

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Алина Молокова¹

22 апреля, 2024, Москва, Россия

¹Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

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

Задачи лабораторной работы

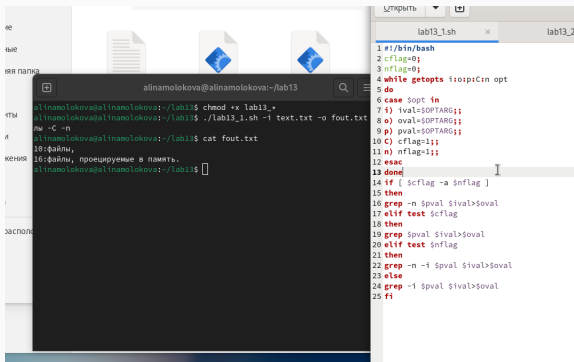
1 Выполнить 4 задания

Процесс выполнения 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

Выполнение работы



The image shows a terminal window and a script file. The terminal window, titled 'alinalokova@alinalokova:~/lab13', displays the following commands and output:

```
alinalokova@alinalokova:~/lab13$ chmod +x lab13_1.sh
alinalokova@alinalokova:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt
alinalokova@alinalokova:~/lab13$ cat fout.txt
16:файлы, проецируемые в память.
alinalokova@alinalokova:~/lab13$
```

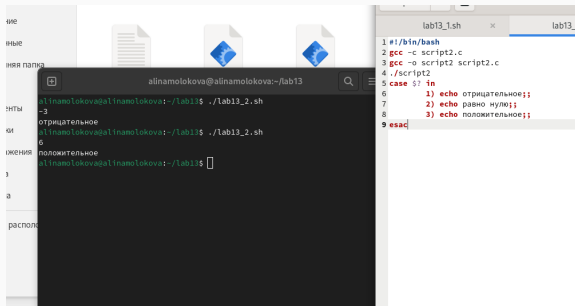
The script file, titled 'lab13_1.sh', contains the following code:

```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:ospic: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
```

Рис. 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

Выполнение работы



The image shows a Linux desktop environment. In the foreground, a terminal window titled 'alinamolokova@alinamolokova:~/lab13' displays the execution of two scripts. The first script, 'lab13_2.sh', outputs '-3' and 'отрицательное'. The second script, 'lab13_1.sh', outputs '6' and 'положительное'. In the background, a file editor window shows the content of 'lab13_1.sh', which is a shell script that compiles 'script2.c' and runs it with arguments based on a case statement.

```
alinamolokova@alinamolokova:~/lab13$ ./lab13_2.sh
-3
отрицательное
alinamolokova@alinamolokova:~/lab13$ ./lab13_1.sh
6
положительное
alinamolokova@alinamolokova:~/lab13$
```

```
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: Задание 2

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

Выполнение работы

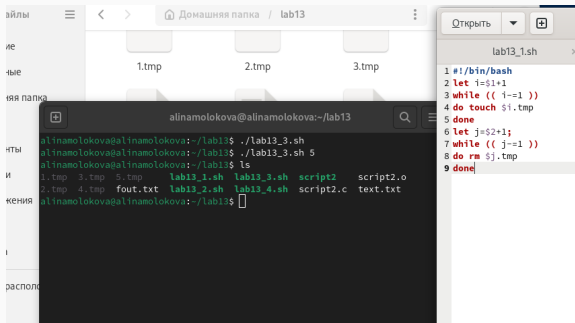


Рис. 3: Задание 3

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.

Выполнение работы

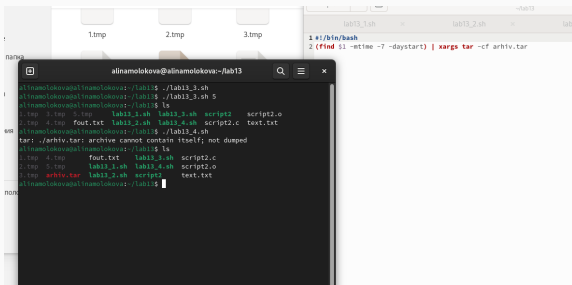


Рис. 4: Задание 4

Выводы по проделанной работе

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