

# Операционные системы

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Кайнова Екатерина Андреевна

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

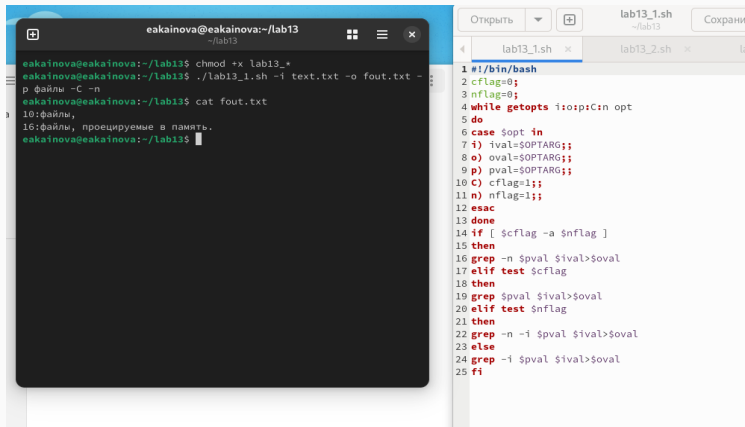
## Процесс выполнения лабораторной работы

---

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; `-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 'eakainova@eakainova:~/lab13', shows the following commands and output:

```
eakainova@eakainova:~/lab13$ chmod +x lab13_*
eakainova@eakainova:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -
р файлы -С -п
eakainova@eakainova:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
eakainova@eakainova:~/lab13$
```

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

```
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
```

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



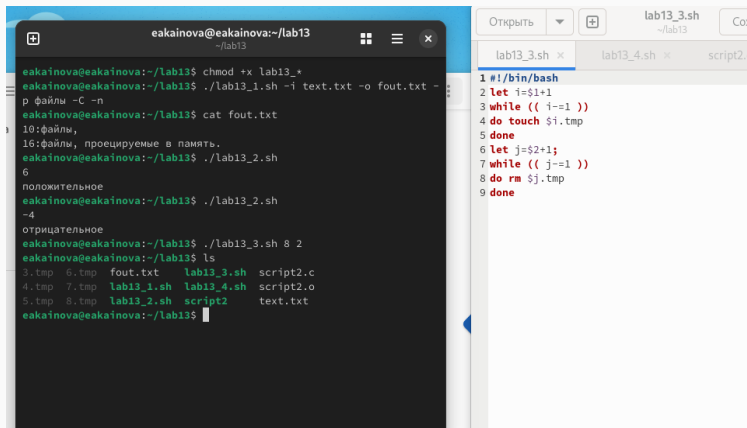
The image shows a terminal window on the left and a code editor on the right. The terminal window, titled 'eakainova@eakainova:~/lab13', shows the execution of several shell scripts. The code editor, titled 'lab13\_2.sh', shows the source code of the script being executed.

```
eakainova@eakainova:~/lab13$ chmod +x lab13_*
eakainova@eakainova:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p
р файлы -C -n
eakainova@eakainova:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
eakainova@eakainova:~/lab13$ ./lab13_2.sh
6
положительное
eakainova@eakainova:~/lab13$ ./lab13_2.sh
-4
отрицательное
eakainova@eakainova:~/lab13$
```

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

Рис. 2: Задание 2

3. Напишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N



The image shows a terminal window and a file manager interface. The terminal window, titled 'eakainova@eakainova:~/lab13', displays the following commands and output:

```
eakainova@eakainova:~/lab13$ chmod +x lab13_*
eakainova@eakainova:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -n
р файлы -С -п
eakainova@eakainova:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
eakainova@eakainova:~/lab13$ ./lab13_2.sh
6
положительное
eakainova@eakainova:~/lab13$ ./lab13_2.sh
-4
отрицательное
eakainova@eakainova:~/lab13$ ./lab13_3.sh 8 2
eakainova@eakainova:~/lab13$ ls
3.tmp 6.tmp fout.txt lab13_3.sh script2.c
4.tmp 7.tmp lab13_1.sh lab13_4.sh script2.o
5.tmp 8.tmp lab13_2.sh script2 text.txt
eakainova@eakainova:~/lab13$
```

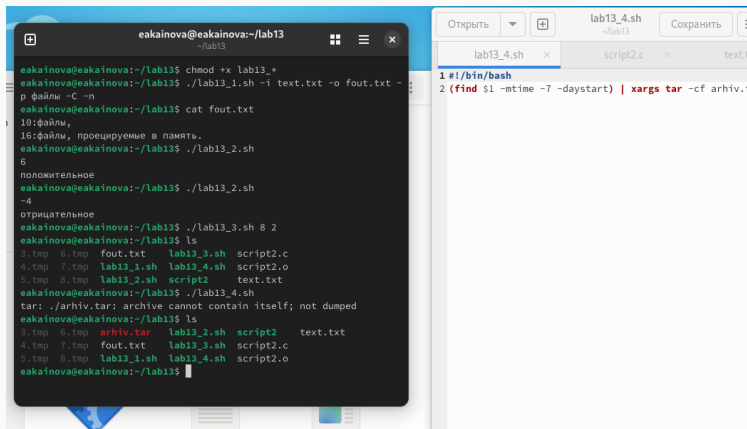
The file manager interface shows a directory named 'lab13\_3.sh' with a file 'lab13\_3.sh' selected. The file's content is displayed in a code editor:

```
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 side-by-side. The terminal window, titled 'eakainova@eakainova:~/lab13', displays the following commands and output:

```
eakainova@eakainova:~/lab13$ chmod +x lab13_*
eakainova@eakainova:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -r
р файлы -C -n
eakainova@eakainova:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
eakainova@eakainova:~/lab13$ ./lab13_2.sh
6
положительное
eakainova@eakainova:~/lab13$ ./lab13_2.sh
-4
отрицательное
eakainova@eakainova:~/lab13$ ./lab13_3.sh 8 2
eakainova@eakainova:~/lab13$ ls
3.tmp 6.tmp fout.txt lab13_3.sh script2.c
4.tmp 7.tmp lab13_1.sh lab13_4.sh script2.o
5.tmp 8.tmp lab13_2.sh script2 text.txt
eakainova@eakainova:~/lab13$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
eakainova@eakainova:~/lab13$ ls
3.tmp 6.tmp arhiv.tar lab13_2.sh script2 text.txt
4.tmp 7.tmp fout.txt lab13_3.sh script2.c
5.tmp 8.tmp lab13_1.sh lab13_4.sh script2.o
eakainova@eakainova:~/lab13$
```

The file manager window shows the contents of the 'lab13' directory. The files listed are: lab13\_4.sh, script2.c, and text.txt. The terminal window also shows the output of the 'ls' command, which lists all files in the directory, including the temporary files (3.tmp, 4.tmp, 5.tmp, 6.tmp, 7.tmp, 8.tmp) and the output files (fout.txt, script2.o, script2.c, text.txt).

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.