

Лабораторная работа №11

**Программирование в командном процессоре ОС UNIX. Командные
файлы**

Ким Михаил Алексеевич

Содержание

1	Цель работы	3
2	Выполнение лабораторной работы	4
3	Выводы	12
4	Термины	13

1. Цель работы

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

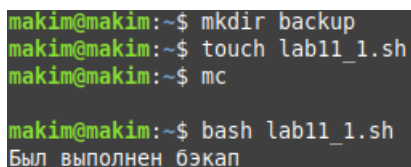
2. Выполнение лабораторной работы

1. Пишем скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup в вашем домашнем каталоге. При этом файл будет архивироваться с помощью gzip. (рис. 2.1-2.3)

```
// Используем shebang, чтобы сообщить командной оболочке, каким образом мы бу  
#!/bin/bash
```

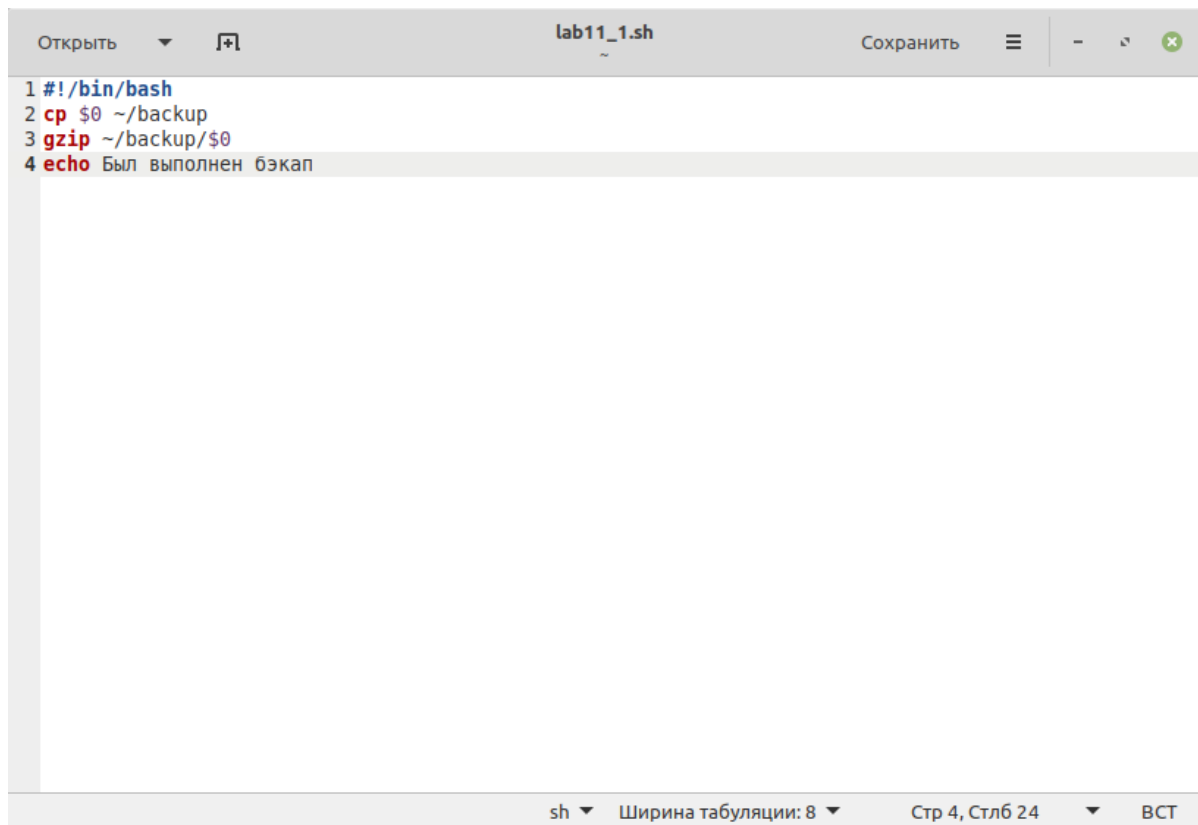
```
// копируем командный файл в директорию backup  
// $0 содержит в себе имя командного файла  
cp $0 ~/backup
```

```
// архивируем командный файл  
gzip ~/backup/$0
```



```
makim@makim:~$ mkdir backup  
makim@makim:~$ touch lab11_1.sh  
makim@makim:~$ mc  
  
makim@makim:~$ bash lab11_1.sh  
Был выполнен бэкап
```

