

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Иовлев Максим Андреевич НПИбд-01-22¹

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

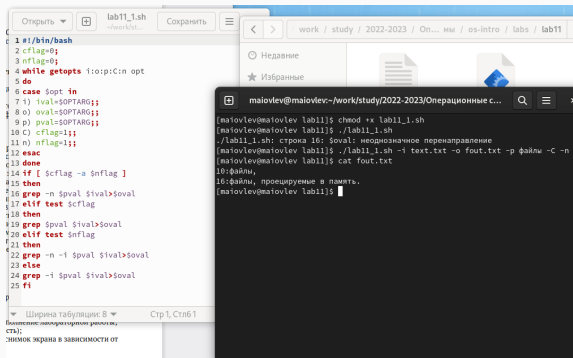
1 Выполнить 4 задания

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 terminal window with a light blue title bar. The window has a menu bar with 'Открыть', 'lab11_1.sh', and 'Сохранить'. Below the menu bar is a toolbar with icons for opening, saving, and a list icon. The main area of the window displays a shell script with line numbers 1 through 25. The script uses a while loop to getopts, a case statement to handle flags, and a series of if, then, else, and fi statements to process the flags. The script also uses grep to search for patterns in a file. The terminal window is titled 'maiolev@maiolev:~/work/study/2022-2023/Операционные с...'. The prompt is 'maiolev@maiolev lab11\$'. The user has entered 'chmod +x lab11_1.sh' and './lab11_1.sh'. The output shows the script's execution, including the creation of a file 'fout.txt' and the output of the cat command.

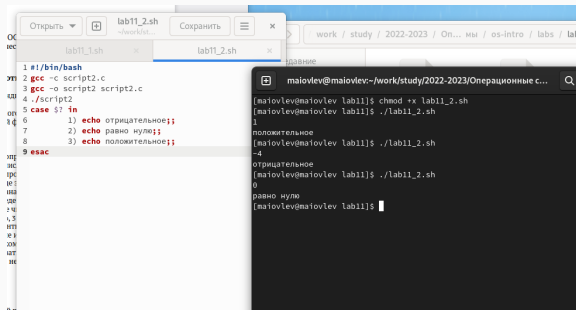
```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:op:C:n opt
5 do
6 case $opt in
7 i) ival=$OPTARG;;
8 o) oval=$OPTARG;;
9 p) pval=$OPTARG;;
10 C) cflag=1;;
11 n) nflag=1;;
12 esac
13 done
14 if [ $cflag -a $nflag ]
15 then
16 grep -n $pval $ival>$oval
17 elif test $cflag
18 then
19 grep $pval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

maiolev@maiolev lab11\$ chmod +x lab11_1.sh
maiolev@maiolev lab11\$./lab11_1.sh
./lab11_1.sh: строка 16: \$oval: неоднозначное перенаправление
maiolev@maiolev lab11\$./lab11_1.sh -i text.txt -o fout.txt -p файлы -C -n
maiolev@maiolev lab11\$ cat fout.txt
16:файлы,
16:файлы, проецируемые в память.
maiolev@maiolev 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



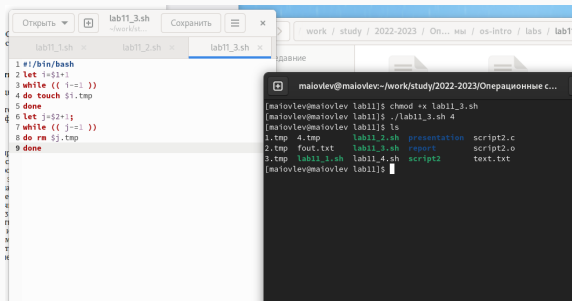
```
1 #!/bin/bash
2 gcc -c script2.c
3 gcc -o script2 script2.c
4 ./script2
5 case $? in
6     1) echo отрицательное;;
7     2) echo равно нулю;;
8     3) echo положительное;;
9 esac
```

```
[maiovlev@maiovlev ~/work/study/2022-2023/Операционные с...]$ chmod +x lab11_2.sh
[maiovlev@maiovlev lab11]$ ./lab11_2.sh
1
положительное
[maiovlev@maiovlev lab11]$ ./lab11_2.sh
-4
отрицательное
[maiovlev@maiovlev lab11]$ ./lab11_2.sh
0
равно нулю
[maiovlev@maiovlev lab11]$
```

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 with a light blue header bar. The window title is "maiolev@maiolev:~/work/study/2022-2023/Операционные с...". The terminal content shows the following commands and output:

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

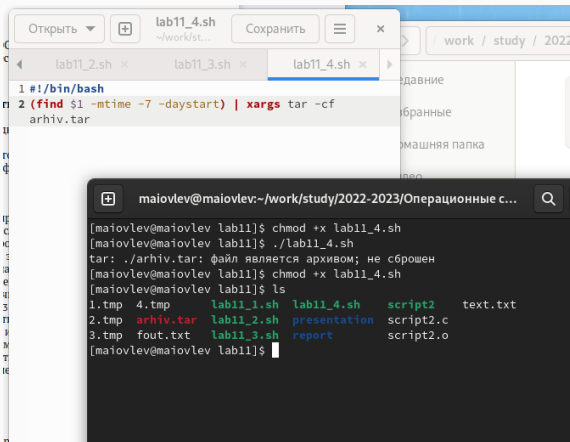
In the background, another window titled "lab11_3.sh" is visible, showing the script code:

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



The image shows a terminal window and a file manager. The terminal window displays the execution of a script named `lab11_4.sh`. The script uses `find` to locate files older than 7 days and archives them into `arhiv.tar` using `xargs tar -cf`. The file manager shows the contents of the `lab11_4.sh` file, which is a shell script. The terminal output shows the execution of `chmod +x lab11_4.sh`, `./lab11_4.sh`, and `ls`. The `ls` command output lists the files in the current directory, including `1.tmp`, `4.tmp`, `lab11_1.sh`, `lab11_4.sh`, `script2`, `text.txt`, `2.tmp`, `arhiv.tar`, `lab11_2.sh`, `presentation`, `script2.c`, `3.tmp`, `fout.txt`, `lab11_3.sh`, `report`, and `script2.o`.

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

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

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.