

# **Отчёт по лабораторной работе №11**

Серегин Денис Алексеевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задания</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выводы</b>	<b>13</b>
<b>5</b>	<b>Использованные ресурсы</b>	<b>14</b>

# List of Tables

# List of Figures

3.1	Создание файл . . . . .	7
3.2	Код скрипта . . . . .	7
3.3	Создание исполняемого файла . . . . .	8
3.4	Получившаяся резервная копия . . . . .	8
3.5	Создание файла . . . . .	8
3.6	Код командного файла . . . . .	9
3.7	Создание исполняемого файла и проверка его работы . . . . .	9
3.8	Создание файла . . . . .	9
3.9	Код программы . . . . .	10
3.10	Создание исполняемого файла . . . . .	10
3.11	Результат работы кода . . . . .	11
3.12	Создание файла . . . . .	11
3.13	Код программы . . . . .	12
3.14	Создание исполняемого файла . . . . .	12
3.15	Результат работы кода . . . . .	12

# 1 Цель работы

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

## 2 Задания

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.
4. Написать 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.

### 3 Выполнение лабораторной работы

1. Написал скрипт, который при запуске делает резервную копию самого себя в другую директорию backup в своём домашнем каталоге. Файл архивировался предустановленным в системе архиватором gzip. Способ использования команд архивации узнал, изучив справку.

(рис 1. 3.1)

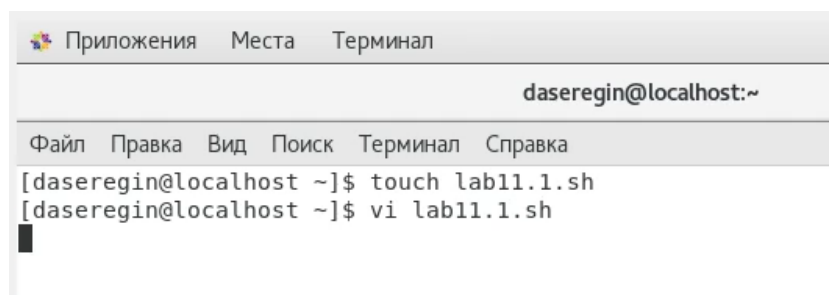


Figure 3.1: Создание файл

(рис 2. 3.2)

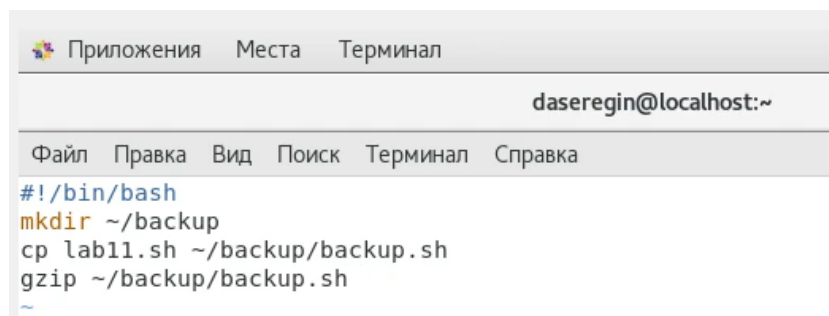
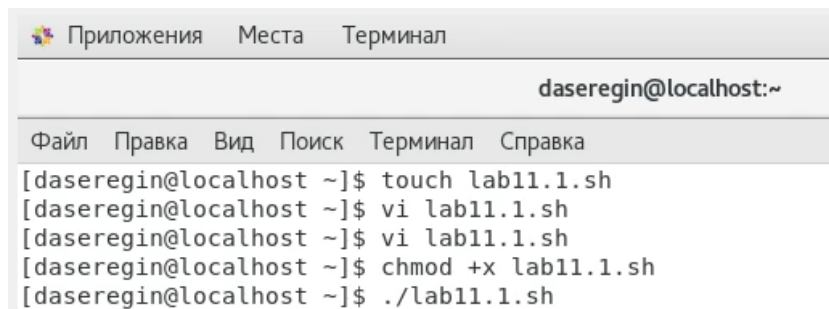


