

# Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Видмаер Егор Романович<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

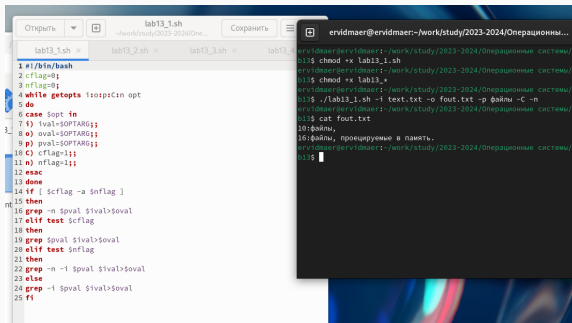
# **Процесс выполнения лабораторной работы**

---

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



```
lab13_1.sh
~/work/study/2023-2024/Операционные системы/

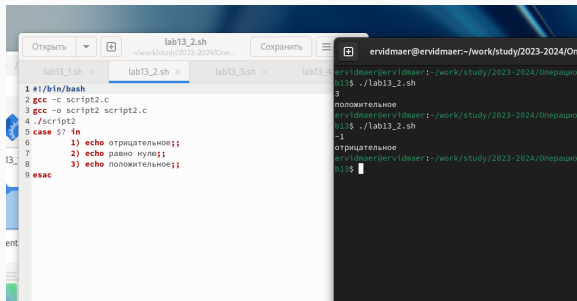
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts :ioispc:n opt
5 do
6 case $opt in
7 f) ival=$OPTARG;;
8 o) oval=$OPTARG;;
9 p) pval=$OPTARG;;
10 C) cflag=1;;
11 n) nflag=1;;
12 esac
13 done
14 if [ $cflag -a $nflag ]
15 then
16 grep -n $pval $ival:$oval
17 elif test $cflag
18 then
19 grep $pval $ival:$oval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



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

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

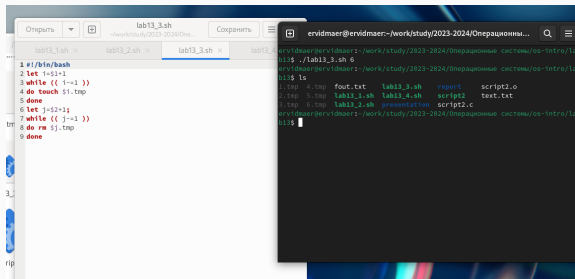
1 #!/bin/bash
2 gcc -c script2.c
3 gcc -o script2 script2.c
4 ./script2
5 case $? in
6     1) echo отрицательное;;
7     2) echo равно нулю;;
8     3) echo положительное;;
9 esac

ervidmaer@ervidmaer:~/work/study/2023-2024/Операци
b13$ ./lab13_2.sh
3
положительное
ervidmaer@ervidmaer:~/work/study/2023-2024/Операци
b13$ ./lab13_2.sh
-1
отрицательное
ervidmaer@ervidmaer:~/work/study/2023-2024/Операци
b13$
```

Рис. 2: Задание 2

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

# Выполнение работы



The image shows a code editor window with a file named `lab13_3.sh` and a terminal window running the script. The code editor has tabs for `lab13_1.sh`, `lab13_2.sh`, and `lab13_3.sh`. The terminal window shows the user `ervidmaer` at `ervidmaer@ervidmaer:~/work/study/2023-2024/Операционны...` running the script `./lab13_3.sh`. The script contains a loop that touches a file and then removes it.

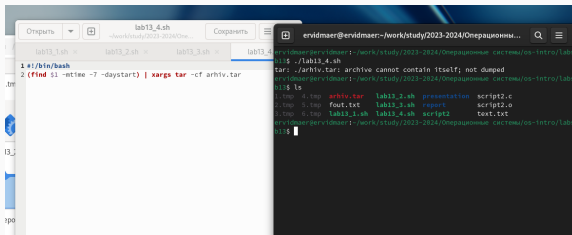
```
1 #!/bin/bash
2 let i=$1+1
3 while (( i-->0 ))
4 do touch $1.tmp
5 done
6 let j=$2+1
7 while (( j-->0 ))
8 do rm $j.tmp
9 done
```

```
ervidmaer@ervidmaer:~/work/study/2023-2024/Операционны...$ ./lab13_3.sh 6
ervidmaer@ervidmaer:~/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_1.sh  presentation  script2.c
ervidmaer@ervidmaer:~/work/study/2023-2024/Операционны...$
```

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 screenshot shows a terminal window with a dark background. The prompt is `ervidmaer@ervidmaer:~/work/study/2023-2024/Операционные системы/ос-Intro/lab13`. The user has executed the command `./lab13_4.sh`, which has produced the following output:

```
tar: ./arhiv.tar: archive cannot contain itself; not dumped
ervidmaer@ervidmaer:~/work/study/2023-2024/Операционные системы/ос-Intro/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  lab12_1.sh lab13_4.sh  script2     text.txt
ervidmaer@ervidmaer:~/work/study/2023-2024/Операционные системы/ос-Intro/lab13$
```

In the background, another terminal window is visible with the prompt `lab13_4.sh` and the command `(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.