что всё работает

```
[garutalek@fedora ~]$ ./script03
australia: is dir
backup: is dir
conf.txt: is a file andwriteable
feathers: is a file andwriteable
feathers123: is a file andwriteable
file.txt: is a file andwriteable
grep: is a file andwriteable
#lab07.sh#: is a file andwriteable
lab07.sh: is a file andwriteable
lab07.sh~: is a file andwriteable
mkdir: is dir
my_os: is a file andreadable
play: is dir
script~: is a file andwriteable
script01: is a file andwriteable
script02: is a file andwriteable
script02~: is a file andwriteable
script03: is a file andwriteable
```

Изображение3.2: Запуск скрипта

4. Напишем скрипт, который выводит количество файлов заданного разрешения в заданной директории.

```
echo "Insert dir: "
read dirr
echo "Insert file format: "
read form
find $dirr -name "$form" -type f | wc -l
```

Изображение4.1: Код скрипта

Удостовермся что всё работает

```
[garutalek@fedora ~]$ ./script04
Insert dir:
/hcme/garutalek/
Insert file format:
.txt
6
```

Изображение4.2: Запуск скрипта

# Вывод

Изучили основы программирования в оболочке ОС UNIX/Linux. Научились писать небольшие командные файлы.

# Контрольные вопросы

1. Командная оболочка Unix (англ. Unix shell, часто просто «шелл» или «sh») — командный интерпретатор, используемый в операционных системах семейства Unix, в котором пользователь может либо давать команды операционной системе по отдельности, либо запускать скрипты, состоящие из списка команд.
2. POSIX (англ. Portable Operating System Interface — переносимый интерфейс операционных систем) — набор стандартов, описывающих интерфейсы между операционной системой и прикладной программой (системный API), библиотеку языка C и набор приложений и их интерфейсов. Стандарт создан для обеспечения совместимости различных UNIX-подобных операционных систем и переносимости прикладных программ на уровне исходного кода, но может быть использован и для не-Unix систем.