Рис. 2.1: Создание файл



```
1 #!/bin/bash
2 cp $0 ~/backup
3 gzip ~/backup/$0
4 echo Был выполнен бэкап
```

sh Ширина табуляции: 8 Стр 4, Стлб 24 ВСТ

Рис. 2.2: Исходный код скрипта

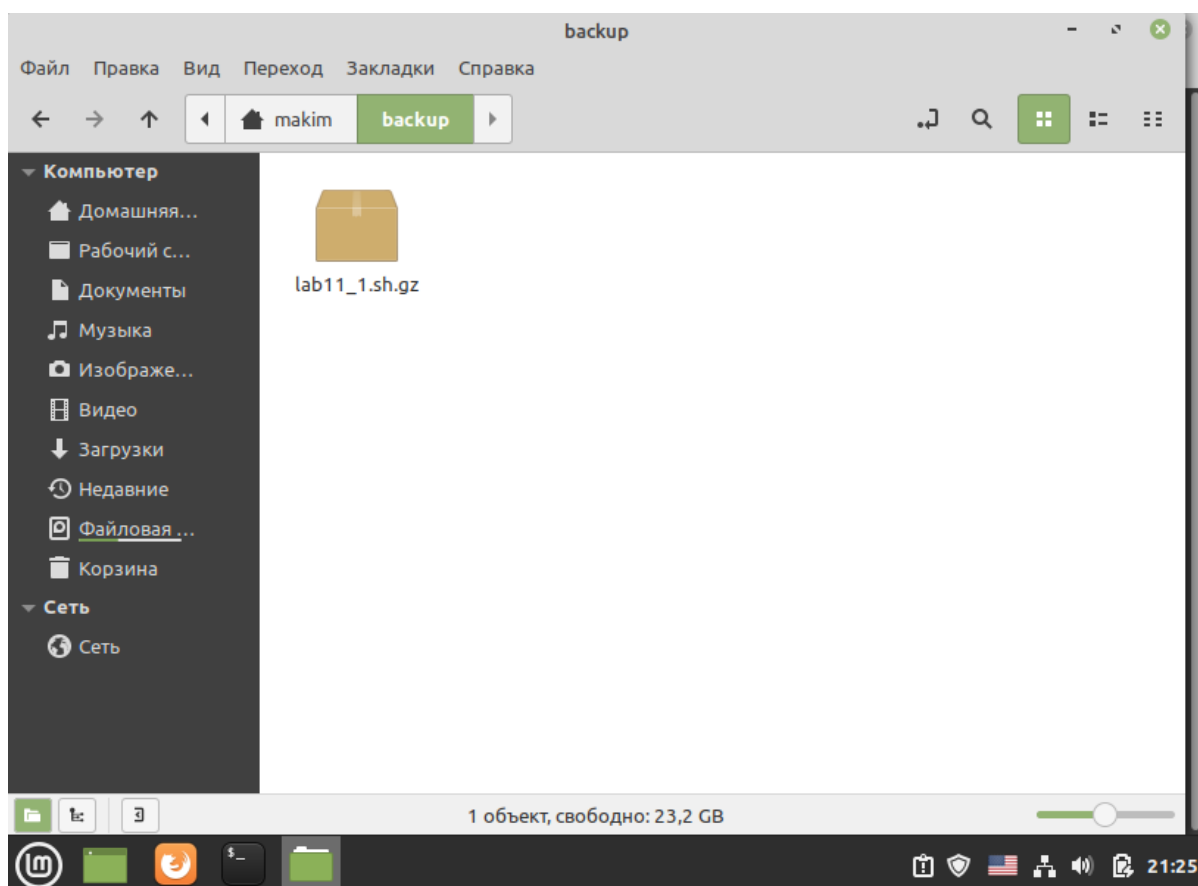


Рис. 2.3: Результат

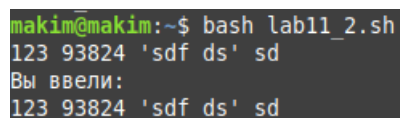
2. Пишем командный файл, который обрабатывает любое произвольное число аргументов командной строки, после чего скрипт последовательно распечатывает значения всех переданных аргументов. (рис. 2.4 - 2.5)

```
// Используем шобанг, чтобы сообщить командной оболочке, каким образом мы бу  
#!/bin/bash
```

```
// создаём массив array, где будут храниться все переданные значения  
declare -a array
```

```
// считываем значения в массив array  
read -a array
```

```
// выводим весь массив array  
echo Вы ввели:  
echo ${array[@]}
```



```
makim@makim:~$ bash lab11_2.sh  
123 93824 'sdf ds' sd  
Вы ввели:  
123 93824 'sdf ds' sd
```

