

Отчёт по лабораторной работе №12

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

Евгений Алексеевич Лисягин НБИбд-01-20

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

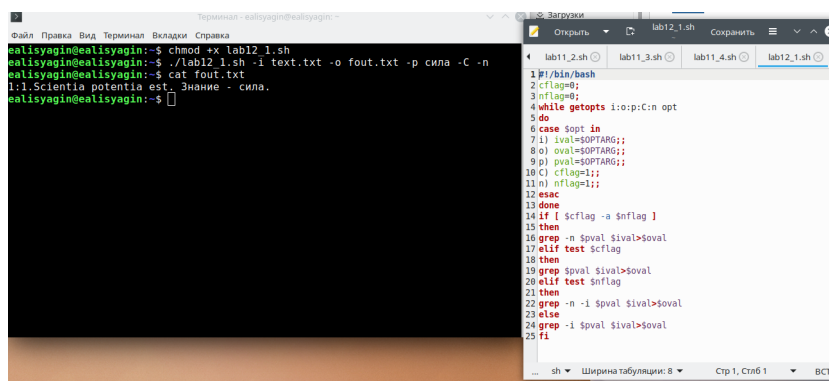
1 Цель работы

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



The image shows two windows. The left window is a terminal with the following commands and output:

```
ealisyagin@ealisyagin:~$ chmod +x lab12.1.sh
ealisyagin@ealisyagin:~$ ./lab12.1.sh -i text.txt -o fout.txt -p сила -C -n
ealisyagin@ealisyagin:~$ cat fout.txt
1:1:Scientia potentia est. Знание - сила.
ealisyagin@ealisyagin:~$
```

The right window is a text editor showing the script `lab12.1.sh`:

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

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

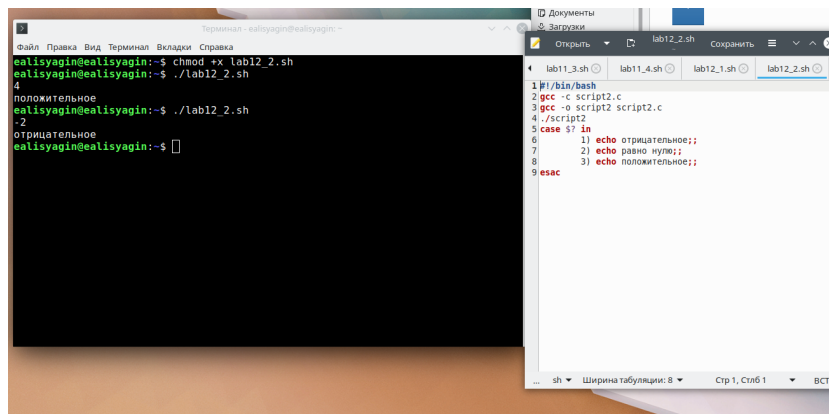


Figure 2.2: Задание 2

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

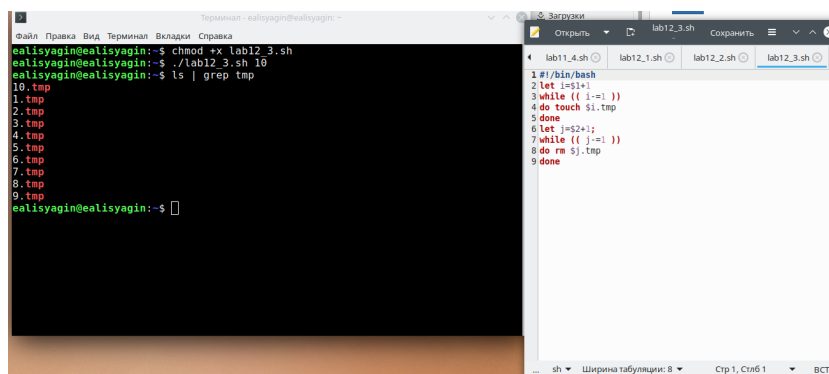


Figure 2.3: Задание 3

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.

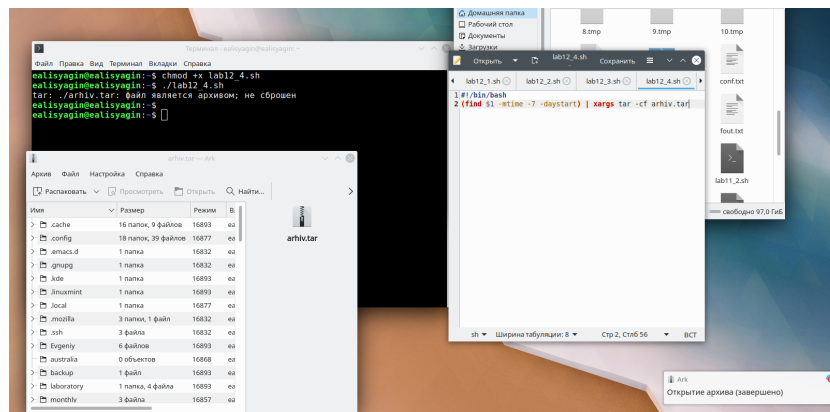


Figure 2.4: Задание 4

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.