

# Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Николай Рыбалко<sup>1</sup>

21 апреля, 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

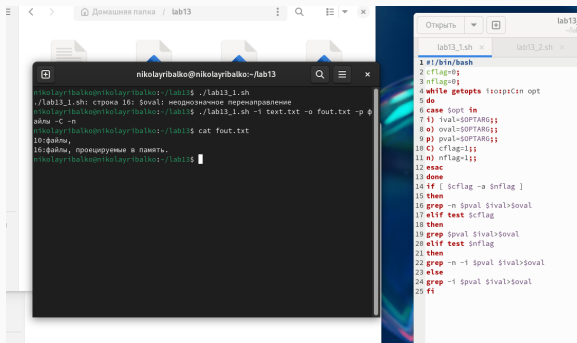
# **Процесс выполнения лабораторной работы**

---

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 Linux environment with a terminal window and a file editor. The terminal window, titled 'nikolayribalko@nikolayribalko:~/lab13', shows the execution of a script 'lab13\_1.sh'. The script's output indicates that a variable 'oval' is being used for redirection, and a file 'fout.txt' is being created and populated with file names. The file editor, titled 'lab13', shows the source code of 'lab13\_1.sh'. The script is a Bash script that sets flags, iterates over command-line arguments, and uses 'grep' to search for patterns in files.

```
nikolayribalko@nikolayribalko:~/lab13$ ./lab13_1.sh
./lab13_1.sh: строка 16: $oval: неоднозначное перенаправление
nikolayribalko@nikolayribalko:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p ф
айлы -C -n
nikolayribalko@nikolayribalko:~/lab13$ cat fout.txt
10:файлы,
16:файлы, прощериемые в памяти.
nikolayribalko@nikolayribalko:~/lab13$
```

```
#!/bin/bash
1 cflag=0
2 nflag=0
3 while getopts f:io:p:C:n opt
4 do
5 case $opt in
6 f) ival=$OPTARG;;
7 o) oval=$OPTARG;;
8 p) pval=$OPTARG;;
9 C) cflag=1;;
10 n) nflag=1;;
11 esac
12 done
13 if [ $cflag -a $nflag ]
14 then
15 grep -n $pval $ival>$oval
16 elif test $cflag
17 then
18 grep $pval $ival>$oval
19 elif test $nflag
20 then
21 grep -n -i $pval $ival>$oval
22 else
23 grep -i $pval $ival>$oval
24 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

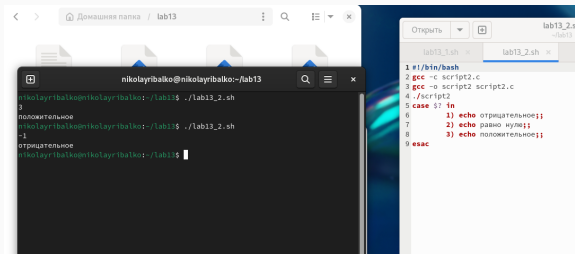


Рис. 2: Задание 2

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

# Выполнение работы

```
nikolayribalko@nikolayribalko:~/lab13$ ./lab13_3.sh 7
nikolayribalko@nikolayribalko:~/lab13$ ls
1.tmp  2.tmp  3.tmp  4.tmp  5.tmp  6.tmp  7.tmp  lab13_1.sh  lab13_2.sh  lab13_3.sh  script2  script2.o
2.tmp  4.tmp  6.tmp  fout.txt  lab13_2.sh  lab13_4.sh  script2.c  text.txt
nikolayribalko@nikolayribalko:~/lab13$
```

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

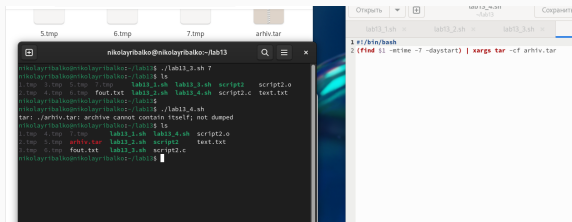


Рис. 4: Задание 4

## **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.