

# Операционные системы

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Курбанов Рахман

16 июня 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

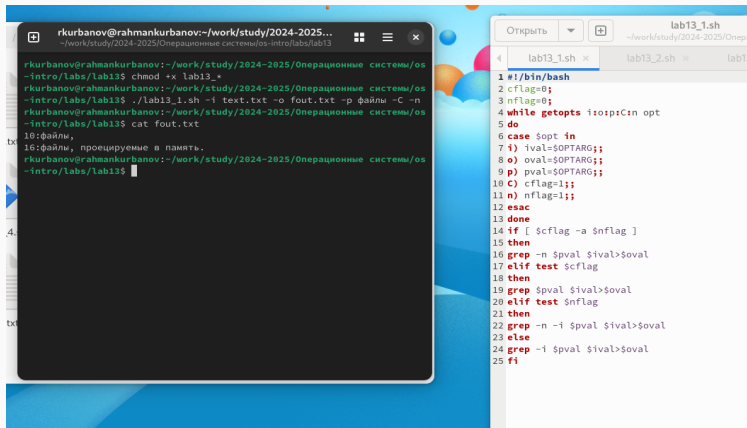
## Процесс выполнения лабораторной работы

---

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 script file on the right. The terminal window has a title bar that reads "rkurbanov@rahmankurbanov:~/work/study/2024-2025...". The terminal content shows the user navigating to a directory and running a script. The script file on the right is titled "lab13\_1.sh" and contains a shell script with various flags and conditional logic.

```
rkurbanov@rahmankurbanov:~/work/study/2024-2025...  
~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13  
  
rkurbanov@rahmankurbanov:~/work/study/2024-2025/Операционные системы/os  
-intro/labs/lab13$ chmod +x lab13_*  
rkurbanov@rahmankurbanov:~/work/study/2024-2025/Операционные системы/os  
-intro/labs/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n  
rkurbanov@rahmankurbanov:~/work/study/2024-2025/Операционные системы/os  
-intro/labs/lab13$ cat fout.txt  
10:файлы,  
16:файлы, проецируемые в память.  
rkurbanov@rahmankurbanov:~/work/study/2024-2025/Операционные системы/os  
-intro/labs/lab13$
```

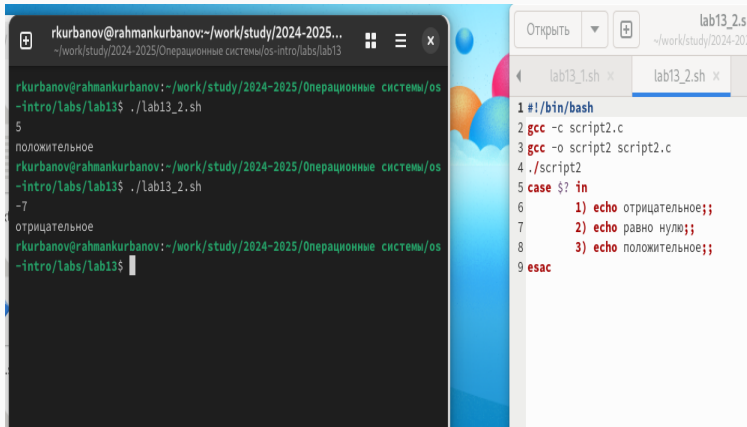
```
lab13_1.sh  
1 #!/bin/bash  
2 cflag=0;  
3 nflag=0;  
4 while getopts i:o:p:C:n opt  
5 do  
6 case $opt in  
7 i) ival=$OPTARG;;  
8 o) oval=$OPTARG;;  
9 p) pval=$OPTARG;;  
10 C) cflag=1;;  
11 n) nflag=1;;  
12 esac  
13 done  
14 if [ $cflag -a $nflag ]  
15 then  
16 grep -n $pval $ival>$oval  
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



# Выполнение работы



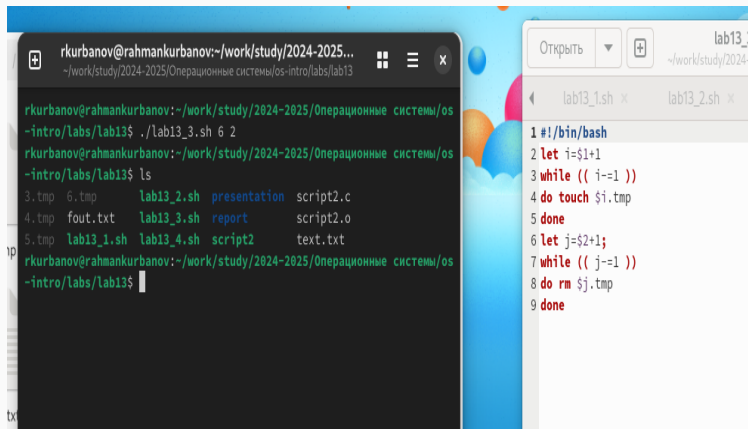
The image shows a terminal window on the left and a code editor on the right. The terminal window displays the execution of a script named `lab13_2.sh`. The script outputs the number 5, followed by the word "положительное" (positive), and then the number -7, followed by the word "отрицательное" (negative). The code editor on the right shows the source code of `lab13_2.sh`, which includes a bash shebang, compilation of `script2.c`, execution of `script2`, and a `case` statement that prints "отрицательное;;", "равно нулю;;", or "положительное;;" based on the input.