Рис. 2.4: Создание файл

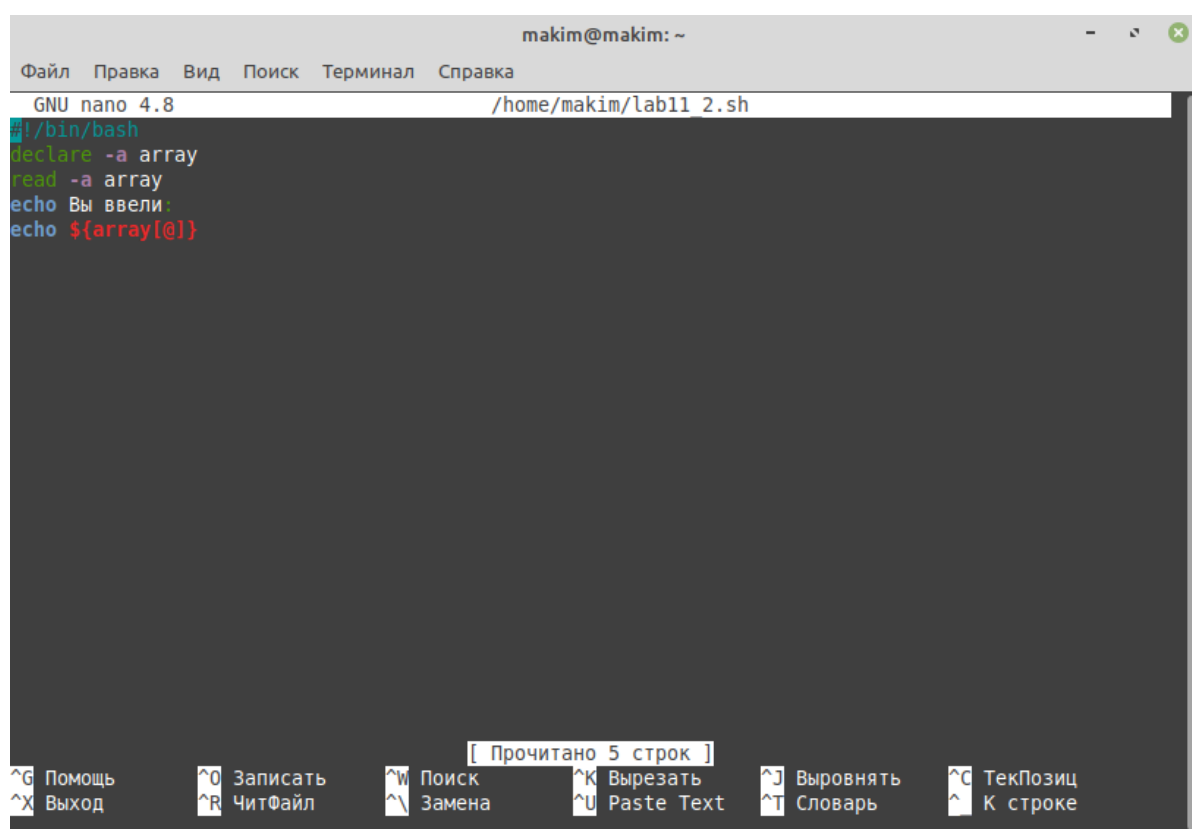


Рис. 2.5: Исходный код командного файла

3. Пишем командный файл — аналог команды `ls` (без использования самой этой команды и команды `dir`). Он выводит информацию о нужном каталоге и выводит информацию о возможностях доступа к файлам этого каталога. (рис. 2.6 - 2.8)

```
#!/bin/bash

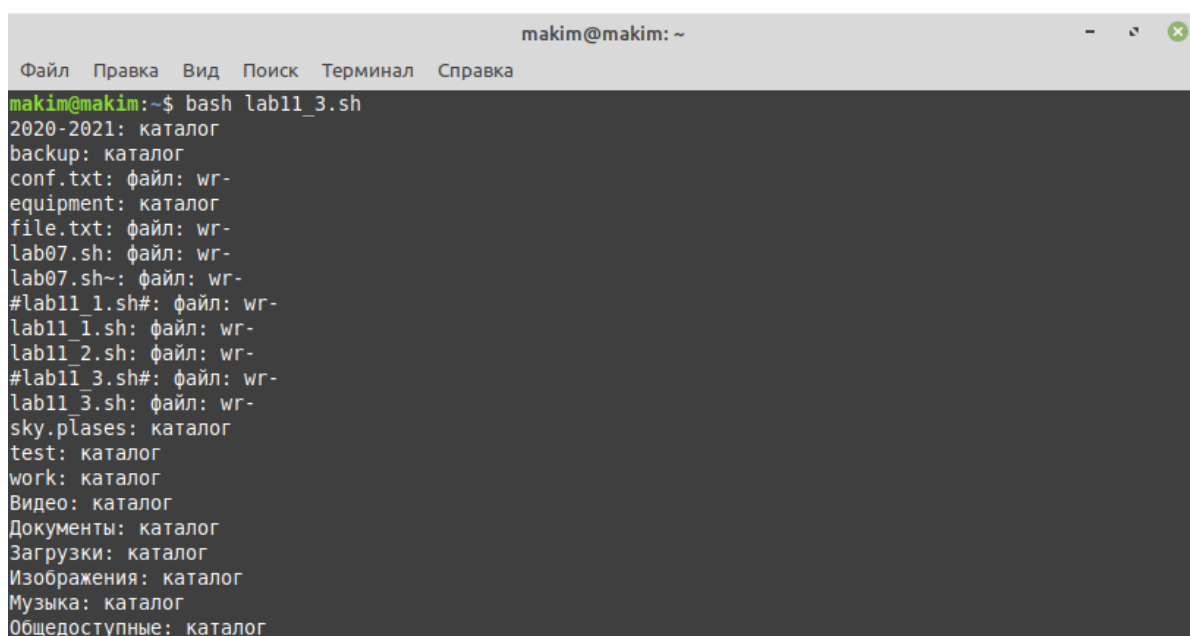
// Запускаем цикл, который проходит по всем элементам каталога.
// В самом цикле делаем ветвления if с командой test в условии.
// Благодаря команде test мы узнаем всю необходимую информацию.
// И в зависимости от этой информации выводим результат.

for i in *
do
if test -d $i; then
    echo $i': 'каталог
else
    echo -n $i': 'файл': '
    if test -w $i; then
        echo -n w
    else echo -n -
    fi
    if test -r $i; then
        echo -n r
    else echo -n -
    fi
    if test -x $i;
    then echo -n x
    else echo -
    fi
fi
done
```




```
1 for i in *
2 do
3 if test -d $i; then
4     echo $i: 'каталог'
5 else
6     echo -n $i: 'файл': '
7     if test -w $i; then
8         echo -n w
9     else echo -n -
10    fi
11    if test -r $i; then
12        echo -n r
13    else echo -n -
14    fi
15    if test -x $i;
16    then echo -n x
17    else echo -
18    fi
19 fi
20 done
```

Рис. 2.6: Исходный код



```
makim@makim:~$ bash lab11_3.sh
2020-2021: каталог
backup: каталог
conf.txt: файл: wr-
equipment: каталог
file.txt: файл: wr-
lab07.sh: файл: wr-
lab07.sh~: файл: wr-
#lab11_1.sh#: файл: wr-
lab11_1.sh: файл: wr-
lab11_2.sh: файл: wr-
#lab11_3.sh#: файл: wr-
lab11_3.sh: файл: wr-
sky.places: каталог
test: каталог
work: каталог
Видео: каталог
Документы: каталог
Загрузки: каталог
Изображения: каталог
Музыка: каталог
Общедоступные: каталог
```

