

Отчёт по лабораторной работе №11

**Программирование в командном процессоре ОС UNIX. Ветвления и
циклы**

Фарида Гулиева НКНбд 01-21

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	9
4	Контрольные вопросы	10

List of Figures

2.1	Задание 1	5
2.2	Задание 2	6
2.3	Задание 3	7
2.4	Задание 4	8

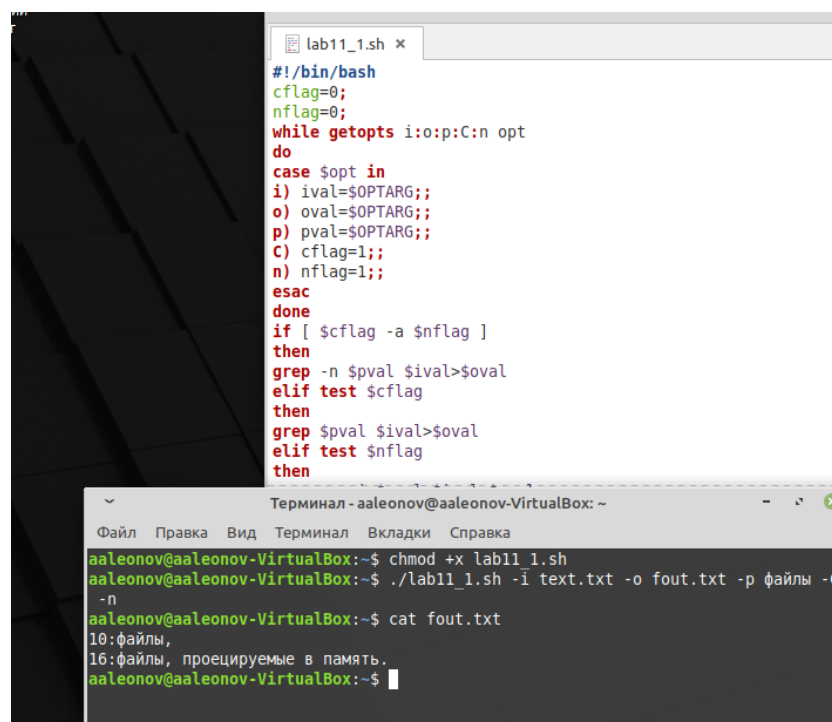
1 Цель работы

Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

2 Выполнение лабораторной работы

1. Используя команды `getopts` `grep` напишем командный файл, который анализирует командную строку с ключами и выполним его: `-i inputfile` — прочитать данные из указанного файла; `-o outputfile` — вывести данные в указанный файл; `-p шаблон` — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк;

а затем ищет в указанном файле нужные строки



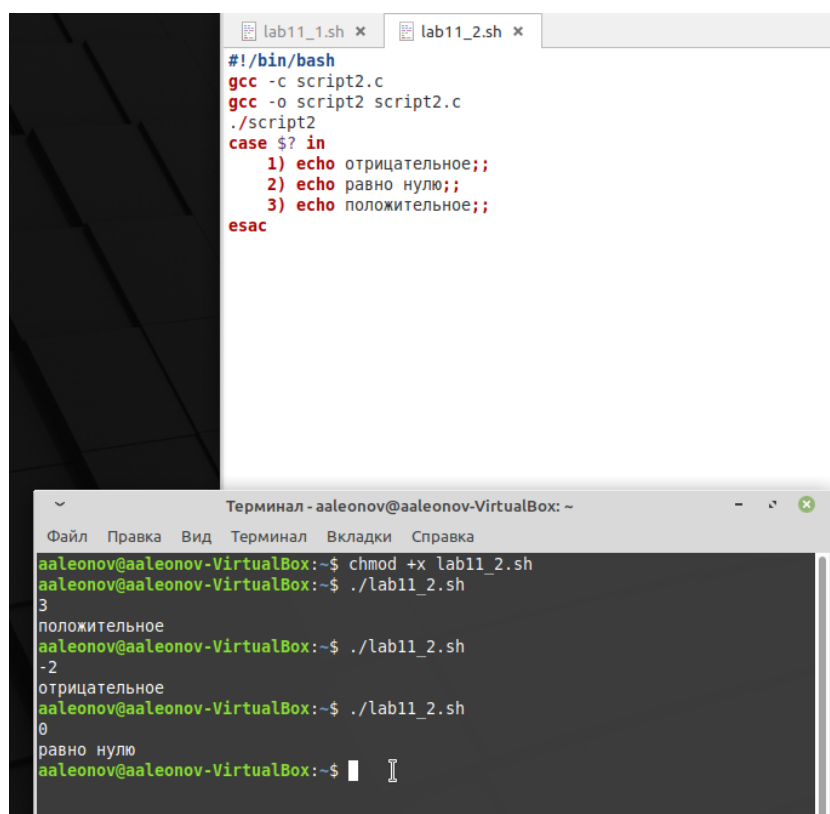
```
#!/bin/bash
cflag=0;
nflag=0;
while getopts i:o:p:C:n opt
do
case $opt in
i) ival=$OPTARG;;
o) oval=$OPTARG;;
p) pval=$OPTARG;;
C) cflag=1;;
n) nflag=1;;
esac
done
if [ $cflag -a $nflag ]
then
grep -n $pval $ival>$oval
elif test $cflag
then
grep $pval $ival>$oval
elif test $nflag
then
```

```
Терминал - aaleonov@aaleonov-VirtualBox: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка
aaleonov@aaleonov-VirtualBox:~$ chmod +x lab11 1.sh
aaleonov@aaleonov-VirtualBox:~$ ./lab11_1.sh -i text.txt -o fout.txt -p файлы -C
-n
aaleonov@aaleonov-VirtualBox:~$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
aaleonov@aaleonov-VirtualBox:~$
```

