

# Операционные системы

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Хоссейн Исмаил Ибрахим Аль-хаязи

16 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

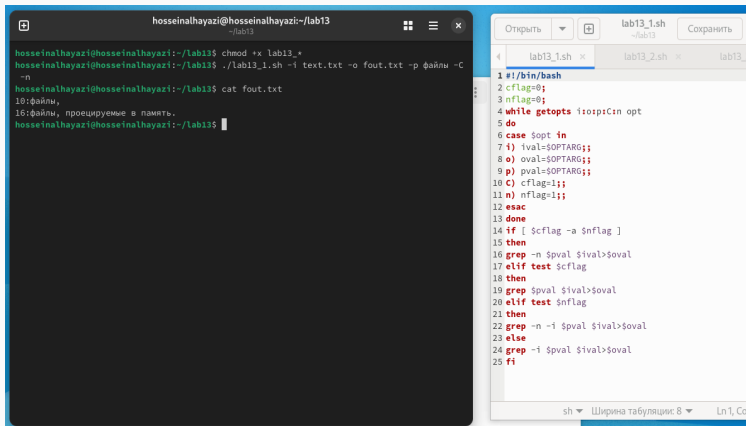
## Процесс выполнения лабораторной работы

---

1. Используя команды `getopts` `grep` напомним 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, titled 'hosseinalhayazi@hosseinalhayazi:~/lab13', shows the execution of a script 'lab13\_1.sh' with arguments '-i text.txt -o fout.txt -p файлы -C -n'. The output of the script is displayed in 'fout.txt':

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

The code editor on the right shows the source code of 'lab13\_1.sh'. The script is a shell script that processes command-line arguments and uses 'grep' to search for patterns in the input file.

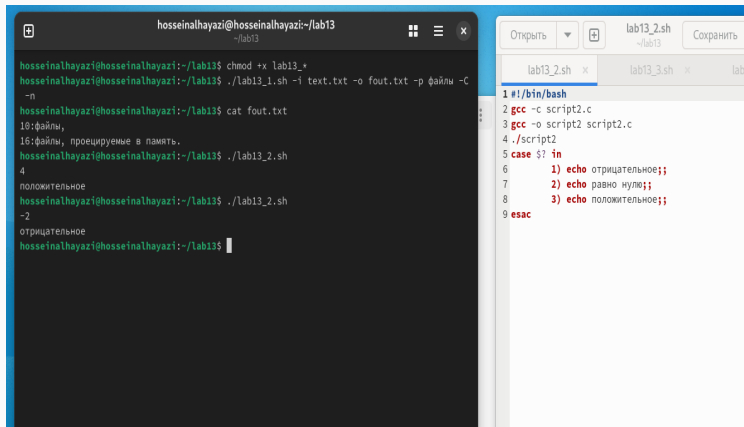
```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:osp:C:n opt
5 do
6 case $opt in
7 i) ival=$OPTARG;;
8 o) oval=$OPTARG;;
9 p) pval=$OPTARG;;
10 C) cflag=1;;
11 n) nflag=1;;
12 esac
13 done
14 if [ $cflag -a $nflag ]
15 then
16 grep -n $pval $ival>$oval
17 elif test $cflag
18 then
19 grep $pval $ival>$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 on the left and a code editor on the right. The terminal window has a title bar 'hosseinalhayazi@hosseinalhayazi:~/lab13' and shows the following commands and output:

```
hosseinalhayazi@hosseinalhayazi:~/lab13$ chmod +x lab13_*
hosseinalhayazi@hosseinalhayazi:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C
-n
hosseinalhayazi@hosseinalhayazi:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
hosseinalhayazi@hosseinalhayazi:~/lab13$ ./lab13_2.sh
4
положительное
hosseinalhayazi@hosseinalhayazi:~/lab13$ ./lab13_2.sh
-2
отрицательное
hosseinalhayazi@hosseinalhayazi:~/lab13$
```

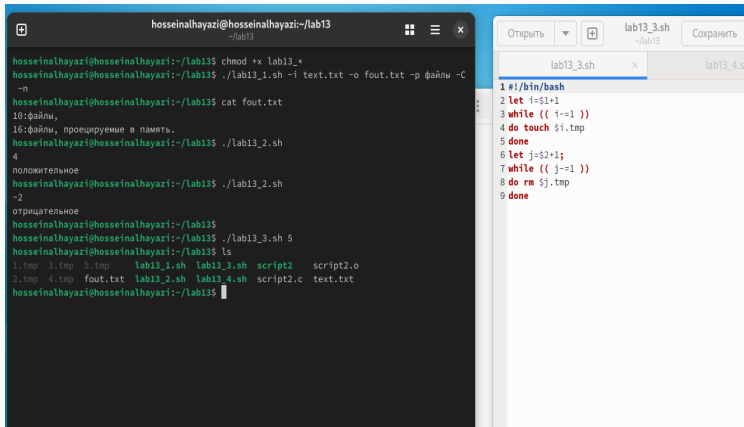
The code editor on the right has a title bar 'lab13\_2.sh' and shows the following code:

