

# Программирование в командном процессоре ОС UNIX.

## Командные файлы

---

Матвеева Анастасия Сергеевна НПМбд-02-20<sup>1</sup>

26 мая, 2021, Москва, Россия

<sup>1</sup>RUDN University, Moscow, Russian Federation

## Цель работы

---

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

## Задачи лабораторной работы

---

Задачи:

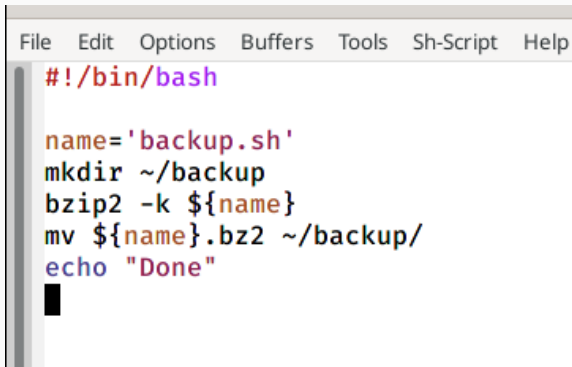
1. Познакомиться с командными процессорами.
2. Изучить переменные, арифметические операторы в языке программирования `bash`.
3. Изучить операторы цикла `for`, `while` и `until`, оператор выбора `case`, условный оператор `if`.
4. В ходе работы написать 4 скрипта.

## Выполнение лабораторной работы

---

## Выполнение лабораторной работы

Для начала напишем скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup в домашнем каталоге. При этом файл должен архивироваться архиватором bzip2. После написания проверяем его работу. (рис. 1).

A screenshot of a text editor window with a menu bar containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The editor displays a shell script with the following content: 

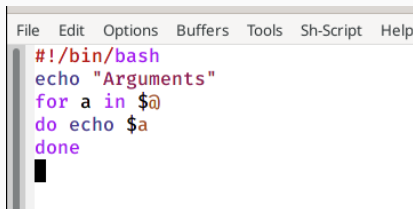
```
#!/bin/bash

name='backup.sh'
mkdir ~/backup
bzip2 -k ${name}
mv ${name}.bz2 ~/backup/
echo "Done"
```

 The script is written in a monospaced font with syntax highlighting: the shebang is red and purple, variable assignments are orange, and the echo command is blue. A black cursor is visible at the end of the last line.

Figure 1: 1 скрипт

Далее напишем пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять и проверим его работу.(рис. 2)

A screenshot of a terminal window with a menu bar containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The terminal displays a bash script with the following content: 

```
#!/bin/bash
echo "Arguments"
for a in $@
do echo $a
done
```

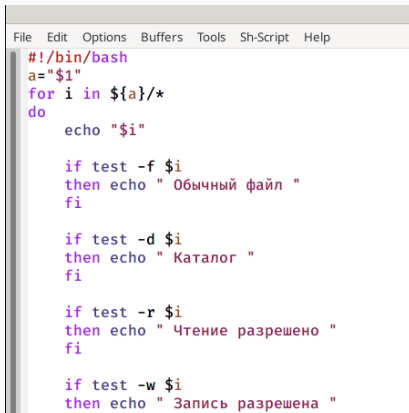
 A black cursor is visible at the end of the last line.

Figure 2: 2 скрипт



## Выполнение лабораторной работы

Теперь напишем командный файл — аналог команды `ls` (без использования самой этой команды и команды `dir`). Требуется, чтобы он выдавал информацию о нужном каталоге и выводил информацию о возможностях доступа к файлам этого каталога. Проверим работу данного скрипта (рис. 3)

A screenshot of a terminal window with a menu bar containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The terminal displays a shell script that iterates through files in a directory 'a' and prints their permissions and names in Russian. The script uses 'test' commands to check for file, directory, and read/write permissions.

```
#!/bin/bash
a="$1"
for i in ${a}/*
do
    echo "$i"

    if test -f $i
    then echo " Обычный файл "
    fi

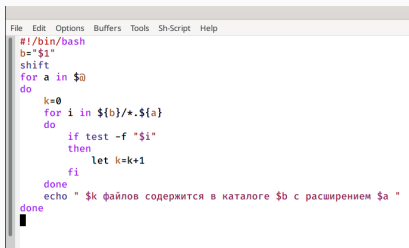
    if test -d $i
    then echo " Каталог "
    fi

    if test -r $i
    then echo " Чтение разрешено "
    fi

    if test -w $i
    then echo " Запись разрешена "
```

# Выполнение лабораторной работы

Напишем командный файл, который получает в качестве аргумента командной строки формат файла (.txt,.doc,.jpg,.pdfи т.д.) и вычисляет количество таких файлов в указанной директории. Путь к директории также передаётся ввиде аргумента командной строки. Также проверяем работу скрипта.(рис. 4)

A screenshot of a terminal window with a menu bar containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The terminal displays a bash script with the following content:

```
#!/bin/bash
b="$1"
shift
for a in $@
do
    k=0
    for i in ${b}/*.${a}
    do
        if test -f "$i"
        then
            let k=k+1
        fi
    done
    echo " $k файлов содержится в каталоге $b с расширением $a "
done
```

Figure 4: 4 скрипт

## Выводы

---

В ходе выполнения данной лабораторной работы я изучила основы программирования в оболочке ОС UNIX/Linux и научилась писать небольшие командные файлы.