

# **Лабораторная работа №11**

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

Галиева Аделина Руслановна

# Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	10
4	Контрольные вопросы	11
	Список литературы	12

## Список иллюстраций

2.1	Задание 1 . . . . .	7
2.2	Задание 2 . . . . .	8
2.3	Задание 3 . . . . .	8
2.4	Задание 4 . . . . .	9

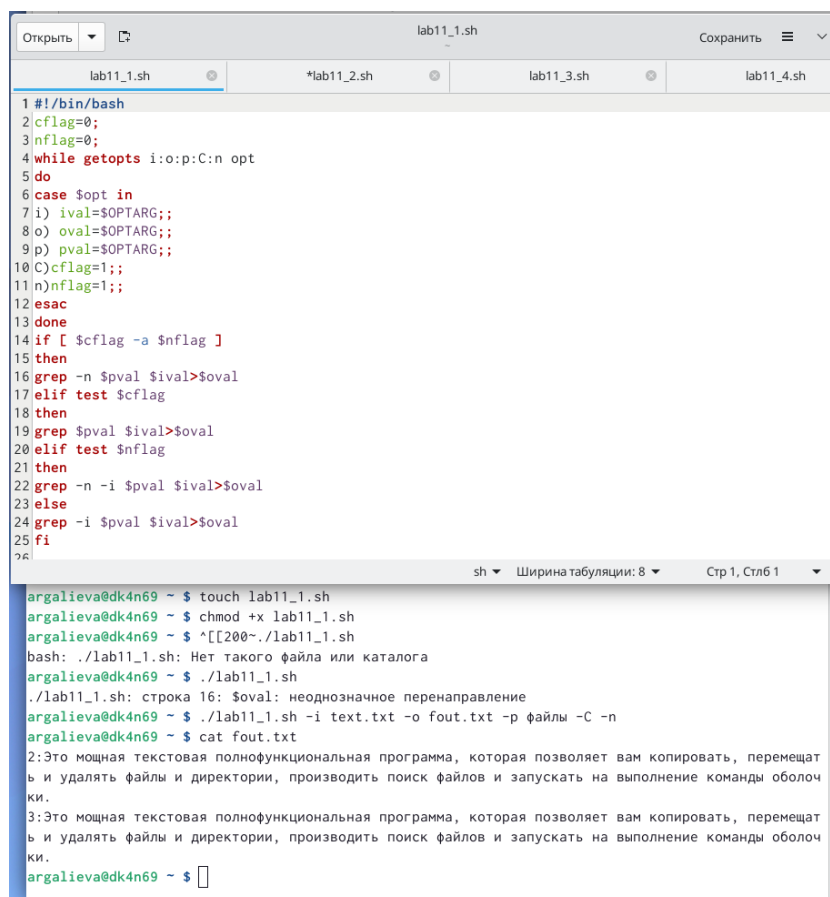
## Список таблиц

# 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. Используем команды `getopts` `grep`, пишем командный файл, который анализирует командную строку с ключами: `-iinputfile` — читает данные из указанного файла; `-ooutputfile` — выводит данные в указанный файл; `-rшаблон` — указывает шаблон для поиска; `-C` — различает большие и малые буквы; `-n` — выдаёт номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом `-r`. (рис. 2.1)



```
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
26
```

```
argalievadk4n69 ~ $ touch lab11_1.sh
argalievadk4n69 ~ $ chmod +x lab11_1.sh
argalievadk4n69 ~ $ ^[[200~./lab11_1.sh
bash: ./lab11_1.sh: Нет такого файла или каталога
argalievadk4n69 ~ $ ./lab11_1.sh
./lab11_1.sh: строка 16: $oval: неоднозначное перенаправление
argalievadk4n69 ~ $ ./lab11_1.sh -i text.txt -o fout.txt -p файлы -C -n
argalievadk4n69 ~ $ cat fout.txt
2: Это мощная текстовая полнофункциональная программа, которая позволяет вам копировать, перемещать и удалять файлы и директории, производить поиск файлов и запускать на выполнение команды оболочки.
3: Это мощная текстовая полнофункциональная программа, которая позволяет вам копировать, перемещать и удалять файлы и директории, производить поиск файлов и запускать на выполнение команды оболочки.
argalievadk4n69 ~ $
```