Figure 2.1: Задание 1

2. Напишем сначала на языке Си программу, которая вводит число и опре-

деляет, является ли оно больше нуля, меньше нуля или равно нулю. Затем завершим программу при помощи функции `exit(n)`, передавая информацию о коде завершения в оболочку. Командный файл вызовет эту программу и, проанализировав с помощью команды `$?`, выдаст сообщение о том, какое число было введено



```
#!/bin/bash
gcc -c script2.c
gcc -o script2 script2.c
./script2
case $? in
  1) echo отрицательное;;
  2) echo равно нулю;;
  3) echo положительное;;
esac
```

```
Терминал - aaleonov@aaleonov-VirtualBox: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка
aaleonov@aaleonov-VirtualBox:~$ chmod +x lab11_2.sh
aaleonov@aaleonov-VirtualBox:~$ ./lab11_2.sh
3
положительное
aaleonov@aaleonov-VirtualBox:~$ ./lab11_2.sh
-2
отрицательное
aaleonov@aaleonov-VirtualBox:~$ ./lab11_2.sh
0
равно нулю
aaleonov@aaleonov-VirtualBox:~$
```

Figure 2.2: Задание 2

3. Напишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N

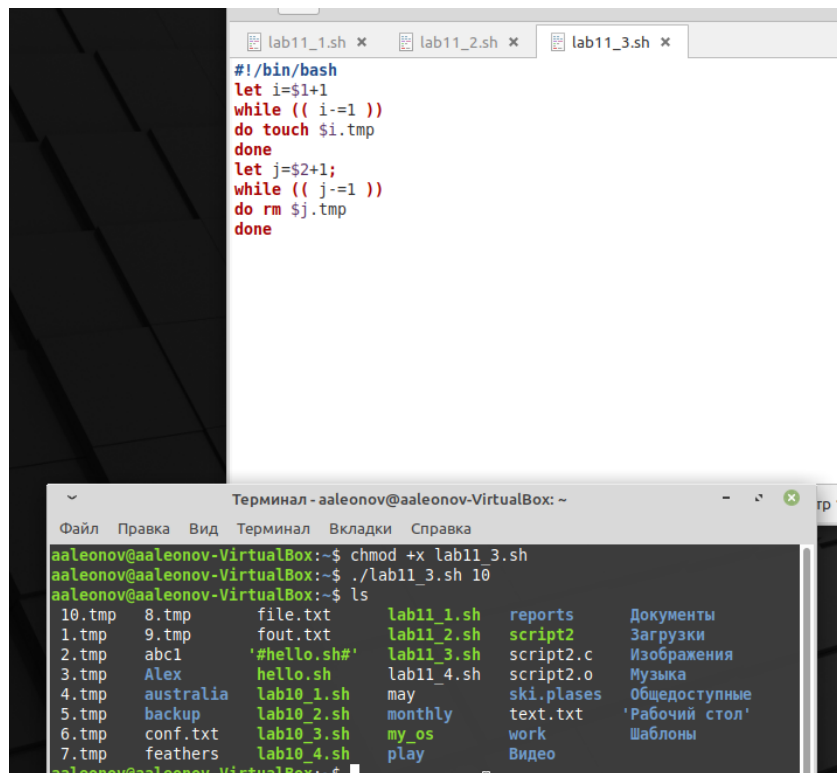
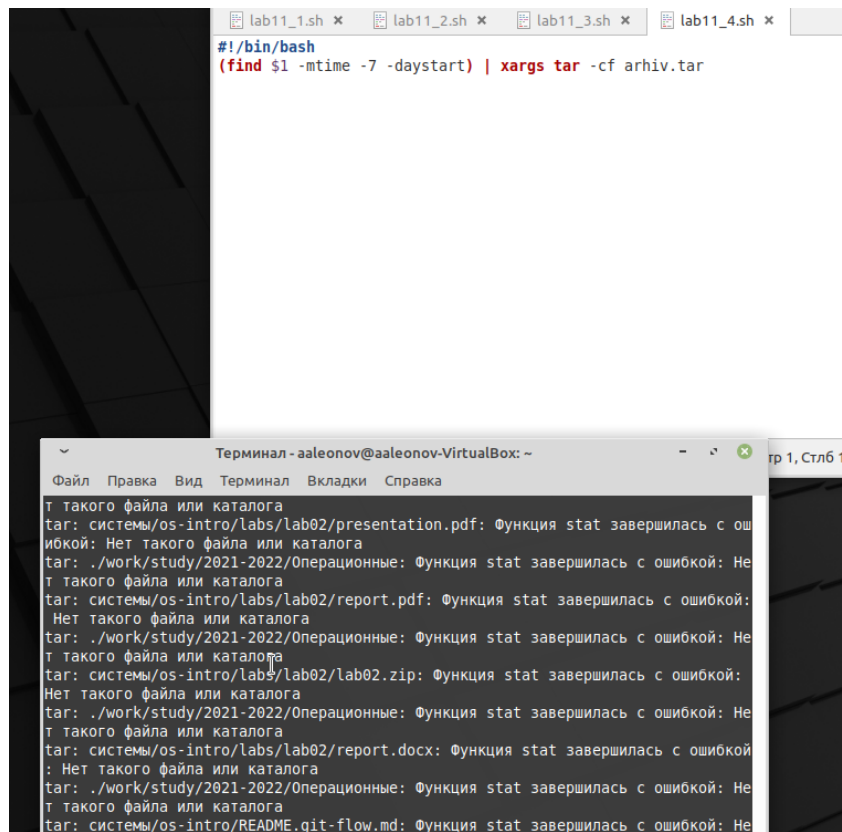


