

Презентация к лабораторной работе №11

Тема:

Программирование в командном процессоре ОС UNIX. Командные файлы

Российский Университет Дружбы Народов

Факультет Физико-Математических и Естественных Наук

Дисциплина: *Операционные системы*

Студент: Мухамедияр Адиль

Группа: НКНбд-01-20

Москва, 2021г.

Цель работы

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

Введение

Командные процессоры или оболочки – это программы, позволяющие пользователю взаимодействовать с компьютером. Их можно рассматривать как настоящие интерпретируемые языки, которые воспринимают команды пользователя и обрабатывают их. Поэтому командные процессоры также называют интерпретаторами команд. На языках оболочек можно писать программы и выполнять их подобно любым другим программам. UNIX обладает большим количеством оболочек. Наиболее популярными являются следующие четыре оболочки:

- оболочка Борна (Bourne) – первоначальная командная оболочка UNIX: базовый, но полный набор функций;
- C-оболочка – добавка университета Беркли к коллекции оболочек: она надстраивается над оболочкой Борна, используя C-подобный синтаксис команд, и сохраняет историю выполненных команд;
- оболочка Корна – напоминает оболочку C, но операторы управления программой совместимы с операторами оболочки Борна;
- BASH – сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей совмещает свойства оболочек C и Корна (разработка компании Free Software Foundation).

Последовательность выполнения работы

1. Написать 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 должен архивироваться одним из архиваторов на выбор zip, bzip2 или tar. Способ использования команд архивации необходимо узнать, изучив справку.
2. Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов.
3. Написать 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.

Ход работы:

1. Изучил опции команды, и саму команду *gzip*, с помощью команды *man*.

Файл Правка Вид Поиск Терминал Справка

NAME

gzip, gunzip, zcat - compress or expand files

SYNOPSIS

```
gzip [ -acdfhLLnNrtvV19 ] [-S suffix] [ name ... ]
gunzip [ -acfhlLnNrtvV ] [-S suffix] [ name ... ]
zcat [ -fhLV ] [ name ... ]
```

DESCRIPTION

Gzip reduces the size of the named files using Lempel-Ziv coding (LZ77). Whenever possible, each file is replaced by one with the extension **.gz**, while keeping the same ownership modes, access and modification times. (The default extension is **-gz** for VMS, **z** for MSDOS, OS/2 FAT, Windows NT FAT and Atari.) If no files are specified, or if a file name is "-", the standard input is compressed to the standard output. Gzip will only attempt to compress regular files. In particular, it will ignore symbolic links.

If the compressed file name is too long for its file system, gzip truncates it. Gzip attempts to truncate only the parts of the file name longer than 3 characters. (A part is delimited by dots.) If the name consists of small parts only, the longest parts are truncated. For example, if file names are limited to 14 characters, `gzip.msdos.exe` is compressed to `gzi.msdx.exe.gz`. Names are not truncated on systems which do not have a limit on file name length.

Manual page gzip(1) line 5 (press h for help or q to quit)

- Создал текстовой файл *lab11.sh*, используя команду *touch*. Открываю текстовый редактор *emacs*.

```
[amukhamediyar@amukhamediyar ~]$ man gzip
[amukhamediyar@amukhamediyar ~]$ touch lab11.sh
[amukhamediyar@amukhamediyar ~]$ emacs
```

- Написал код, в котором я создаю папку *backup*, копировал текстовой файл *lab11.sh*, указав путь для сохранения файла. После архивировал данный файл через команду *gzip*.

