

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Егор Рыжов¹

20 апреля, 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

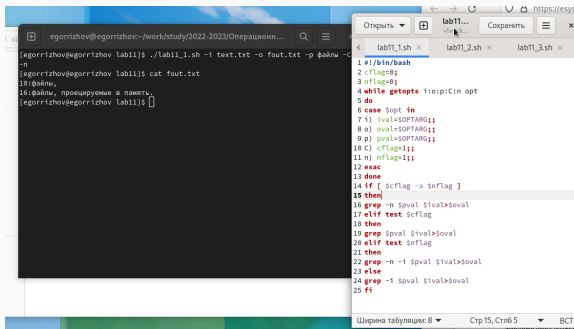
1 Выполнить 4 задания

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 terminal window on the left and a code editor on the right. The terminal window displays the execution of a script named `lab11_1.sh` with the following commands and output:

```
egorizhov@egorizhov:~/work/study/2022-2023/Операционн...  
[egorizhov@egorizhov lab11]$ ./lab11_1.sh -f text.txt -o fout.txt -p 4096 -C  
-n  
[egorizhov@egorizhov lab11]$ cat fout.txt  
18: $0line,  
16: @line, процессорные а память.  
[egorizhov@egorizhov lab11]$
```

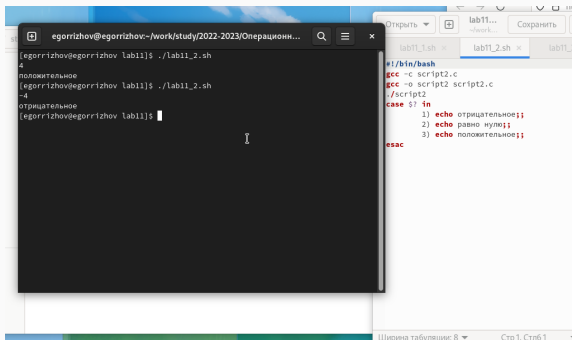
The code editor on the right shows the content of `lab11_1.sh`:

```
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 -f $pval $ival>$oval  
23 else  
24 grep -f $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



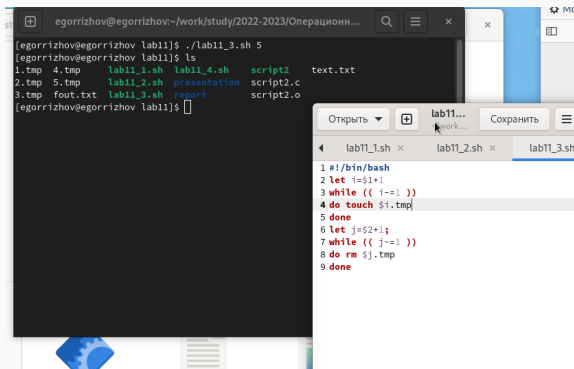
The image shows a terminal window and a code editor. The terminal window, titled 'egorizhov@egorizhov:~/work/study/2022-2023/Операционн...', displays the execution of a script named 'lab11_2.sh'. The script outputs the number '4' followed by the word 'положительное' (positive) on a new line, and '-4' followed by 'отрицательное' (negative) on another line. The code editor, titled 'lab11...', shows the source code of the script. It starts with a shebang line '#!/bin/bash', followed by compilation commands 'gcc -c script2.c' and 'gcc -o script2 script2.c', and then the execution command './script2'. A 'case' statement follows, with three branches: '1) echo отрицательное;;', '2) echo равно нулю;;', and '3) echo положительное;;', all enclosed in an 'esac' block.

```
egorizhov@egorizhov:~/work/study/2022-2023/Операционн...  
[egorizhov@egorizhov lab11]$ ./lab11_2.sh  
4  
положительное  
[egorizhov@egorizhov lab11]$ ./lab11_2.sh  
-4  
отрицательное  
[egorizhov@egorizhov lab11]$  
  
#!/bin/bash  
gcc -c script2.c  
gcc -o script2 script2.c  
./script2  
case $? in  
  1) echo отрицательное;;  
  2) echo равно нулю;;  
  3) echo положительное;;  
esac
```

Рис. 2: Задание 2

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

Выполнение работы



The image shows a terminal window and a file editor. The terminal window, titled 'egorrizhov@egorrizhov:~/work/study/2022-2023/Операционн...', displays the output of a script execution. The file editor, titled 'lab11...', shows the source code of the script 'lab11_3.sh'.

```
[egorrizhov@egorrizhov lab11]$ ./lab11_3.sh 5
[egorrizhov@egorrizhov 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
[egorrizhov@egorrizhov lab11]$
```

```
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. 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

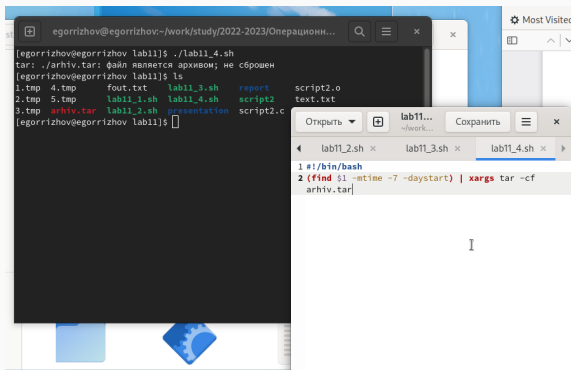


Рис. 4: Задание 4

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.