Figure 2.3: Задание 3

4. Напишем командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории. Модифицируем его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад.



The image shows a terminal window with a dark background. At the top, there are four tabs labeled 'lab11_1.sh', 'lab11_2.sh', 'lab11_3.sh', and 'lab11_4.sh'. The active tab is 'lab11_4.sh'. The terminal prompt is '#!/bin/bash'. The command being executed is '(find \$1 -mtime -7 -daystart) | xargs tar -cf arhiv.tar'. The output of the command is a list of files and directories that have been archived, including 'системы/os-intro/labs/lab02/presentation.pdf', 'системы/os-intro/labs/lab02/report.pdf', 'системы/os-intro/labs/lab02/lab02.zip', 'системы/os-intro/labs/lab02/report.docx', and 'системы/os-intro/README.git-flow.md'. The output is displayed in a light-colored font on a dark background.

```
#!/bin/bash
(find $1 -mtime -7 -daystart) | xargs tar -cf arhiv.tar

Т такого файла или каталога
tar: системы/os-intro/labs/lab02/presentation.pdf: Функция stat завершилась с ошибкой: Нет такого файла или каталога
tar: ./work/study/2021-2022/Операционные: Функция stat завершилась с ошибкой: Нет такого файла или каталога
tar: системы/os-intro/labs/lab02/report.pdf: Функция stat завершилась с ошибкой: Нет такого файла или каталога
tar: ./work/study/2021-2022/Операционные: Функция stat завершилась с ошибкой: Нет такого файла или каталога
tar: системы/os-intro/labs/lab02/lab02.zip: Функция stat завершилась с ошибкой: Нет такого файла или каталога
tar: ./work/study/2021-2022/Операционные: Функция stat завершилась с ошибкой: Нет такого файла или каталога
tar: системы/os-intro/labs/lab02/report.docx: Функция stat завершилась с ошибкой: Нет такого файла или каталога
tar: ./work/study/2021-2022/Операционные: Функция stat завершилась с ошибкой: Нет такого файла или каталога
tar: системы/os-intro/README.git-flow.md: Функция stat завершилась с ошибкой: Нет
```

Figure 2.4: Задание 4

3 Вывод

В данной работе мы изучили основы программирования в оболочке ОС UNIX и писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

4 Контрольные вопросы

1. Каково предназначение команды `getopts`? Ответ: Создание по пользовательским аргументам.
2. Какое отношение метасимволы имеют к генерации имён файлов? Ответ: Используют как файлы так и аргументы.
3. Какие операторы управления действиями вы знаете? Ответ: `if`, `else`, `elif`, `fi`, `while`, `do`, `done`, `until`, `do`, `done`, `for`, `in`, `do`, `done`, `case`, `in`, `esac`
4. Какие операторы используются для прерывания цикла? Ответ:
 - a) `for` – будет выполнять действие до тех пор, пока есть объекты для выполнения.
 - b) `while` – выполняет действие до тех пор, пока условие является истинным.
 - c) `until` – будет выполняться пока условие не станет правдиво.
5. Для чего нужны команды `false` и `true`? Ответ: `until` – будет выполняться до тех пор, пока условие не станет `true`, т.е. пока оно не станет `false`.
6. Что означает строка `if test -f mans/i.$s`, встреченная в командном файле? Ответ: Проверяет если существует файл его размерность и тип с двумя разными расширениями, заменяя через переменные.

7. Объясните различия между конструкциями while и until. Ответ:

while – выполняет действие до тех пор, пока условие является истинным.

until – будет выполняться до тех пор, пока условие не станет истинным, т.е. пока оно false.