File Edit Options Buffers Tools Sh-Script Help



Save



Undo



#!/bin/bash

mkdir ~/backup

cp lab11.sh ~/backup/backup.sh

gzip ~/backup/backup.sh

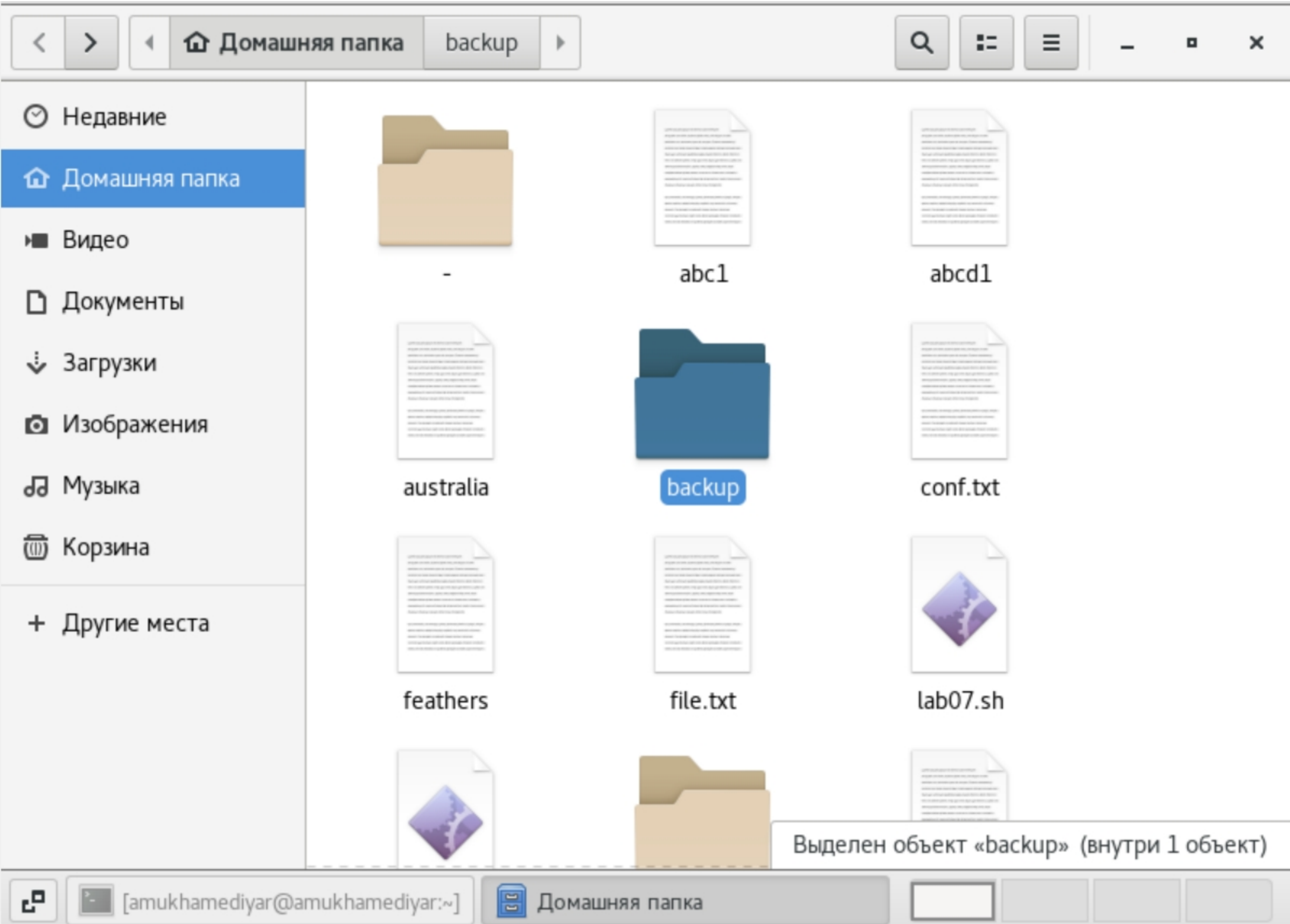
- Командой *chmod* я сделал так, чтобы файл был исполняемым в **Linux**. Следующая строка для того, чтобы он выполнил нашу ранее написанную программу.

