

Операционные системы

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Хиджази Мохамад

21 апреля 2025

Российский университет дружбы народов, Москва, Россия

Цели и задачи работы

Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

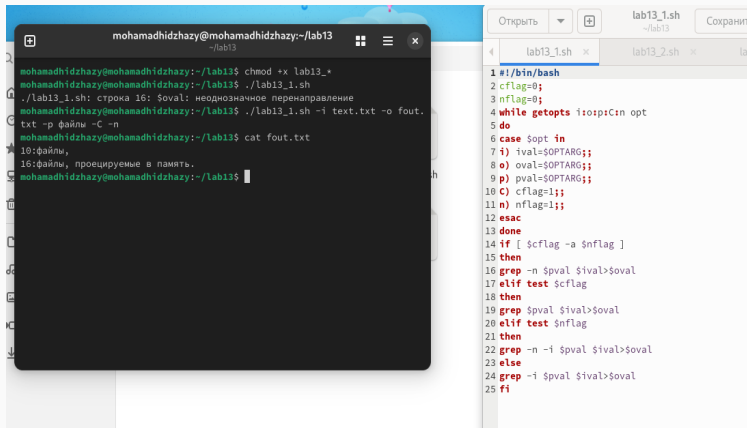
1 Выполнить 4 задания

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; `-p 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, titled 'mohamadhazzy@mohamadhazzy:~/lab13', shows the execution of a script 'lab13_1.sh' with various options. The output of the script is displayed in the terminal. The code editor on the right shows the source code of 'lab13_1.sh'.

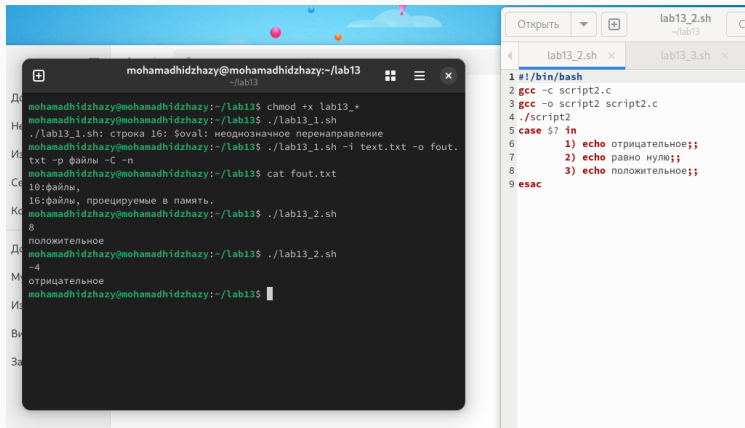
```
mohamadhazzy@mohamadhazzy:~/lab13$ chmod +x lab13_*
mohamadhazzy@mohamadhazzy:~/lab13$ ./lab13_1.sh
./lab13_1.sh: строка 16: $oval: неоднозначное перенаправление
mohamadhazzy@mohamadhazzy:~/lab13$ ./lab13_1.sh -i text.txt -o fout.
txt -p файлы -C -n
mohamadhazzy@mohamadhazzy:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
mohamadhazzy@mohamadhazzy:~/lab13$
```

```
#!/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
```

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 and a code editor. The terminal window, titled 'mohamadhidzhazy@mohamadhidzhazy:~/lab13', displays the execution of two shell scripts. The first script, 'lab13_1.sh', is run with the command './lab13_1.sh' and produces output for a 'case' statement. The second script, 'lab13_2.sh', is run with the command './lab13_2.sh' and produces output for an 'if' statement. The code editor, titled 'lab13_2.sh', shows the source code of the second script, which is a shell script that uses 'gcc' to compile 'script2.c' and then runs it. The script uses a 'case' statement to echo different messages based on the input value.