Figure 3.2: Код скрипта

(рис 3. 3.3)



```
daseregin@localhost:~  
[daseregin@localhost ~]$ touch lab11.1.sh  
[daseregin@localhost ~]$ vi lab11.1.sh  
[daseregin@localhost ~]$ vi lab11.1.sh  
[daseregin@localhost ~]$ chmod +x lab11.1.sh  
[daseregin@localhost ~]$ ./lab11.1.sh
```

Figure 3.3: Создание исполняемого файла

(рис 4. 3.4)

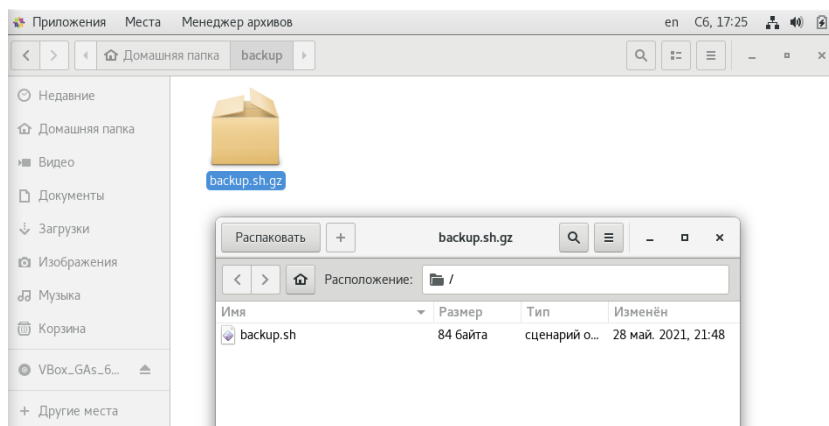
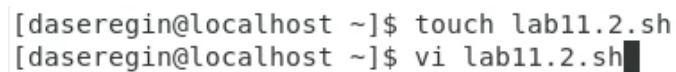


Figure 3.4: Получившаяся резервная копия

2. Написал пример командного файла, обрабатывающего любое произвольное число аргументов, в том числе превышающее десять. Скрипт последовательно распечатывал значения всех переданных аргументов.

(рис 5. 3.5)



```
[daseregin@localhost ~]$ touch lab11.2.sh  
[daseregin@localhost ~]$ vi lab11.2.sh
```

Figure 3.5: Создание файла

(рис 6. 3.6)



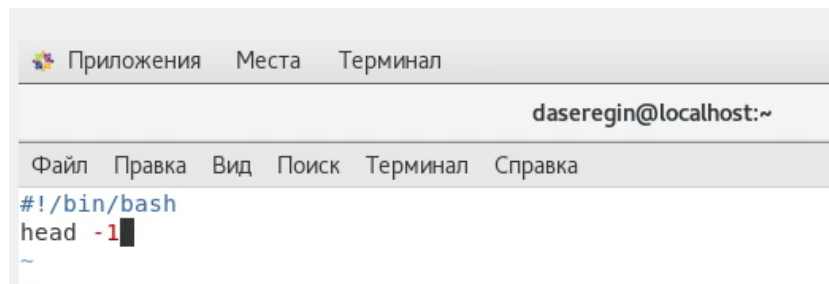


Figure 3.6: Код командного файла

(рис 7. 3.7)

```
[daseregin@localhost ~]$ chmod +x lab11.2.sh
[daseregin@localhost ~]$ ./lab11.2.sh
1 22 333 28
1 22 333 28
```

Figure 3.7: Создание исполняемого файла и проверка его работы

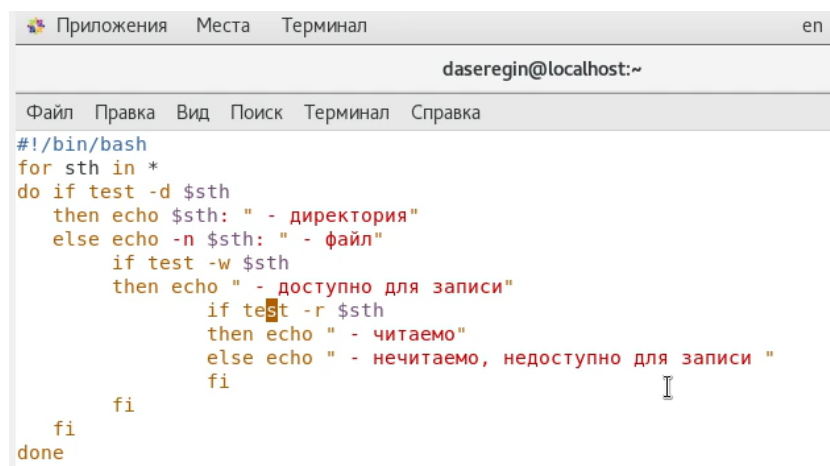
3. Написал командный файл - аналог команды ls. Файл выдаёт информацию о нужном каталоге и выводит информацию о возможностях доступа к файлам этого каталога.