[amukhamediyar@amukhamediyar ~]\$ chmod +x lab11.sh

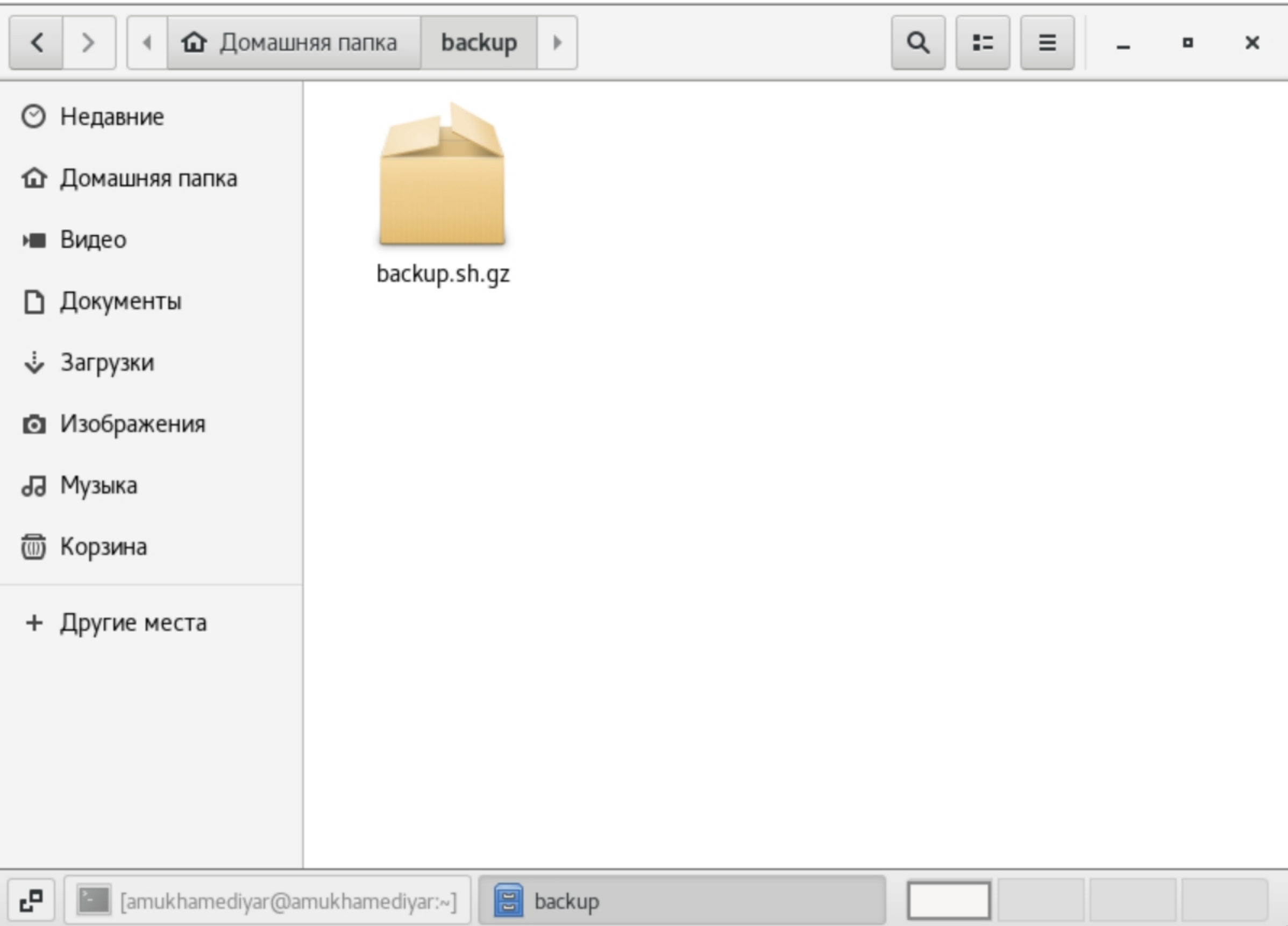
[amukhamediyar@amukhamediyar ~]\$./lab11.sh