```
rkurbanov@rahmankurbanov:~/work/study/2024-2025...  
~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13  
  
rkurbanov@rahmankurbanov:~/work/study/2024-2025/Операционные системы/os  
-intro/labs/lab13$ ./lab13_2.sh  
5  
положительное  
rkurbanov@rahmankurbanov:~/work/study/2024-2025/Операционные системы/os  
-intro/labs/lab13$ ./lab13_2.sh  
-7  
отрицательное  
rkurbanov@rahmankurbanov:~/work/study/2024-2025/Операционные системы/os  
-intro/labs/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 editor. The terminal window, titled 'rkurbanov@rahmankurbanov:~/work/study/2024-2025...', displays the following commands and output:

```
rkurbanov@rahmankurbanov:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$ ./lab13_3.sh 6 2
rkurbanov@rahmankurbanov:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$ ls
3.tmp  6.tmp      lab13_2.sh  presentation  script2.c
4.tmp  fout.txt    lab13_3.sh  report        script2.o
5.tmp  lab13_1.sh  lab13_4.sh  script2       text.txt
rkurbanov@rahmankurbanov:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$
```

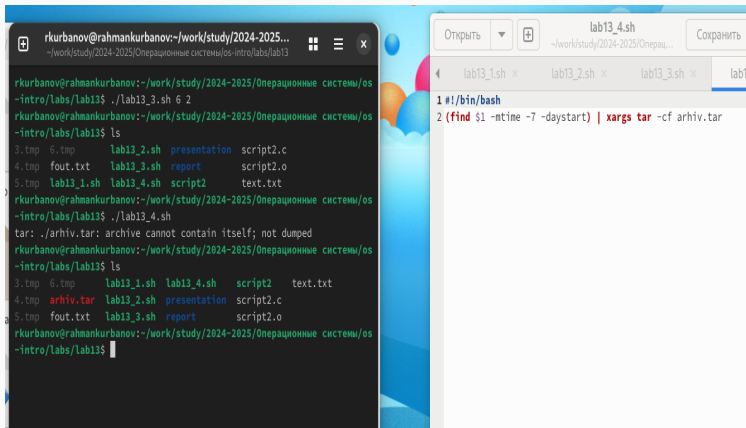
The file editor, titled 'lab13\_...', shows the content of 'lab13\_2.sh':

```
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 on the left and a file editor on the right. The terminal window displays the execution of a script named `lab13_4.sh`. The script creates a directory `3.tmp` and a file `6.tmp`, then runs `ls` to show the contents of the directory. The output of `ls` is:

```
3.tmp 6.tmp lab13_2.sh presentation script2.c
4.tmp fout.txt lab13_3.sh report script2.o
5.tmp lab13_1.sh lab13_4.sh script2 text.txt
```

The terminal then shows the execution of `tar: ./arhiv.tar: archive cannot contain itself; not dumped`. The file editor on the right shows the contents of `lab13_4.sh`, which is a shell script with two lines:

```
1 #!/bin/bash
2 (find $1 -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.