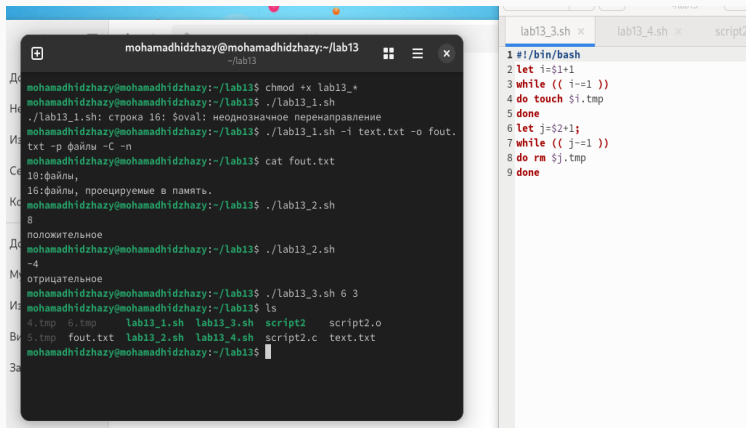
```
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ chmod +x lab13_*
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ ./lab13_1.sh
./lab13_1.sh: строка 16: $oval: неоднозначное перенаправление
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ ./lab13_1.sh -i text.txt -o fout.
txt -p файлы -C -n
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ ./lab13_2.sh
8
положительное
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ ./lab13_2.sh
-4
отрицательное
mohamadhidzhazy@mohamadhidzhazy:~/lab13$
```

```
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
```

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 separate window displaying shell script code.

Terminal Window:

```
mohamadhidzhazy@mohamadhidzhazy:~/lab13
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ chmod +x lab13_*
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ ./lab13_1.sh
./lab13_1.sh: строка 16: $oval: неоднозначное перенаправление
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ ./lab13_1.sh -i text.txt -o fout.
txt -p файлы -C -n
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ ./lab13_2.sh
8
положительное
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ ./lab13_2.sh
-4
отрицательное
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ ./lab13_3.sh 6 3
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ ls
4.tmp  6.tmp  lab13_1.sh  lab13_3.sh  script2  script2.o
5.tmp  fout.txt lab13_2.sh  lab13_4.sh  script2.c  text.txt
mohamadhidzhazy@mohamadhidzhazy:~/lab13$
```

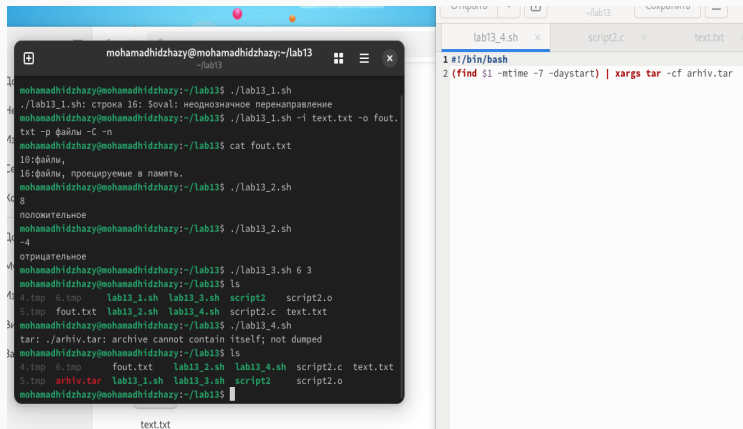
Script Code Window:

```
lab13_3.sh x  lab13_4.sh x  script2
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
```

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 'mohamadhidzhazy@mohamadhidzhazy:~/lab13', displays the following commands and output:

```
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ ./lab13_1.sh
./lab13_1.sh: строка 16: $0val: неоднозначное перенаправление
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ ./lab13_2.sh
положительное
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ ./lab13_2.sh
-4
отрицательное
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ ./lab13_3.sh 6 3
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ ls
4.tmp 6.tmp lab13_1.sh lab13_3.sh script2 script2.o
5.tmp fout.txt lab13_2.sh lab13_4.sh script2.c text.txt
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
mohamadhidzhazy@mohamadhidzhazy:~/lab13$ ls
4.tmp 6.tmp fout.txt lab13_2.sh lab13_4.sh script2.c text.txt
5.tmp arhiv.tar lab13_1.sh lab13_3.sh script2 script2.o
mohamadhidzhazy@mohamadhidzhazy:~/lab13$
```

The file manager shows the following files in the ~/lab13 directory:

- lab13_4.sh
- script2.c
- text.txt

The terminal window also shows the output of the 'ls' command, which includes the following files:

- 4.tmp
- 6.tmp
- lab13_1.sh
- lab13_3.sh
- script2
- script2.o
- 5.tmp
- fout.txt
- lab13_2.sh
- lab13_4.sh
- script2.c
- text.txt
- arhiv.tar
- lab13_1.sh
- lab13_3.sh
- script2
- script2.o

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.