

# Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Алькамаль Ибрахим Мохсейн Мохаммед Али<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

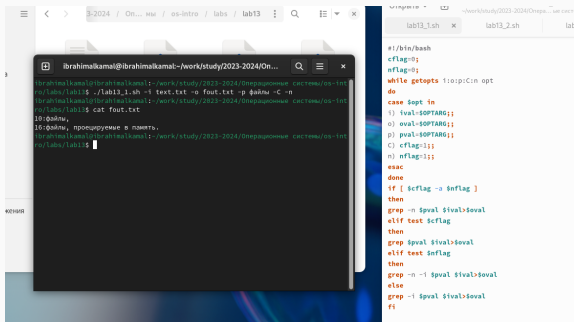
# **Процесс выполнения лабораторной работы**

---

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 code editor. The terminal window, titled 'ibrahimalkamal@ibrahimalkamal:~/work/study/2023-2024/On...', displays the following commands and output:

```
ibrahimalkamal@ibrahimalkamal:~/work/study/2023-2024/Операционные системы/os-intro/labs/lab1$ ./lab1_1.sh -i text.txt -o fout.txt -p файл -C -n
ibrahimalkamal@ibrahimalkamal:~/work/study/2023-2024/Операционные системы/os-intro/labs/lab1$ cat fout.txt
10:файл,
16:файл, проецируемые в память.
```

The code editor, titled 'vimscript', shows a shell script with the following content:

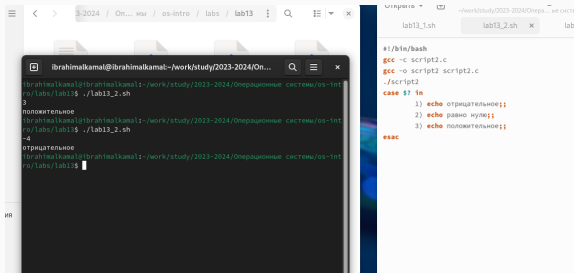
```
#!/bin/bash
cflag=0;
nflag=0;
while getopts i:op:C:n opt
do
case $opt in
i) ival=$OPTARG;;
o)  oval=$OPTARG;;
p)  pval=$OPTARG;;
C)  cflag=1;;
n)  nflag=1;;
esac
done
if [ $cflag -> $nflag ]
then
grep -n $pval $ival>$oval
elif test $cflag
then
grep $pval $ival>$oval
elif test $nflag
then
grep -n -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 a terminal window and a code editor side-by-side. The terminal window on the left displays the execution of a script named `lab13_2.sh`. The prompt is `ibrahimalkamal@ibrahimalkamal:~/work/study/2023-2024/Операционные системы/os-intro/labs/lab13$`. The script outputs `положительное` followed by the number `3`, and `отрицательное` followed by the number `-4`. The code editor on the right shows the source code of `lab13_2.sh`. It starts with a shebang `#!/bin/bash`, compiles a C program `script2.c` using `gcc`, and runs it with `./script2`. A `case` statement follows, with three branches: `1) echo отрицательное;;`, `2) echo равно нулю;;`, and `3) echo положительное;;`, ending with `esac`.

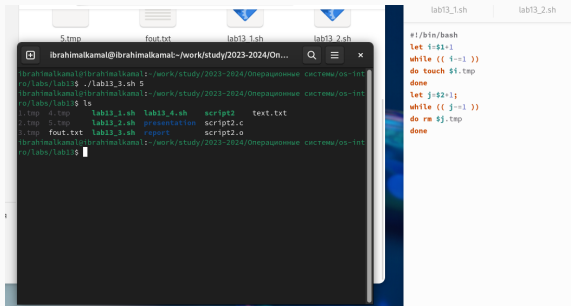
```
ibrahimalkamal@ibrahimalkamal:~/work/study/2023-2024/Операционные системы/os-intro/labs/lab13$ ./lab13_2.sh
положительное
3
ibrahimalkamal@ibrahimalkamal:~/work/study/2023-2024/Операционные системы/os-intro/labs/lab13$ ./lab13_2.sh
отрицательное
-4
ibrahimalkamal@ibrahimalkamal:~/work/study/2023-2024/Операционные системы/os-intro/labs/lab13$
```

```
#!/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 and a code editor. The terminal window, titled 'ibrahimalkamal@ibrahimalkamal:~/work/study/2023-2024/Операционные системы/os-intro/labs/lab13', displays the execution of a script 'lab13\_3.sh' with argument '5'. It lists files in the current directory and then runs a loop that creates files 'text.txt', 'script2.c', and 'script2.o' using 'touch' and 'cat' commands. The code editor shows the content of 'lab13\_1.sh' and 'lab13\_2.sh'. 'lab13\_1.sh' is a simple script that prints 'Hello, World!'. 'lab13\_2.sh' is a script that creates a file 'text.txt' and then enters an infinite loop.

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

```
#!/bin/bash
let i=$1
while (( i-- ))
do touch $i.tmp
done
let j=$2+1
while (( j-- ))
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

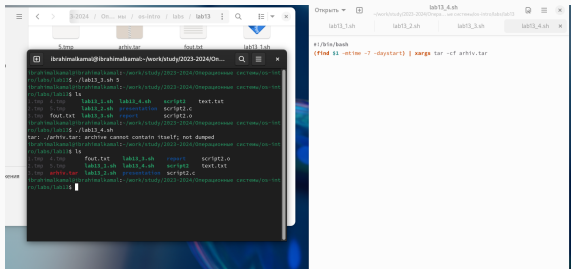


Рис. 4: Задание 4

## **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.