

# Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Евгений Алексеевич Лисягин НБИбд-01-20<sup>1</sup>

26 мая, 2021, 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

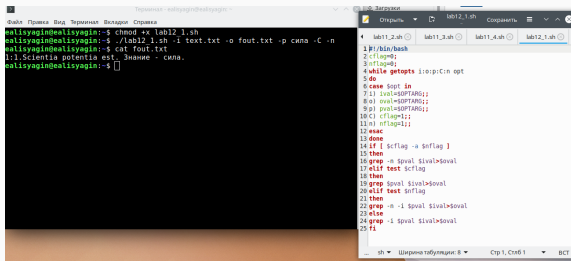
# **Процесс выполнения лабораторной работы**

---

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 with the following commands and output:

```
realisyagin@realisyagin:~$ chmod +x lab12_1.sh
realisyagin@realisyagin:~$ ./lab12_1.sh -i text.txt -o fout.txt -p сила -C -n
1:1.Scientia potentia est, Знание - сила.
realisyagin@realisyagin:~$
```

Overlaid on the terminal is a script editor window titled 'lab12\_1.sh' containing the following script:

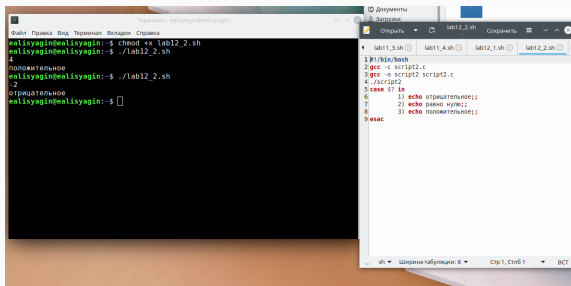
```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts :ioip:C:n opt
5 do
6 case $opt in
7 i) ival=$OPTARG;;
8 o) oval=$OPTARG;;
9 p) pval=$OPTARG;;
10 C) cflag=1;;
11 n) nflag=1;;
12 esac
13 done
14 if [ $cflag -a $nflag ]
15 then
16 grep -n $pval $ival>$oval
17 elif test $cflag
18 then
19 grep $pval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

Figure 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



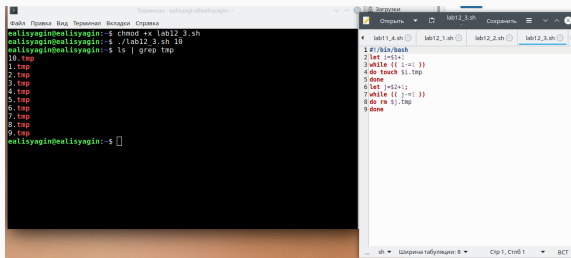
The image shows a terminal window and a code editor. The terminal window, titled 'Terminal - ealisyagin@ealisyagin', shows the execution of a shell script named 'lab12\_2.sh'. The user runs 'chmod +x lab12\_2.sh' and then './lab12\_2.sh'. The script outputs '4', 'положительное', and '-2', followed by the prompt 'отрицательное ealisyagin@ealisyagin:~\$'. The code editor, titled 'lab12\_2.sh', shows the script content:

```
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
```

Figure 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 file editor. The terminal window, titled 'Терминал: ealisyagin@ealisyagin: ~', displays the following commands and output:

```
ealisyagin@ealisyagin:~$ chmod +x lab12_3.sh
ealisyagin@ealisyagin:~$ ./lab12_3.sh 18
ealisyagin@ealisyagin:~$ ls | grep tmp
10.tmp
1.tmp
2.tmp
3.tmp
4.tmp
5.tmp
6.tmp
7.tmp
8.tmp
9.tmp
ealisyagin@ealisyagin:~$
```

The file editor, titled 'P: 2/2016', shows the content of the file 'lab12\_3.sh'. The script content is as follows:

```
1 #!/bin/bash
2 let i=i+1
3 while (( i <= 10 ))
4 do touch $i.tmp
5 done
6 let j=i2+1
7 while (( j <= 10 ))
8 do rm $j.tmp
9 done
```

Figure 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

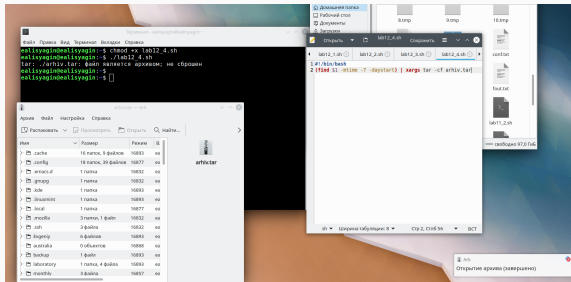


Figure 4: Задание 4

## **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.