

# **Отчёт по лабораторной работе №11**

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

Алламе Ормиз НФИбд-01-21

# Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	8
4	Контрольные вопросы	9

# List of Figures

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

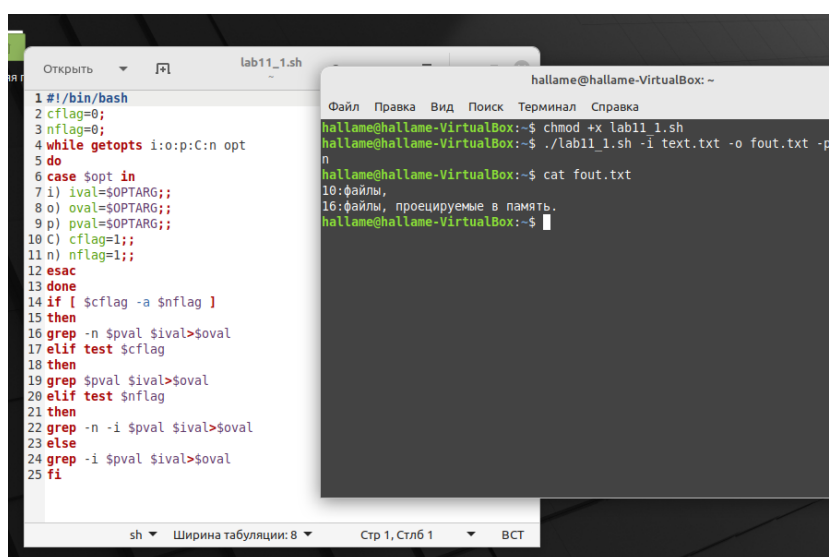
# 1 Цель работы

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; `-r шаблон` — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк;

а затем ищет в указанном файле нужные строки



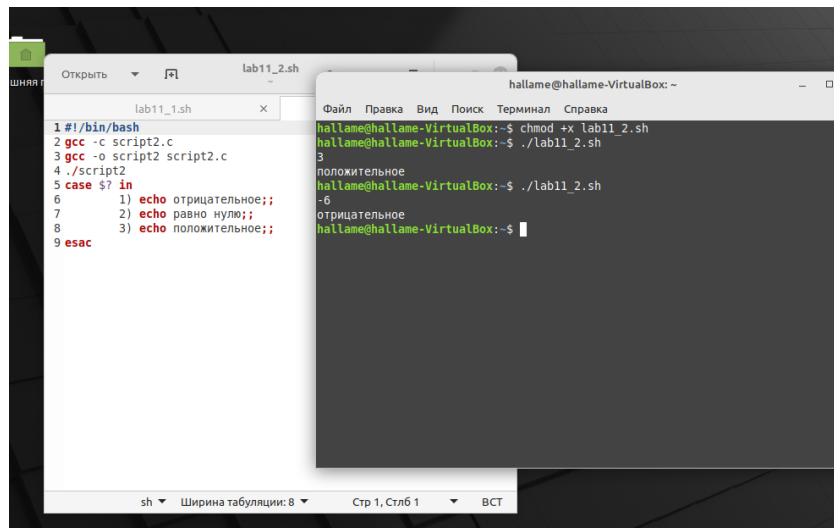
```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:o:p:C:n opt
5 do
6 case $opt in
7 i) ival=$OPTARG;;
8 o) oval=$OPTARG;;
9 p) pval=$OPTARG;;
10 C) cflag=1;;
11 n) nflag=1;;
12 esac
13 done
14 if [ $cflag -a $nflag ]
15 then
16 grep -n $pval $ival>$oval
17 elif test $cflag
18 then
19 grep $pval $ival>$oval
20 elif test $nflag
21 then
22 grep -n -i $pval $ival>$oval
23 else
24 grep -i $pval $ival>$oval
25 fi
```

```
hallame@hallame-VirtualBox:~$ chmod +x lab11_1.sh
hallame@hallame-VirtualBox:~$ ./lab11_1.sh -i text.txt -o fout.txt -p d
n
hallame@hallame-VirtualBox:~$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
```

