

# Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Давыдов Сергей Арсентьевич<sup>1</sup>

21 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

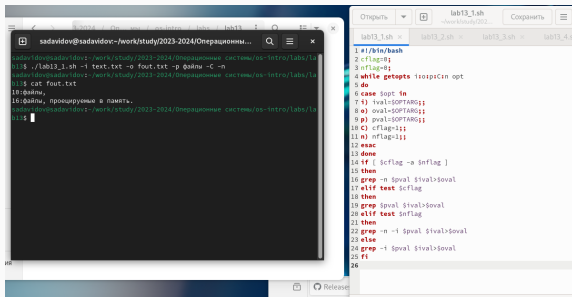
# **Процесс выполнения лабораторной работы**

---

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 image shows two overlapping windows from a Linux desktop environment. The background window is a terminal with the prompt `sadavidov@sadavidov:~/work/study/2023-2024/Операционные системы/ос-intro/labs/la`. It contains the following commands and output:

```
sadavidov@sadavidov:~/work/study/2023-2024/Операционные системы/ос-intro/labs/la
bl1$ ./lab13_1.sh -i text.txt -o fout.txt -p файл -C -n
sadavidov@sadavidov:~/work/study/2023-2024/Операционные системы/ос-intro/labs/la
bl1$ cat fout.txt
10:файл.
16:файл, прощущение в память.
sadavidov@sadavidov:~/work/study/2023-2024/Операционные системы/ос-intro/labs/la
bl1$
```

The foreground window is a text editor titled `lab13_1.sh` with the following code:

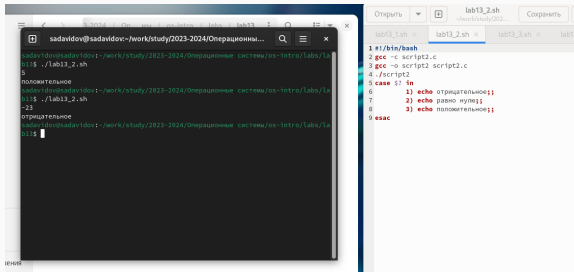
```
1 #!/bin/bash
2 cflag=0
3 nflag=0
4 while getopts isopscin opt
5 do
6 case $opt in
7 t) ival=$OPTARG;;
8 o) oval=$OPTARG;;
9 p) pval=$OPTARG;;
10 C) cflag=1;;
11 n) nflag=1;;
12 esac
13 done
14 if [ $cflag -a $nflag ]
15 then
16 grep -n $pval $ival>$oval
17 elif test $cflag
18 then
19 grep $pval $ival>$oval
20 elif test $nflag
21 then
22 grep -n -i $pval $ival>$oval
23 else
24 grep -i $pval $ival>$oval
25 fi
26 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 'script2.sh' which checks if a number is positive, negative, or zero. The code editor shows the source code of 'script2.c'.

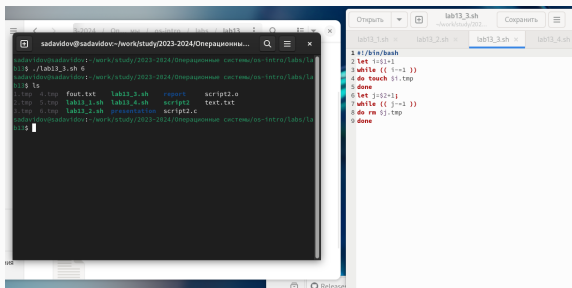
```
sadavidov@sadavidov:~/work/study/2023-2024/Операционны...  
sadavidov@sadavidov:~/work/study/2023-2024/Операционны...  
01$ ./lab13_2.sh  
5  
положительное  
sadavidov@sadavidov:~/work/study/2023-2024/Операционны...  
01$ ./lab13_2.sh  
-23  
отрицательное  
sadavidov@sadavidov:~/work/study/2023-2024/Операционны...  
01$
```

```
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 displays the execution of a shell script named `lab13_3.sh`. The script contains a loop that runs five times, each time executing a different command: `touch fout.txt`, `touch lab13_3.sh`, `touch report`, `touch script2.o`, and `touch script2.c`. The file editor on the right shows the source code of `lab13_3.sh`, which is a bash script that runs a loop from `i=1` to `i=5`, touching each of the same files as the terminal output.

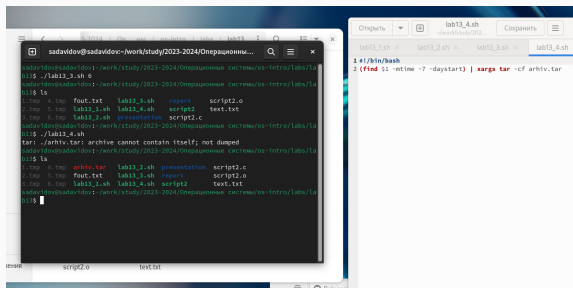
```
sadavidov@sadavidov:~/work/study/2023-2024/Операционны...  
sadavidov@sadavidov:~/work/study/2023-2024/Операционны...$ ./lab13_3.sh 6  
sadavidov@sadavidov:~/work/study/2023-2024/Операционны...$ ls  
1.tmp 4.tmp  fout.txt  lab13_3.sh  report  script2.o  
2.tmp 5.tmp  lab13_1.sh  lab13_4.sh  script2  text.txt  
3.tmp 6.tmp  lab13_2.sh  presentation  script2.c  
sadavidov@sadavidov:~/work/study/2023-2024/Операционны...$  
sadavidov@sadavidov:~/work/study/2023-2024/Операционны...$  
sadavidov@sadavidov:~/work/study/2023-2024/Операционны...$
```

```
#!/bin/bash  
1 let i=1  
2 while (( i<=5 ))  
3 do touch $i.tmp  
4 done  
5 let i=1  
6 while (( i<=5 ))  
7 do rm $i.tmp  
8 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 displays the following commands and output:

```
sadavidov@sadavidov: /work/study/2023-2024/Операционные системы/os-intro/labs/lab13
b11$ ./lab13_3.sh 6
sadavidov@sadavidov: /work/study/2023-2024/Операционные системы/os-intro/labs/lab13$ ls
1.tmp 4.tmp fout.txt lab13_3.sh report script2.o
2.tmp 5.tmp lab13_1.sh lab13_4.sh script2 text.txt
3.tmp 6.tmp lab13_2.sh presentation script2.c
sadavidov@sadavidov: /work/study/2023-2024/Операционные системы/os-intro/labs/lab13$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
sadavidov@sadavidov: /work/study/2023-2024/Операционные системы/os-intro/labs/lab13$ ls
1.tmp 4.tmp arhiv.tar lab13_2.sh presentation script2.c
2.tmp 5.tmp fout.txt lab13_3.sh report script2.o
3.tmp 6.tmp lab13_1.sh lab13_4.sh script2 text.txt
sadavidov@sadavidov: /work/study/2023-2024/Операционные системы/os-intro/labs/lab13$
```

The file manager shows the following files:

```
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.