(рис 8. 3.8)

```
[daseregin@localhost ~]$ touch lab11.3.sh
[daseregin@localhost ~]$ vi lab11.3.sh
```

Figure 3.8: Создание файла

(рис 9. 3.9)

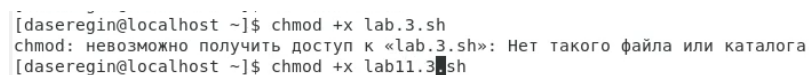


The screenshot shows a terminal window with a menu bar at the top containing 'Приложения', 'Места', 'Терминал', and 'en'. Below the menu bar is a title bar with 'daseregin@localhost:~'. The terminal content displays a shell script with the following code:

```
#!/bin/bash
for sth in *
do if test -d $sth
  then echo $sth: " - директория"
  else echo -n $sth: " - файл"
    if test -w $sth
    then echo " - доступно для записи"
    if test -r $sth
    then echo " - читаемо"
    else echo " - нечитаемо, недоступно для записи "
    fi
  fi
done
```

Figure 3.9: Код программы

(рис 10. 3.10)

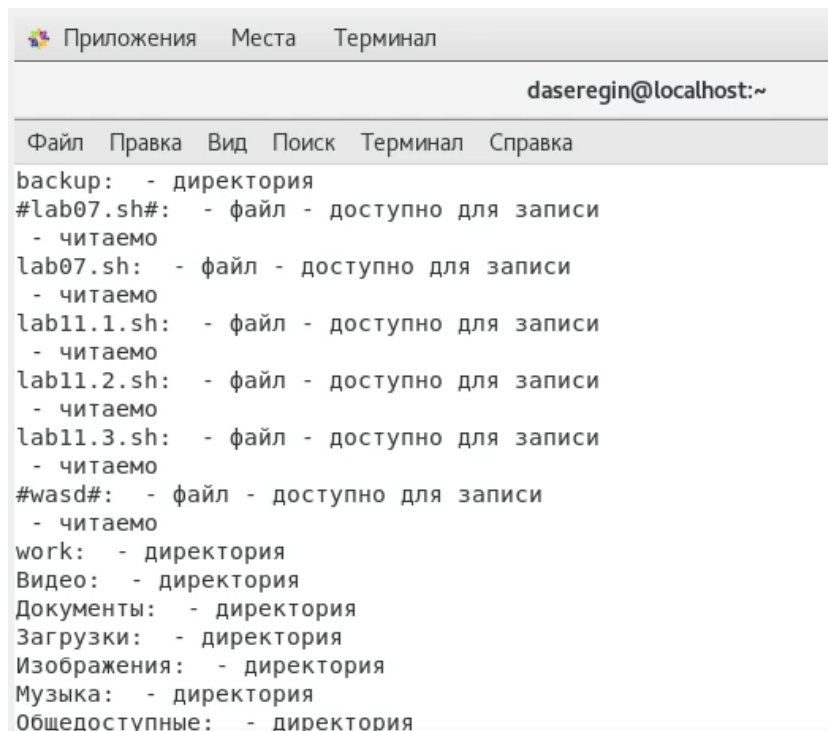


The screenshot shows a terminal window with the following commands and output:

```
[daseregin@localhost ~]$ chmod +x lab.3.sh
chmod: невозможно получить доступ к «lab.3.sh»: Нет такого файла или каталога
[daseregin@localhost ~]$ chmod +x lab11.3.sh
```

Figure 3.10: Создание исполняемого файла

(рис 11. 3.11)

