

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Арсений Февралёв¹

18 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

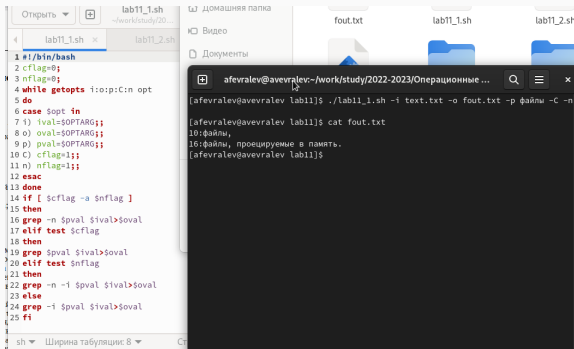
1 Выполнить 4 задания

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 Linux desktop environment. On the left, a text editor window titled 'lab11_1.sh' displays a shell script. The script defines a function 'f' that takes an input file and an output file, and then calls 'f' with 'text.txt' as input and 'fout.txt' as output. The script also includes a 'cat' command to display the contents of 'fout.txt'. On the right, a terminal window titled 'afevralev@afevralev-~/work/study/2022-2023/Операционные ...' shows the execution of the script. The user runs './lab11_1.sh -i text.txt -o fout.txt -p файлы -C -n', and the output of the 'cat' command is displayed: '10:файлы, 16:файлы, проецируемые в память.'

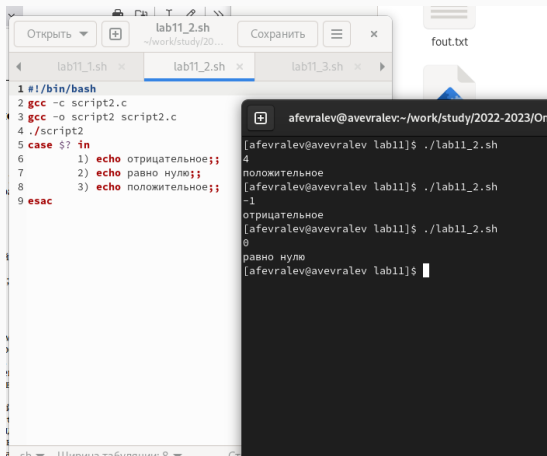
```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:o:p:C:n opt
5 do
6 case $opt in
7 i) ival=$OPTARG;;
8 o) oval=$OPTARG;;
9 p) pval=$OPTARG;;
10 C) cflag=1;;
11 n) nflag=1;;
12 esac
13 done
14 if [ $cflag -a $nflag ]
15 then
16 grep -n $pval $ival>$oval
17 elif test $cflag
18 then
19 grep $pval $ival>$oval
20 elif test $nflag
21 then
22 grep -n -i $pval $ival>$oval
23 else
24 grep -i $pval $ival>$oval
25 fi
26
```

```
[afevralev@afevralev lab11]$ ./lab11_1.sh -i text.txt -o fout.txt -p файлы -C -n
[afevralev@afevralev lab11]$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
[afevralev@afevralev lab11]$
```

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 code editor window with three tabs: lab11_1.sh, lab11_2.sh (active), and lab11_3.sh. The active tab contains a shell script with the following content:

```
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
```

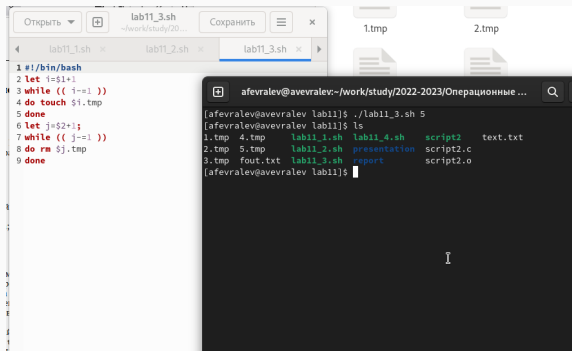
Overlaid on the bottom right is a terminal window showing the execution of the script. The prompt is [afevrarev@avevrarev lab11]\$ and the command is ./lab11_2.sh. The output shows the results of the case statement for three different inputs: 4, -1, and 0.

```
[afevrarev@avevrarev lab11]$ ./lab11_2.sh
4
положительное
[afevrarev@avevrarev lab11]$ ./lab11_2.sh
-1
отрицательное
[afevrarev@avevrarev lab11]$ ./lab11_2.sh
0
равно нулю
[afevrarev@avevrarev lab11]$
```

Рис. 2: Задание 2

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

Выполнение работы



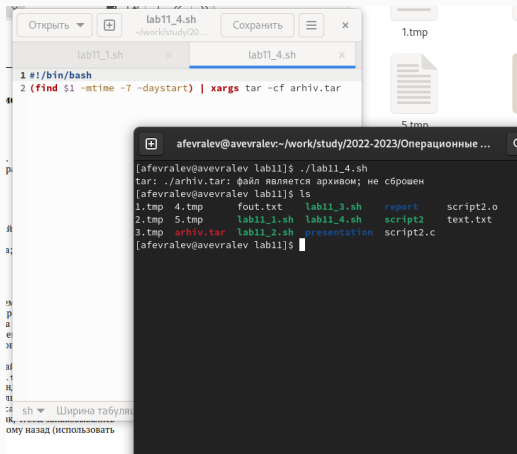
```
1 #!/bin/bash
2 let i=$1+1
3 while (( i-=1 ))
4 do touch $i.tmp
5 done
6 let j=$2+1
7 while (( j-=1 ))
8 do rm $j.tmp
9 done
```

```
[afevralev@avevralev lab11]$ ./lab11_3.sh 5
[afevralev@avevralev 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
[afevralev@avevralev lab11]$
```

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 screenshot shows a terminal window with two tabs: 'lab11_1.sh' and 'lab11_4.sh'. The 'lab11_4.sh' tab is active, displaying the following commands and output:

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

Below the terminal window, a file listing is shown, representing the output of the 'ls' command:

File	File	File	File	File	File
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	

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.