[amukhamediyar@amukhamediyar ~]\$

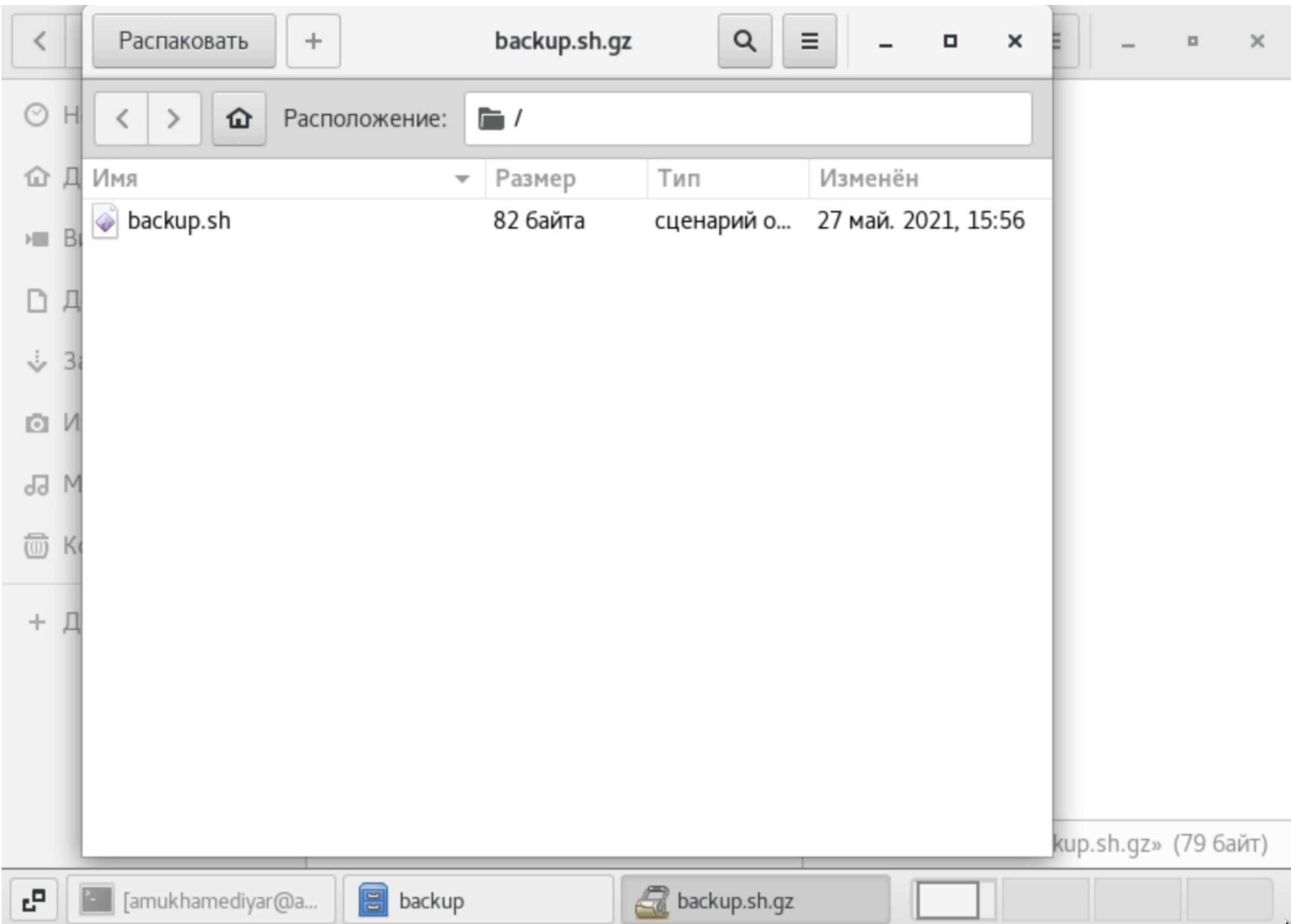
- Проверяю, смог ли выполнить мою программу. Папка с именем *backup* была создана. Видим, папка с именем *backup.sh* был архивирован, теперь называется *backup.sh.gz*.



- Проверяю, смог ли выполнить мою программу.



- Проверяю, смог ли выполнить мою программу. Внутри папки *backup.sh.gz* находится наш скопированный файл *lab11.sh*, который был переименован в *backup.sh*.



2. Создал текстовой файл *lab11_2.sh*(для выполнения 2 пункта данной лабораторной работы), используя команду*touch*. Открываю текстовый редактор *emacs*.

```
[amukhamediyar@amukhamediyar ~]$ touch lab11_2.sh
[amukhamediyar@amukhamediyar ~]$ emacs
```

- Написал программу, для вывода того, что я буду вводить. Для начало надо ввести числа, я решил ввести нечетные числа, как видим, выводит так же нечетные, при это те же самые, которые были введены.