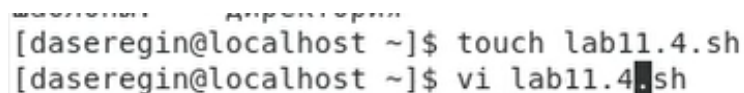


```
Приложения Места Терминал
daseregin@localhost:~
Файл Правка Вид Поиск Терминал Справка
backup: - директория
#lab07.sh#: - файл - доступно для записи
- читаемо
lab07.sh: - файл - доступно для записи
- читаемо
lab11.1.sh: - файл - доступно для записи
- читаемо
lab11.2.sh: - файл - доступно для записи
- читаемо
lab11.3.sh: - файл - доступно для записи
- читаемо
#wasd#: - файл - доступно для записи
- читаемо
work: - директория
Видео: - директория
Документы: - директория
Загрузки: - директория
Изображения: - директория
Музыка: - директория
Общедоступные: - директория
```

Figure 3.11: Результат работы кода

4. Написал командный файл который получает в качестве аргумента командной строки формат файла и вычисляет количество таких файлов в указанной директории. Путь к директории также передаётся в виде аргумента командной строки.

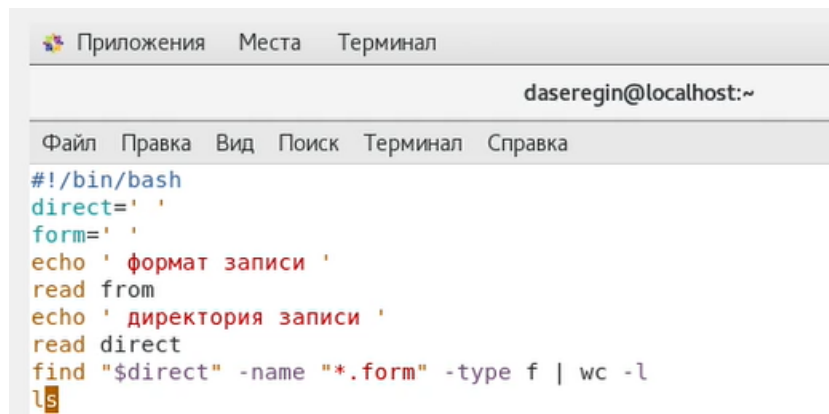
(рис 12. 3.12)



```
[daseregin@localhost ~]$ touch lab11.4.sh
[daseregin@localhost ~]$ vi lab11.4.sh
```

Figure 3.12: Создание файла

(рис 13. 3.13)



```
#!/bin/bash
direct=' '
form=' '
echo ' формат записи '
read form
echo ' директория записи '
read direct
find "$direct" -name "*.form" -type f | wc -l
ls
```

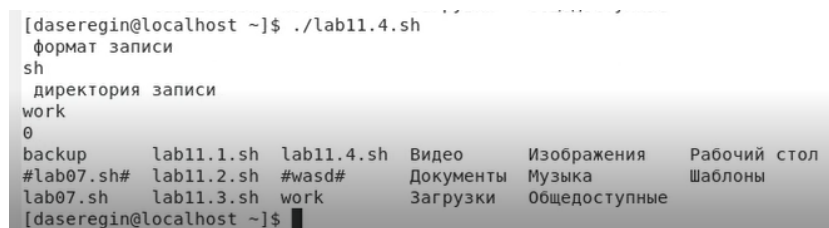
Figure 3.13: Код программы

(рис 14. 3.14)

```
[daseregin@localhost ~]$ chmod +x lab11.4.sh
[daseregin@localhost ~]$ ./lab11.4.sh
```

Figure 3.14: Создание исполняемого файла

(рис 15. 3.15)



```
[daseregin@localhost ~]$ ./lab11.4.sh
формат записи
sh
директория записи
work
0
backup      lab11.1.sh  lab11.4.sh  Видео      Изображения  Рабочий стол
#lab07.sh#  lab11.2.sh  #wasd#      Документы  Музыка       Шаблоны
lab07.sh    lab11.3.sh  work        Загрузки   Общедоступные
[daseregin@localhost ~]$
```

Figure 3.15: Результат работы кода

## **4 Выводы**

В ходе лабораторной работы мне удалось изучить основы программирования в оболочке ОС UNIX/LINUX, а также научиться писать небольшие командные файлы.

## **5 Используемые ресурсы**

<https://vps.ua/wiki/creating-unpacking-files-linux/>

<https://habr.com/ru/company/ruvds/blog/325522/>