

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Узаков Мадатбек¹

8 сентября, 2023, 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

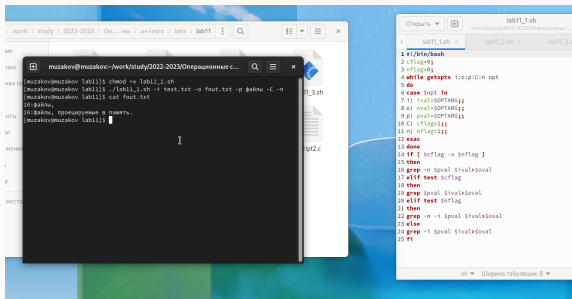
1 Выполнить 4 задания

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 Linux desktop environment with two windows. The left window is a terminal titled 'muzakov@muzakov: ~work/study/2022-2023/Операционные с...' showing the execution of a script 'lab11_1.sh'. The right window is a code editor titled 'lab11_1.sh' showing the source code of the script.

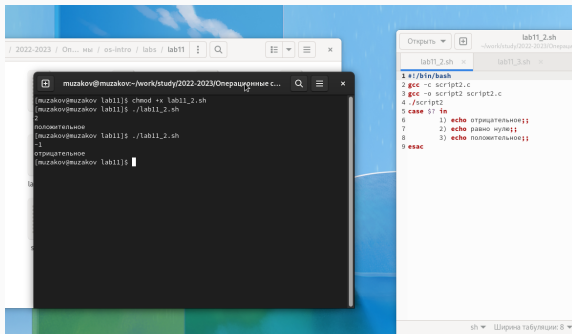
```
muzakov@muzakov lab11$ chmod +x lab11_1.sh
muzakov@muzakov lab11$ ./lab11_1.sh -i test.tet -o fout.tet -p fshu -C -n
muzakov@muzakov lab11$ cat fout.tet
10: fshu,
16: fshu, присоединены к сети.
muzakov@muzakov lab11$
```

```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:op:C:n opt
5 do
6 case $opt in
7 i) sval=$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 $sval>$oval
17 elif test $cflag
18 then
19 grep -pval $sval>$oval
20 elif test $nflag
21 then
22 grep -n -i $pval $sval>$oval
23 else
24 grep -f $pval $sval>$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



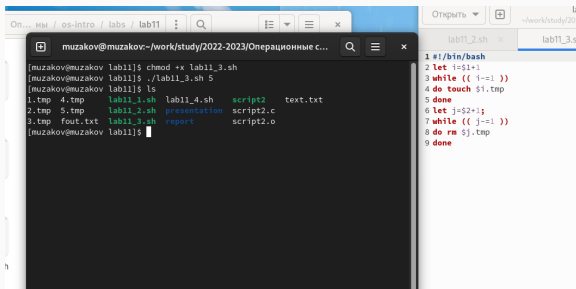
The image shows a terminal window and a file editor. The terminal window, titled 'muzakov@muzakov:~/work/study/2022-2023/Операционные с...', displays the execution of a script named 'lab11_2.sh'. The user runs 'chmod +x lab11_2.sh' and then './lab11_2.sh'. The script outputs 'положительное' and '-1', and then 'отрицательное'. The file editor, titled 'lab11_2.sh', shows the following code:

```
1 #!/bin/bash
2 gcc -c script1.c
3 gcc -o script2 script1.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



The screenshot shows a terminal window with the following commands and output:

```
[muzakov@muzakov lab11]$ chmod +x lab11_3.sh
[muzakov@muzakov lab11]$ ./lab11_3.sh 5
[muzakov@muzakov lab11]$ ls
1.tmp 4.tmp lab11_1.sh lab11_4.sh script2 text.txt
2.tmp 5.tmp lab11_2.sh presentation script2.c
3.tmp fout.txt lab11_3.sh report script2.o
[muzakov@muzakov lab11]$
```

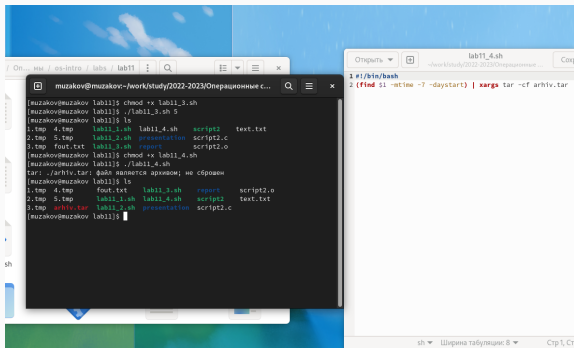
On the right side of the image, there are two snippets of shell script code:

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

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



The image shows a terminal window and a file manager window. The terminal window is titled 'muzakov@muzakov:~/work/study/2022-2023/Операционные...' and shows the following commands and output:

```
[muzakov@muzakov lab11]$ chmod +x lab11_3.sh
[muzakov@muzakov lab11]$ ./lab11_3.sh 5
[muzakov@muzakov lab11]$ ls
1.tmp 4.tmp lab11_1.sh lab11_4.sh script2 text.txt
2.tmp 5.tmp lab11_2.sh presentation script2.c
3.tmp fout.txt lab11_3.sh report script2.o
[muzakov@muzakov lab11]$ chmod +x lab11_4.sh
[muzakov@muzakov lab11]$ ./lab11_4.sh
tar: ./arhiv.tar: файл является архивом; не сбрасыв
[muzakov@muzakov lab11]$ ls
1.tmp 4.tmp fout.txt lab11_3.sh report script2.o
2.tmp 5.tmp lab11_1.sh lab11_4.sh script2 text.txt
3.tmp arhiv.tar lab11_2.sh presentation script2.c
[muzakov@muzakov lab11]$
```

The file manager window is titled 'lab11_4.sh' and shows the following commands and output:

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

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.