File Edit Options Buffers Tools Sh-Script Help



Save



Undo



```
#!/bin/bash
echo "Vvedite chisla:"
head -1
```

- Командой *chmod* я сделал так, чтобы файл был исполняемым в **Linux**. Следующая строка для того, чтобы он выполнил нашу ранее написанную программу.

```
[amukhamediyar@amukhamediyar ~]$ chmod +x lab11_2.sh
[amukhamediyar@amukhamediyar ~]$ ./lab11_2.sh
Vvedite chisla:
1 3 5 7 9 11
1 3 5 7 9 11
[amukhamediyar@amukhamediyar ~]$
```

3. Создал текстовой файл *lab11_3.sh*(для выполнения 2 пункта данной лабораторной работы), используя команду *touch*. Открываю текстовый редактор *emacs*.

```
[amukhamediyar@amukhamediyar ~]$ touch lab11_3.sh
[amukhamediyar@amukhamediyar ~]$ emacs
```

- Написал командный файл — аналог команды *ls* (без использования самой этой команды и команды *dir*).

File Edit Options Buffers Tools Sh-Script Help



Save



Undo



```
#!/bin/bash
for i in *
do if test -d $i
  then echo $i: is a directory
  else echo -n $i: is a file
    if test -w $i
    then echo avaiable for writing
    elif test -r $i
    then echo readable
    else echo neather for readable nor writeable
    fi
  fi
done
```

- Он выдавал информацию о нужном каталоге и выводил информацию о возможностях доступа к файлам этого каталога.


```
[amukhamediyar@amukhamediyar ~]$ chmod +x lab11_3.sh
[amukhamediyar@amukhamediyar ~]$ ./lab11_3.sh
-: is a directory
abc1: is a fileavailable for writing
abcd1: is a fileavailable for writing
australia: is a fileavailable for writing
backup: is a directory
conf.txt: is a fileavailable for writing
feathers: is a fileavailable for writing
file.txt: is a fileavailable for writing
#lab#: is a fileavailable for writing
#lab07.sh#: is a fileavailable for writing
lab07.sh: is a fileavailable for writing
lab_08: is a directory
lab11_2.sh: is a fileavailable for writing
lab11_2.sh~: is a fileavailable for writing
lab11_3.sh: is a fileavailable for writing
lab11_3.sh~: is a fileavailable for writing
lab11.sh: is a fileavailable for writing
lab11.sh~: is a fileavailable for writing
may: is a fileavailable for writing
monthly: is a directory
my_os: is a fileavailable for writing
play: is a fileavailable for writing
reports: is a directory
ski.places: is a directory
```

4. Создал текстовой файл *lab11_4.sh*(для выполнения 2 пункта данной лабораторной работы), используя команду *touch*. Открываю текстовый редактор *emacs*.

```
[amukhamediyar@amukhamediyar ~]$ touch lab11_4.sh
[amukhamediyar@amukhamediyar ~]$ emacs
```

- Написал командный файл, который получает в качестве аргумента командной строки формат файла (.txt, .doc, .jpg, pdf и т.д.).

lab11_4.sh - emacs@amukhamediyar.localdomain

File Edit Options Buffers Tools Sh-Script Help

```
#!/bin/bash
format=""
direct=""
echo "write format"
read format
echo "write direct"
read direct
find "$direct" - name ".$format" -type f | wc -l
ls
```

- Вычисляет количество таких файлов в указанной директории. Путь к директории также передаётся в виде аргумента командной строки.

```
[amukhamediyar@amukhamediyar ~]$ chmod +x lab11_4.sh
[amukhamediyar@amukhamediyar ~]$ ./lab11_4.sh
write format
.sh
write direct
home/amukhamediyar
find: 'home/amukhamediyar': Нет такого файла или каталога
find: 'name': Нет такого файла или каталога
find: '..sh': Нет такого файла или каталога
28
-          feathers      lab11_2.sh  lab11.sh   reports    work        Общедоступные
abc1       file.txt        lab11_2.sh~ lab11.sh~  ski.places Видео        Рабочий стол
abcd1     #lab#             lab11_3.sh  may        test        Документы     Шаблоны
australia #lab07.sh#       lab11_3.sh~ monthly    testlab     Загрузки
backup    lab07.sh          lab11_4.sh  my_os      text.txt    Изображения
conf.txt  lab_08            lab11_4.sh~ play       tutorial     Музыка
[amukhamediyar@amukhamediyar ~]$
```

Изучил основы программирования в оболочке ОС UNIX/Linux, научился писать небольшие командные файлы.

Библиография

[Командные файлы Linux](#) [Командные процессоры \(оболочки\)](#)