Рис. 2.7: Результат

```

makim@makim:~$ ls -l
итого 88
drwxrwxr-x 3 makim makim 4096 окт 23 02:01 2020-2021
drwxrwxr-x 2 makim makim 4096 окт 24 21:11 backup
-rw-rw-r-- 1 makim makim 526 окт 24 01:20 conf.txt
drwxrwxr-x 2 makim makim 4096 окт 23 23:25 equipment
-rw-rw-r-- 1 makim makim 2585 окт 24 01:12 file.txt
-rw-rw-r-- 1 makim makim 185 окт 24 18:58 lab07.sh
-rw-rw-r-- 1 makim makim 94 окт 24 18:32 lab07.sh~
-rw-rw-r-- 1 makim makim 0 окт 24 21:01 '#lab11_1.sh#'
-rw-rw-r-- 1 makim makim 101 окт 24 21:10 lab11_1.sh
-rw-rw-r-- 1 makim makim 82 окт 24 21:20 lab11_2.sh
-rw-rw-r-- 1 makim makim 259 окт 24 21:44 '#lab11_3.sh#'
-rw-rw-r-- 1 makim makim 261 окт 24 21:47 lab11_3.sh
drwxrwxr-x 4 makim makim 4096 окт 24 16:15 sky.plases
drwxrwxr-x 5 makim makim 4096 окт 24 17:03 test
drwxrwxr-x 3 makim makim 4096 окт 24 17:45 work
drwxr-xr-x 2 makim makim 4096 окт 23 01:48 Видео
drwxr-xr-x 2 makim makim 4096 окт 23 01:48 Документы
drwxr-xr-x 3 makim makim 4096 окт 23 18:09 Загрузки
drwxr-xr-x 2 makim makim 4096 окт 23 01:48 Изображения
drwxr-xr-x 2 makim makim 4096 окт 23 01:48 Музыка
drwxr-xr-x 2 makim makim 4096 окт 23 01:48 Общедоступные

```

Рис. 2.8: Сравнение с ls

4. Пишем командный файл, который получает в качестве аргумента командной строки формат файла (.txt, .doc, .jpg, .pdf и т.д.) и вычисляет количество таких файлов в указанной директории. Путь к директории также передаётся в виде аргумента командной строки.(рис. 2.9 - 2.10)

```
#!/bin/bash
```

```

// Первый аргумент (ему соответствует $1) у нас отвечает за формат (.txt, .d
// Второй аргумент ($2) отвечает за путь, по которому мы будем выводить инфо
// Выводим содержимое директории с помощью ls
// Сортируем содержимое директории с помощью grep
// Подсчитываем с помощью wc
ls -l $2 | grep $1 | wc -l

```

```
1 #!/bin/bash
2 ls -l $2 | grep $1 | wc -l
```

Рис. 2.9: Исходный код

```
makim@makim:~$ bash lab11_4.sh .txt ~
2
makim@makim:~$ ls
2020-2021  file.txt      lab11_1.sh    lab11_4.sh    Видео        Музыка
backup    lab07.sh      lab11_2.sh    sky.plases    Документы    Общедоступные
conf.txt   lab07.sh~     '#lab11 3.sh#' test          Загрузки     'Рабочий стол'
equipment  '#lab11_1.sh#' lab11_3.sh    work          Изображения  Шаблоны
```

Рис. 2.10: Результат

3. Выводы

Мы изучили основы программирования в оболочке ОС UNIX/Linux. Научились писать небольшие командные файлы. А также углубили свое понимание консольных команд для Linux

4. Термины

- Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера.
- POSIX (Portable Operating System Interface for Computer Environments) — набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ.
- Последовательность команд может быть помещена в текстовый файл. Такой файл называется командным.
- Флаги — это опции командной строки, обычно помеченные знаком минус; Например, для команды ls флагом может являться -F.
- Каталог, он же директория, (от английского Directory) – это объект в ФС (файловой системе), необходимый для того, чтобы упростить работу с файлами.
- Домашний каталог - каталог, предназначенный для хранения собственных данных пользователя Linux. Как правило, является текущим непосредственно после регистрации пользователя в системе.
- Команда - записанный по специальным правилам текст (возможно с аргументами), представляющий собой указание на выполнение какой-либо функций (или действий) в операционной системе.