```
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
```

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 on the left and a file editor on the right. The terminal window, titled 'hosseinalhayazi@hosseinalhayazi:~/lab13', displays the execution of several shell scripts. The first script, 'lab13\_1.sh', is executed with arguments '-i text.txt -o fout.txt -p файлы -C -n'. The output of 'cat fout.txt' shows a list of files: '10:файлы,' and '16:файлы, проецируемые в память.'. The second script, 'lab13\_2.sh', is executed with arguments '-4' and '-2', resulting in the output 'положительное' and 'отрицательное' respectively. The third script, 'lab13\_3.sh', is executed with argument '5'. Finally, the 'ls' command is used to list the contents of the directory, showing files like '1.tmp', '3.tmp', '5.tmp', 'lab13\_1.sh', 'lab13\_3.sh', 'script2', 'script2.o', '2.tmp', '4.tmp', 'fout.txt', 'lab13\_2.sh', 'lab13\_4.sh', 'script2.c', and 'text.txt'. The file editor on the right, titled 'lab13\_3.sh', shows the source code of the third script, which is a bash script that initializes a variable 'i' to 1, enters a 'while' loop that increments 'i' and touches a file '\$i.tmp' until 'i' reaches 5, and then removes the files.

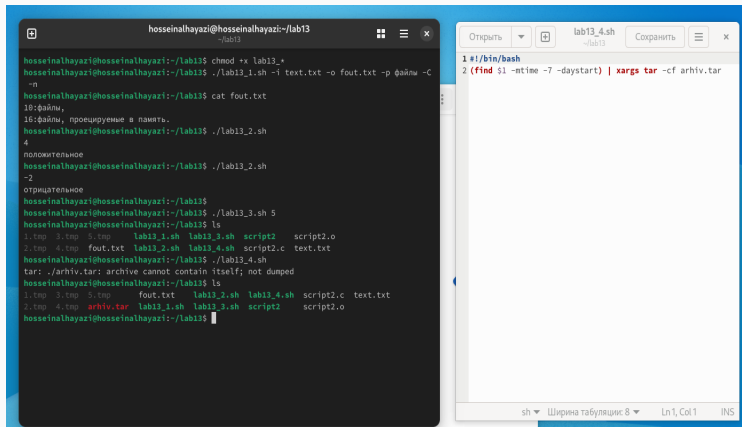
```
hosseinalhayazi@hosseinalhayazi:~/lab13$ chmod +x lab13_*
hosseinalhayazi@hosseinalhayazi:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C
-n
hosseinalhayazi@hosseinalhayazi:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
hosseinalhayazi@hosseinalhayazi:~/lab13$ ./lab13_2.sh
4
положительное
hosseinalhayazi@hosseinalhayazi:~/lab13$ ./lab13_2.sh
-2
отрицательное
hosseinalhayazi@hosseinalhayazi:~/lab13$
hosseinalhayazi@hosseinalhayazi:~/lab13$ ./lab13_3.sh 5
hosseinalhayazi@hosseinalhayazi:~/lab13$ ls
1.tmp  3.tmp  5.tmp  lab13_1.sh  lab13_3.sh  script2  script2.o
2.tmp  4.tmp  fout.txt  lab13_2.sh  lab13_4.sh  script2.c  text.txt
hosseinalhayazi@hosseinalhayazi:~/lab13$
```

```
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 two windows from a Linux desktop environment. The left window is a terminal titled 'hosseinalhayazi@hosseinalhayazi:~/lab13'. It contains the following commands and output:

```
hosseinalhayazi@hosseinalhayazi:~/lab13$ chmod +x lab13.*
hosseinalhayazi@hosseinalhayazi:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n
hosseinalhayazi@hosseinalhayazi:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
hosseinalhayazi@hosseinalhayazi:~/lab13$ ./lab13_2.sh
4
положительное
hosseinalhayazi@hosseinalhayazi:~/lab13$ ./lab13_2.sh
-2
отрицательное
hosseinalhayazi@hosseinalhayazi:~/lab13$
hosseinalhayazi@hosseinalhayazi:~/lab13$ ./lab13_3.sh 5
hosseinalhayazi@hosseinalhayazi:~/lab13$ ls
1.tmp 3.tmp 5.tmp lab13_1.sh lab13_3.sh script2 script2.o
2.tmp 4.tmp fout.txt lab13_2.sh lab13_4.sh script2.c text.txt
hosseinalhayazi@hosseinalhayazi:~/lab13$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
hosseinalhayazi@hosseinalhayazi:~/lab13$ ls
1.tmp 3.tmp 5.tmp fout.txt lab13_2.sh lab13_4.sh script2.c text.txt
2.tmp 4.tmp arhiv.tar lab13_1.sh lab13_3.sh script2 script2.o
hosseinalhayazi@hosseinalhayazi:~/lab13$
```

The right window is a file manager titled 'lab13\_4.sh'. It shows the contents of the file 'lab13\_4.sh':

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

The file manager window also shows a status bar at the bottom: 'sh Ширина таблицы: 8 Ln1, Col1 INS'.

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.