

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Мандрик Анастасия НБИбд-01-21¹

23 августа, 2022, 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

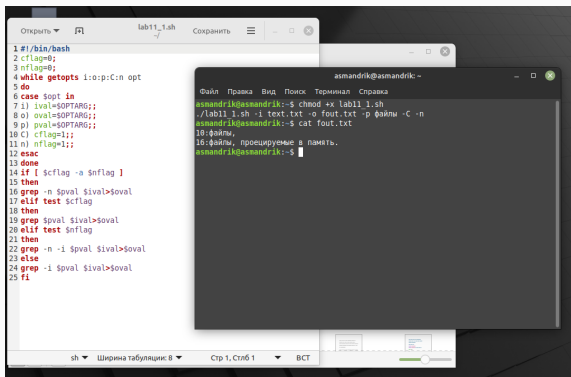
1 Выполнить 4 задания

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 screenshot shows a terminal window with a shell script and its execution. The script is a C flag parser. The execution shows the script being run with arguments, and the output of the script is displayed in a separate window.

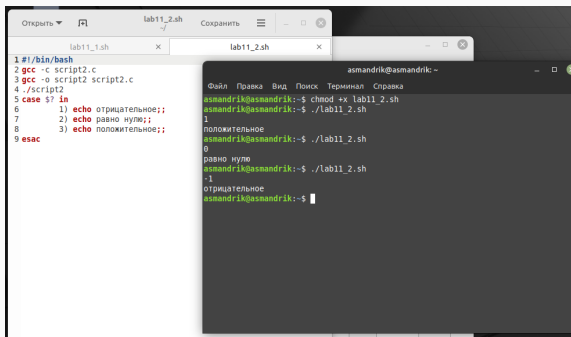
```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:o:p:C:n opt
5 do
6 case $opt in
7 i) ival=$OPTARG;;
8 o) oval=$OPTARG;;
9 p) pval=$OPTARG;;
10 C) cflag=1;;
11 n) nflag=1;;
12 esac
13 done
14 if [ $cflag -a $nflag ]
15 then
16 grep -n $pval $ival>$oval
17 elif test $cflag
18 then
19 grep $pval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

```
asmandrik@asmandrik:~$ chmod +x lab11.1.sh
./lab11.1.sh -i text.txt -o fout.txt -p файл -C -n
asmandrik@asmandrik:~$ cat fout.txt
10:файл,
16:файл, проецируемые в память.
asmandrik@asmandrik:~$
```

Figure 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 screenshot shows a terminal window with two tabs: 'lab11_1.sh' and 'lab11_2.sh'. The 'lab11_1.sh' tab is active and displays the following code:

```
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
```

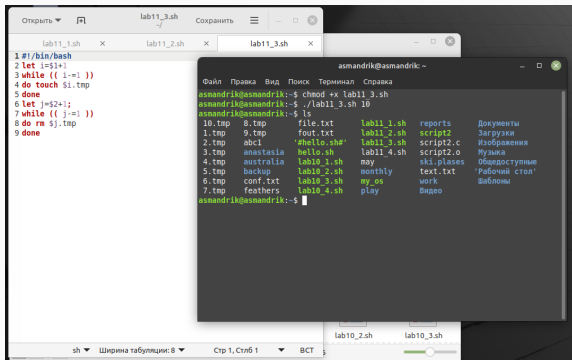
The 'lab11_2.sh' tab is also visible and shows the output of the program:

```
asmandrik@asmandrik:~$ chmod +x lab11_2.sh
asmandrik@asmandrik:~$ ./lab11_2.sh
1
положительное
asmandrik@asmandrik:~$ ./lab11_2.sh
0
равно нулю
asmandrik@asmandrik:~$ ./lab11_2.sh
-1
отрицательное
asmandrik@asmandrik:~$
```

Figure 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 a shell script being executed. The script is as follows:

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

The terminal output shows the user running the script with the command `chmod +x lab11_3.sh` and then `./lab11_3.sh 10`. The output of the script is a list of files and directories:

File	Permissions	Owner	Group	Size	File	Permissions	Owner	Group	Size
10.tmp	-rw-r--r--	asm	asm	4096	file.txt	-rw-r--r--	asm	asm	4096
1.tmp	-rw-r--r--	asm	asm	4096	fout.txt	-rw-r--r--	asm	asm	4096
2.tmp	-rw-r--r--	asm	asm	4096	'hello.sh'	-rw-r--r--	asm	asm	4096
3.tmp	-rw-r--r--	asm	asm	4096	hello.sh	-rw-r--r--	asm	asm	4096
4.tmp	-rw-r--r--	asm	asm	4096	australia	-rw-r--r--	asm	asm	4096
5.tmp	-rw-r--r--	asm	asm	4096	backup	-rw-r--r--	asm	asm	4096
6.tmp	-rw-r--r--	asm	asm	4096	conf.txt	-rw-r--r--	asm	asm	4096
7.tmp	-rw-r--r--	asm	asm	4096	feathers	-rw-r--r--	asm	asm	4096

The terminal window also shows a file manager view with the following files and directories:

File	Permissions	Owner	Group	Size	File	Permissions	Owner	Group	Size
lab11_1.sh	-rw-r--r--	asm	asm	4096	reports	-rw-r--r--	asm	asm	4096
lab11_2.sh	-rw-r--r--	asm	asm	4096	script2	-rw-r--r--	asm	asm	4096
lab11_3.sh	-rw-r--r--	asm	asm	4096	script2.c	-rw-r--r--	asm	asm	4096
lab11_4.sh	-rw-r--r--	asm	asm	4096	script2.o	-rw-r--r--	asm	asm	4096
may	-rw-r--r--	asm	asm	4096	ski_places	-rw-r--r--	asm	asm	4096
monthly	-rw-r--r--	asm	asm	4096	text.txt	-rw-r--r--	asm	asm	4096
my_os	-rw-r--r--	asm	asm	4096	'Рабочий стол'	-rw-r--r--	asm	asm	4096
work	-rw-r--r--	asm	asm	4096	Шаблоны	-rw-r--r--	asm	asm	4096
play	-rw-r--r--	asm	asm	4096	Видео	-rw-r--r--	asm	asm	4096

Figure 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

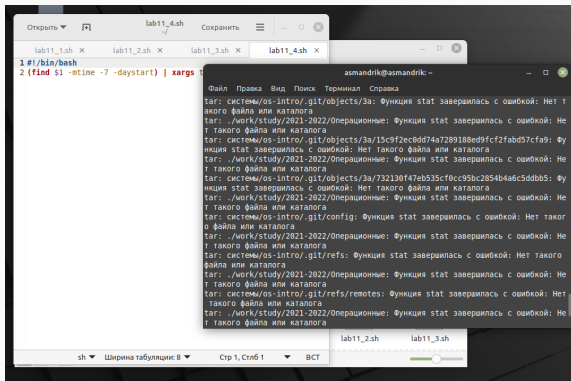


Figure 4: Задание 4

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.