

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Булат Исаев¹

13 июня, 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

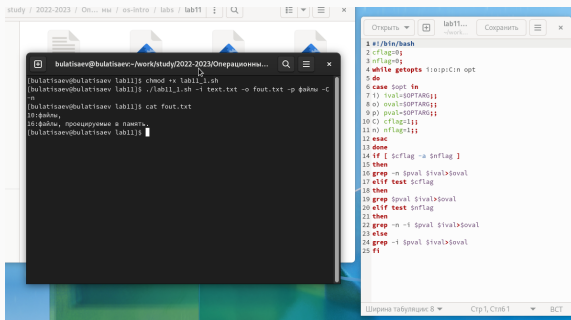
1 Выполнить 4 задания

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 and a file editor. The terminal window, titled 'bulatiseev@bulatiseev:~/work/study/2022-2023/Операционны...', displays the following commands and output:

```
bulatiseev@bulatiseev lab11$ chmod +x lab11_1.sh
bulatiseev@bulatiseev lab11$ ./lab11_1.sh -i text.txt -o fout.txt -p $@nW -C
-a
bulatiseev@bulatiseev lab11$ cat fout.txt
10: $@nW,
10: $@nW, прочитанные в память.
bulatiseev@bulatiseev lab11$
```

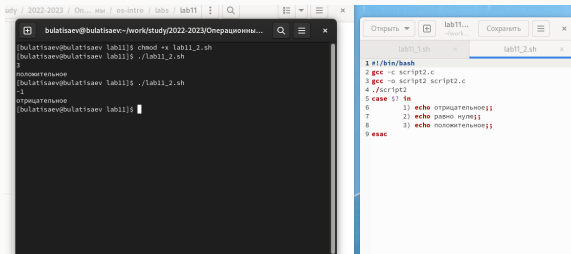
The file editor, titled 'lab11...', shows the following shell script code:

```
1 #!/bin/bash
2 cflag=0
3 nflag=0
4 while getopts 1io:piC: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 -i $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



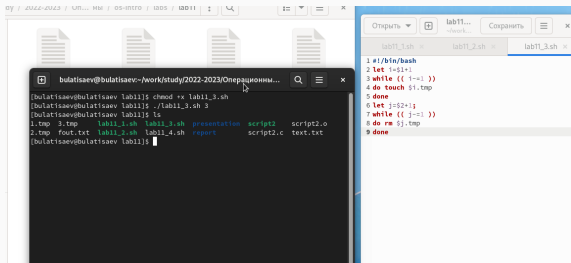
```
bulatisev@bulatisev:~/work/study/2022-2023/Операционны...  
[bulatisev@bulatisev lab11]$ chmod +x lab11_2.sh  
[bulatisev@bulatisev lab11]$ ./lab11_2.sh  
3  
положительное  
-1  
отрицательное  
[bulatisev@bulatisev lab11]$
```

```
#!/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



The image shows a terminal window and a file editor. The terminal window displays the following commands and output:

```
bulatissaev@bulatissaev:~/work/study/2022-2023/Операционны...
bulatissaev@bulatissaev lab11$ chmod +x lab11_3.sh
bulatissaev@bulatissaev lab11$ ./lab11_3.sh 3
bulatissaev@bulatissaev lab11$ ls
1.tmp 3.tmp lab11_1.sh lab11_3.sh presentation script2 script2.o
2.tmp fout.txt lab11_2.sh lab11_4.sh report script2.c text.txt
bulatissaev@bulatissaev lab11$
```

The file editor shows the following code in a file named `lab11_3.sh`:

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

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

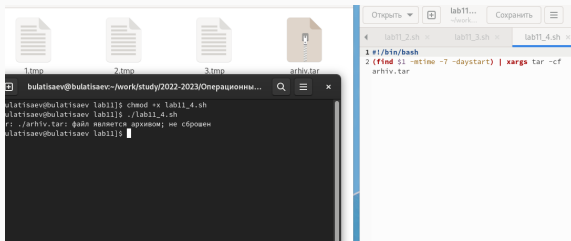


Рис. 4: Задание 4

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.