

# Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Сатторов Икромджон Абдувохидович<sup>1</sup>

20 июня, 2024, Москва, Россия

<sup>1</sup>Российский 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

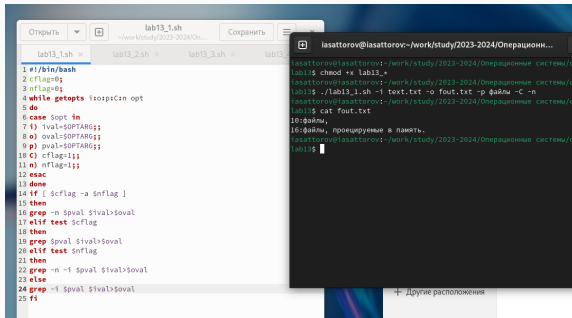
# **Процесс выполнения лабораторной работы**

---

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 screenshot shows a terminal window with a shell script named `lab13_1.sh` and its execution. The script is a shell script that takes command-line arguments and processes them. It uses `getopts` to parse options, `case` to handle different options, and `grep` to search for patterns in files. The script is executed in a terminal window, and the output is shown.

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

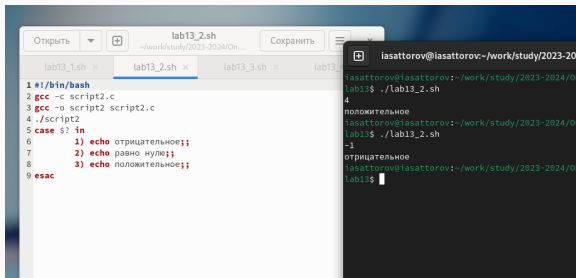
```
iasattorov@iasattorov:~/work/study/2023-2024/Операционн...
lab13$ chmod +x lab13.*
iasattorov@iasattorov:~/work/study/2023-2024/Операционные системы/оп
lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файл -C -n
iasattorov@iasattorov:~/work/study/2023-2024/Операционные системы/оп
lab13$ cat fout.txt
10:файл,
16:файл, проецируемые в память.
iasattorov@iasattorov:~/work/study/2023-2024/Операционные системы/оп
lab13$
```

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



```
lab13_2.sh
~/work/study/2023-2024/On...
Открыть Сохранить

lab13_1.sh lab13_2.sh lab13_3.sh lab13_4.sh

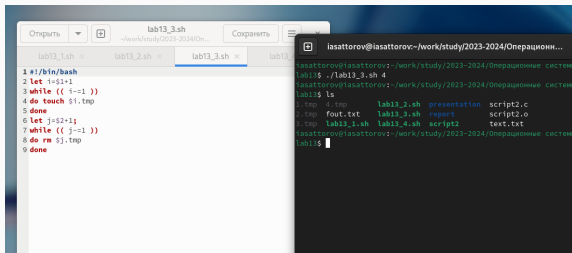
1 #!/bin/bash
2 gcc -c script2.c
3 gcc -o script2 script2.c
4 ./script2
5 case $? in
6     1) echo отрицательное;;
7     2) echo равно нулю;;
8     3) echo положительное;;
9 esac

iasattorov@iasattorov:~/work/study/2023-2024/On...$ ./lab13_2.sh
4
положительное
iasattorov@iasattorov:~/work/study/2023-2024/On...$ ./lab13_2.sh
-1
отрицательное
iasattorov@iasattorov:~/work/study/2023-2024/On...$ ./lab13_2.sh
0
равно нулю
```

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 with a light blue header bar. The window title is "lab13\_3.sh" and the path is "~/work/study/2023-2024/On...". The terminal displays a shell script with the following content:

```
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
```

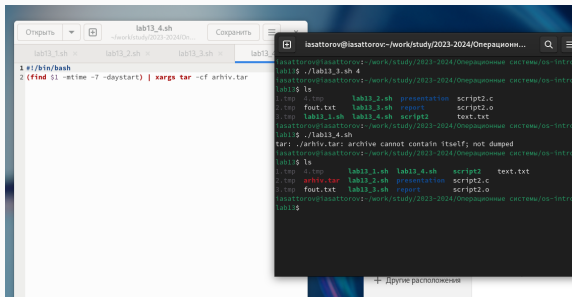
Below the script, the terminal shows the execution of the script with arguments 4 and 2:

```
iasattorov@iasattorov:~/work/study/2023-2024/Операционн...
lab13$ ./lab13_3.sh 4
iasattorov@iasattorov:~/work/study/2023-2024/Операционн...
lab13$ ls
1.tmp 4.tmp lab13_2.sh presentation script2.c
2.tmp fout.txt lab13_1.sh report script2.o
3.tmp lab13_1.sh lab13_4.sh script2 text.txt
iasattorov@iasattorov:~/work/study/2023-2024/Операционн...
lab13$
```

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 with a dark background. The prompt is `iasattorov@iasattorov:~/work/study/2023-2024/Операционные системы/os-intro`. The user has run `./lab13_3.sh 4`. The output shows a directory listing with files `presentation`, `script2.c`, `script2.o`, and `text.txt`. Then, the user runs `tar -cf arhiv.tar`, which results in an error: `tar: ./arhiv.tar: archive cannot contain itself; not dumped`. The user then runs `ls` again, showing the same files. At the bottom of the terminal window, there is a button labeled `+ Другие расположения`.

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

iasattorov@iasattorov:~/work/study/2023-2024/Операционные системы/os-intro$ ./lab13_3.sh 4
iasattorov@iasattorov:~/work/study/2023-2024/Операционные системы/os-intro$ ls
1.tap  4.tap  lab13_1.sh  lab13_2.sh  presentation  script2.c
2.tap  fout.txt lab13_3.sh  report      script2.o     text.txt
iasattorov@iasattorov:~/work/study/2023-2024/Операционные системы/os-intro$ tar -cf arhiv.tar
tar: ./arhiv.tar: archive cannot contain itself; not dumped
iasattorov@iasattorov:~/work/study/2023-2024/Операционные системы/os-intro$ ls
1.tap  4.tap  lab13_1.sh  lab13_2.sh  presentation  script2.c
2.tap  fout.txt lab13_3.sh  report      script2.o     text.txt
iasattorov@iasattorov:~/work/study/2023-2024/Операционные системы/os-intro$
iasattorov@iasattorov:~/work/study/2023-2024/Операционные системы/os-intro$
```

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.