

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Дурдалыев Максат¹

21 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

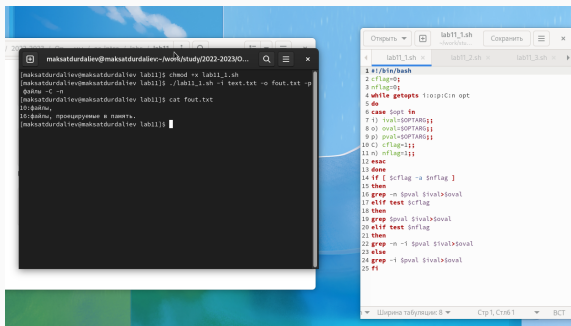
1 Выполнить 4 задания

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 on the left and a code editor on the right. The terminal window displays the execution of a shell script named `lab11_1.sh`. The user runs `chmod +x lab11_1.sh`, then `./lab11_1.sh -i text.txt -o fout.txt -p 5000 -C -n`, and finally `cat fout.txt`. The output of the script is displayed in the terminal. The code editor on the right shows the source code of `lab11_1.sh`, which is a shell script that takes several arguments and uses `grep` to search for patterns in a file.

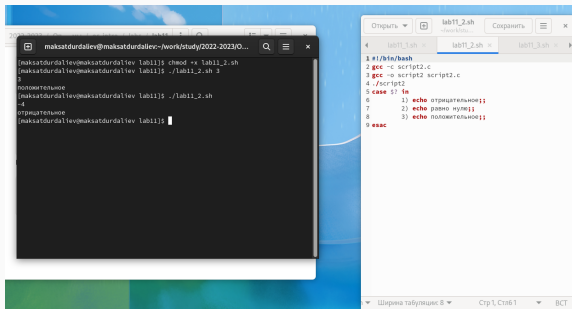
```
[maksatdurdaliev@maksatdurdaliev:~/work/study/2022-2023/IO...]
[maksatdurdaliev@maksatdurdaliev lab11]$ chmod +x lab11_1.sh
[maksatdurdaliev@maksatdurdaliev lab11]$ ./lab11_1.sh -i text.txt -o fout.txt -p
5000 -C -n
[maksatdurdaliev@maksatdurdaliev lab11]$ cat fout.txt
10:failw,
11:failw, пропускаящие в numbers.
[maksatdurdaliev@maksatdurdaliev lab11]$
```

```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:opiCin 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
```

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 on the left and a code editor on the right. The terminal window displays the execution of a shell script named `lab11_2.sh`. The script prompts the user to enter a number, and the user enters `3`. The script then prints `положительное` (positive). The code editor shows the source code of `lab11_2.sh`, which includes a `case` statement to check if the input is positive, zero, or negative.

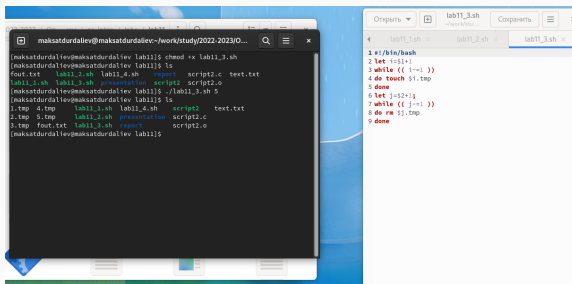
```
maksatdurdaliev@maksatdurdaliev:~/work/study/2022-2023/O...  
[maksatdurdaliev@maksatdurdaliev: lab11]$ pwd -P  
[maksatdurdaliev@maksatdurdaliev: lab11]$ ./lab11_2.sh  
3  
положительное  
[maksatdurdaliev@maksatdurdaliev: lab11]$ ./lab11_2.sh  
-4  
отрицательное  
[maksatdurdaliev@maksatdurdaliev: lab11]$
```

```
1 #!/bin/bash  
2 gcc -o 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, titled 'maksatduraliev@maksatduraliev:~/work/study/2022-2023/O...', displays the execution of a shell script 'lab11_3.sh'. The script performs several actions: it sets permissions on 'lab11_3.sh', lists files, runs 'fout.txt', and executes 'script2.c' and 'script2.o'. The file editor, titled 'lab11_3.sh', shows the script's source code, which includes a shebang, variable assignments, loops, and file operations.

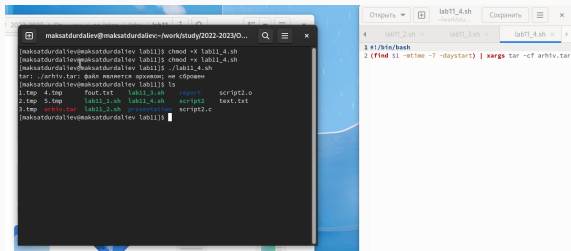
```
maksatduraliev@maksatduraliev:~/work/study/2022-2023/O...  
[maksatduraliev@maksatduraliev lab11]$ chmod +x lab11_3.sh  
[maksatduraliev@maksatduraliev lab11]$ ls  
fout.txt  lab11_2.sh  lab11_4.sh  report  script2.c  text.txt  
lab11_3.sh  lab11_3.sh  presentation  script2.o  
[maksatduraliev@maksatduraliev lab11]$ ./lab11_3.sh 5  
[maksatduraliev@maksatduraliev lab11]$ ls  
1.tmp  4.tmp  lab11_1.sh  lab11_4.sh  script1  text.txt  
2.tmp  5.tmp  lab11_2.sh  presentation  script2.c  
3.tmp  fout.txt  lab11_3.sh  report  script2.o  
[maksatduraliev@maksatduraliev lab11]$
```

```
#!/bin/bash  
2 let i=i+1  
3 while (( i-->1 ))  
4 do touch $1.tmp  
5 done  
6 let j=i2+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, titled 'maksatduraliev@maksatduraliev:~/work/study/2022-2023/O...', displays the following commands and output:

```
[maksatduraliev@maksatduraliev lab11]$ chmod +x lab11_4.sh
[maksatduraliev@maksatduraliev lab11]$ chmod +x lab11_4.sh
[maksatduraliev@maksatduraliev lab11]$ ./lab11_4.sh
tar: ./arhiv.tar: file not found: archive not found
[maksatduraliev@maksatduraliev lab11]$ ls
1.tmp 4.tmp      fout.txt  lab11_3.sh  export  script2.o
2.tmp 5.tmp      lab11_4.sh  lab11_4.sh  script2  text.txt
3.tmp arhiv.tar  lab11_3.sh  presentation script2.c
[maksatduraliev@maksatduraliev lab11]$
```

The file manager window, titled 'lab11_4.sh', shows the following commands:

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

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.