Рис. 2.1: Задание 1

- Пишем на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в код завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено. (рис. 2.2)

```

КИ.
argalievadk4n69 ~ $ touch lab11_2.sh
argalievadk4n69 ~ $ chmod +x lab11_2.sh
argalievadk4n69 ~ $ ./lab11_2.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

```

Рис. 2.2: Задание 2

- Пишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до  $\infty$  (например 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют). (рис. 2.3)

```

argalievadk4n69 ~ $ touch lab11_3.sh
argalievadk4n69 ~ $ chmod +x lab11_3.sh
argalievadk4n69 ~ $ ./lab11_3.sh
argalievadk4n69 ~ $ ls

```

```

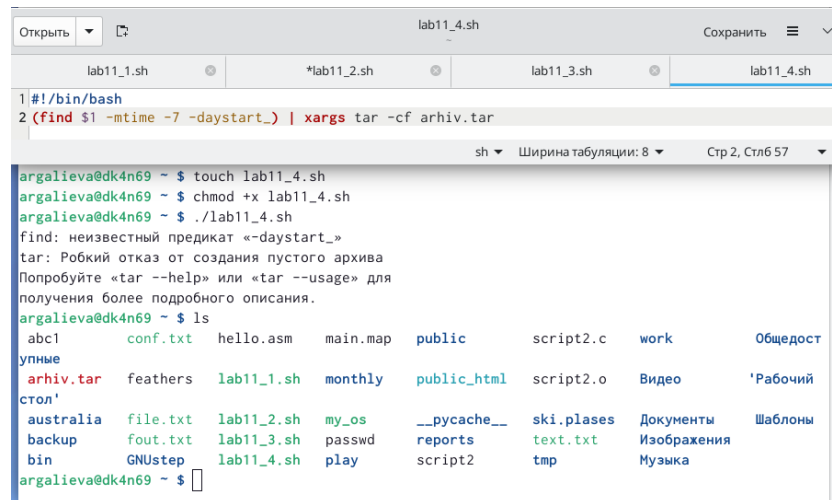
1 #!/bin/bash
2 let i=$1+1
3 while (( i-=1 ))
4 do touch $i.tmp
5 done
6 let j=$2+1
7 while (( j-=1 ))
8 do rm $j.tmp
9 done

```

Рис. 2.3: Задание 3



4. Пишем 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 (используем команду find). (рис. 2.4)



```
lab11_4.sh
1 #!/bin/bash
2 (find $1 -mtime -7 -daystart_) | xargs tar -cf arhiv.tar

argalievadk4n69 ~ $ touch lab11_4.sh
argalievadk4n69 ~ $ chmod +x lab11_4.sh
argalievadk4n69 ~ $ ./lab11_4.sh
find: неизвестный предикат «-daystart_»
tar: Роккий отказ от создания пустого архива
Попробуйте «tar --help» или «tar --usage» для
получения более подробного описания.
argalievadk4n69 ~ $ ls
abc1      conf.txt  hello.asm  main.map  public    script2.c  work      Общедост
упные
arhiv.tar  feathers  lab11_1.sh  monthly   public_html  script2.o  Видео     'Рабочий
свол'
australia  file.txt  lab11_2.sh  my_os     __pycache__  ski.plases  Документы  Шаблоны
backup     fout.txt  lab11_3.sh  passwd    reports      text.txt    Изображения
bin        GNUstep  lab11_4.sh  play      script2      tmp         Музыка
```

Рис. 2.4: Задание 4

## **3 Выводы**

Я изучила основы программирования в оболочке ОС UNIX. Научилась 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`, встречающаяся 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`.

## **Список литературы**