Figure 2.1: Задание 1

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

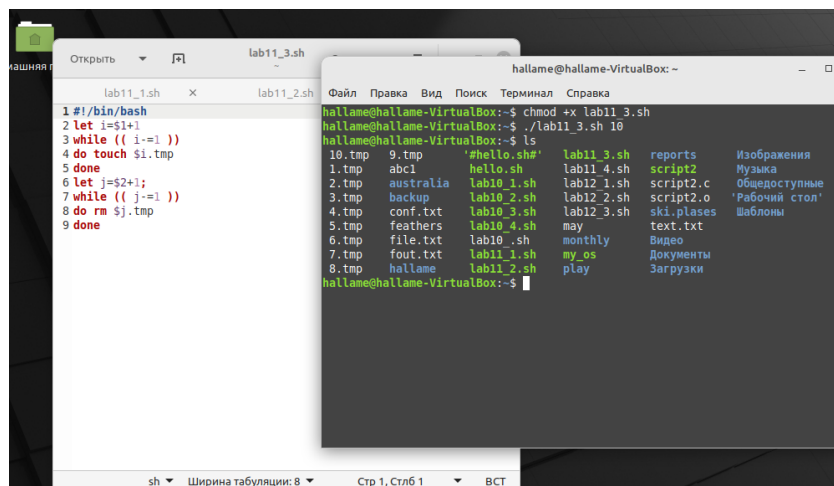


```
lab11_1.sh
1 #!/bin/bash
2 gcc -c script2.c
3 gcc -o script2 script2.c
4 ./script2
5 case $? in
6   1) echo отрицательное;;
7   2) echo равно нулю;;
8   3) echo положительное;;
9 esac

hallame@hallame-VirtualBox: ~
hallame@hallame-VirtualBox:~$ chmod +x lab11_2.sh
hallame@hallame-VirtualBox:~$ ./lab11_2.sh
3
положительное
hallame@hallame-VirtualBox:~$ ./lab11_2.sh
-6
отрицательное
hallame@hallame-VirtualBox:~$
```

Figure 2.2: Задание 2

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



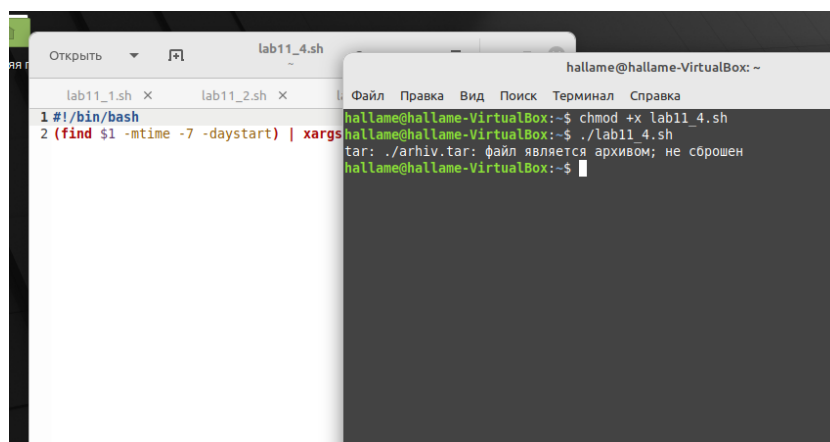
```
lab11_1.sh
1 #!/bin/bash
2 let i=$1+1
3 while (( i=1 ))
4 do touch $i.tmp
5 done
6 let j=$2+1
7 while (( j=1 ))
8 do rm $j.tmp
9 done

hallame@hallame-VirtualBox: ~
hallame@hallame-VirtualBox:~$ chmod +x lab11_3.sh
hallame@hallame-VirtualBox:~$ ./lab11_3.sh 10
hallame@hallame-VirtualBox:~$ ls
10.tmp 9.tmp 'hello.sh' lab11_3.sh reports Изображения
1.tmp abc1 hello.sh lab11_4.sh script2 Музыка
2.tmp australia lab10_1.sh lab12_1.sh script2.c Общедоступные
3.tmp backup lab10_2.sh lab12_2.sh script2.o 'Рабочий стол'
4.tmp conf.txt lab10_3.sh lab12_3.sh ski_places Шаблоны
5.tmp feathers lab10_4.sh may text.txt
6.tmp file.txt lab10_.sh monthly Видео
7.tmp fout.txt lab11_1.sh my_os Документы
8.tmp hallame lab11_2.sh play Загрузки
hallame@hallame-VirtualBox:~$
```

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.



```
lab11_1.sh x lab11_2.sh x lab11_4.sh
1 #!/bin/bash
2 (find $1 -mtime -7 -daystart) | xargs tar -czf arhiv.tar

hallame@hallame-VirtualBox:~$ chmod +x lab11_4.sh
hallame@hallame-VirtualBox:~$ ./lab11_4.sh
tar: ./arhiv.tar: файл является архивом; не сброшен
hallame@hallame-VirtualBox:~$
```

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.