

# Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Саруханов Артур Евгеньевич<sup>1</sup>

22 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

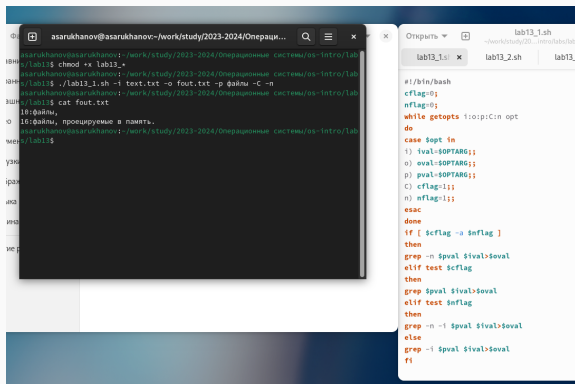
# **Процесс выполнения лабораторной работы**

---

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 a terminal window and a file editor side-by-side. The terminal window on the left shows the execution of several commands in a Linux environment. The file editor on the right shows the content of a file named 'lab13\_1.sh'.

```
asarukhanov@asarukhanov:~/work/study/2023-2024/Операционные системы/os-intro/lab13_1$ chmod +x lab13_1
asarukhanov@asarukhanov:~/work/study/2023-2024/Операционные системы/os-intro/lab13_1$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n
asarukhanov@asarukhanov:~/work/study/2023-2024/Операционные системы/os-intro/lab13_1$ cat fout.txt
10:файлы,
16:файлы, проширение в памяти.
```

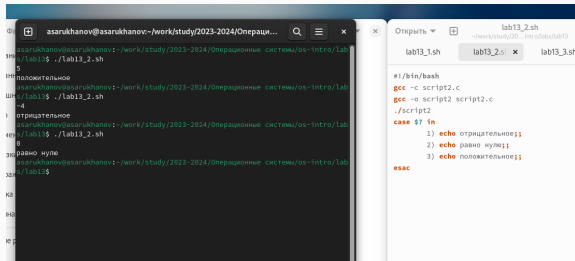
```
#!/bin/bash
cflag=0
nflag=0
while getopts fio:p:C:n opt
do
case $opt in
f) ival=$OPTARG;;
o) oval=$OPTARG;;
p) pval=$OPTARG;;
C) cflag=1;;
n) nflag=1;;
esac
done
if [ $cflag -a $nflag ]
then
grep -n $pval $ival>$oval
elif test $cflag
then
grep $pval $ival>$oval
elif test $nflag
then
grep -n -i $pval $ival>$oval
else
grep -i $pval $ival>$oval
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 two terminal windows. The left window is a standard Linux terminal with a dark background, showing the execution of a script named 'lab13\_2.sh'. The user 'asarukhanov' is in the directory '/work/study/2023-2024/Операционные системы/os-intro/lab13'. The script outputs the number '4' and then asks for input. The user enters 'отрицательное', and the script outputs 'равно нулю'. The right window is a Visual Studio Code editor showing the source code of 'lab13\_2.sh'. The code is a C program that uses 'scanf' to read an integer and 'if' statements to check if it is positive, zero, or negative, printing the corresponding message.

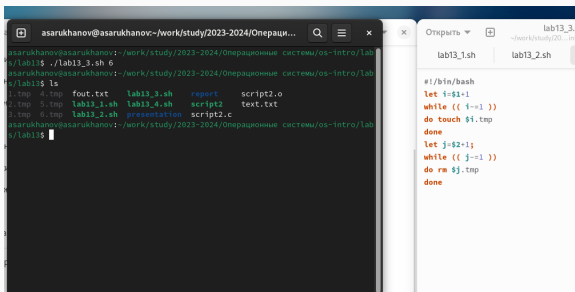
```
asarukhanov@asarukhanov:~/work/study/2023-2024/Операционные системы/os-intro/lab13$ ./lab13_2.sh
4
отрицательное
равно нулю
```

```
#!/bin/bash
gcc -c script2.c
gcc -o script2 script2.c
./script2
case $? in
  1) echo отрицательное;;
  2) echo равно нулю;;
  3) echo положительное;;
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 code editor on the right. The terminal window displays the execution of a shell script named lab13\_3.sh, which lists files and runs other scripts. The code editor shows the content of lab13\_3.sh, which is a bash script that iterates over files and runs other scripts.

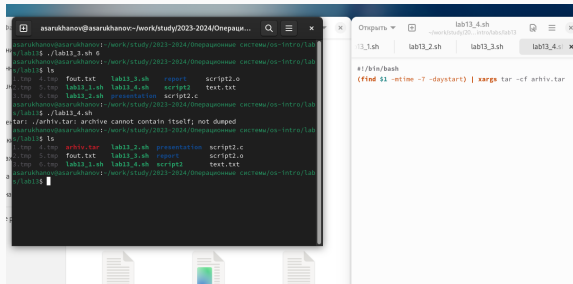
```
asarukhanov@asarukhanov:~/work/study/2023-2024/Операционные системы/os-intro/lab13$ ./lab13_3.sh 6
asarukhanov@asarukhanov:~/work/study/2023-2024/Операционные системы/os-intro/lab13$ ls
1.tmp 4.tmp fout.txt lab13_3.sh report script2.o
2.tmp 5.tmp lab13_1.sh lab13_4.sh script2 text.txt
3.tmp 6.tmp lab13_2.sh presentation script2.c
asarukhanov@asarukhanov:~/work/study/2023-2024/Операционные системы/os-intro/lab13$
```

```
#!/bin/bash
let i=$1+1
while (( i-->1 ))
do touch $i.tmp
done
let j=$2+1;
while (( j-->1 ))
do rm $j.tmp
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 displays the execution of a script named `lab13_3.sh` which creates a directory structure and files. The file manager shows the resulting files and directories.

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

The file manager shows the following structure:

- lab13\_4.sh
- lab13\_1.sh
- lab13\_2.sh
- lab13\_3.sh
- report
- script2.o
- script2.c
- script2
- 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.