

# Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Артем Абрикосов<sup>1</sup>

17 апреля, 2023, 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

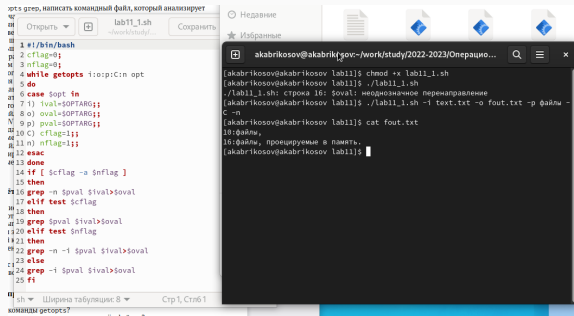
# **Процесс выполнения лабораторной работы**

---

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 desktop environment. On the left, a text editor window titled 'lab11\_1.sh' contains a shell script. The script starts with a shebang, sets flags, and uses a while loop to process command-line options. It then uses a case statement to handle different options, setting variables and flags. Finally, it uses grep to search for a pattern in a file. On the right, a terminal window shows the execution of the script. The user runs 'chmod +x lab11\_1.sh', then './lab11\_1.sh'. The terminal output shows the script's execution, including the case statement and the grep command. The user then runs 'cat fout.txt' to see the output of the script.

```
#!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:opt: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
26
27 sh  Ширина табуляции: 8  Стр 1, Стр 1
команда getopts?
какая команда используется для работы с файлами?
```

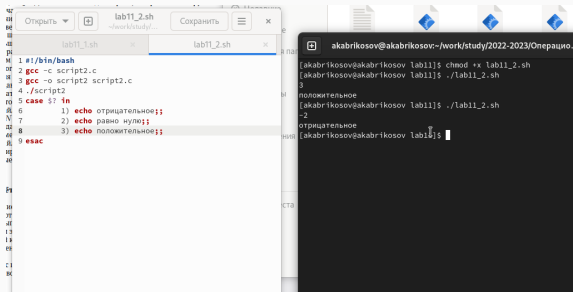
```
[akabrikosov@akabrikosov lab11]$ chmod +x lab11_1.sh
[akabrikosov@akabrikosov lab11]$ ./lab11_1.sh
./lab11_1.sh: строка 16: $oval: неоднозначное перенаправление
[akabrikosov@akabrikosov lab11]$ ./lab11_1.sh -i text.txt -o fout.txt -p файл -
C -n
[akabrikosov@akabrikosov lab11]$ cat fout.txt
16:файл,
16:файл, проецируемые в память,
[akabrikosov@akabrikosov lab11]$
```

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



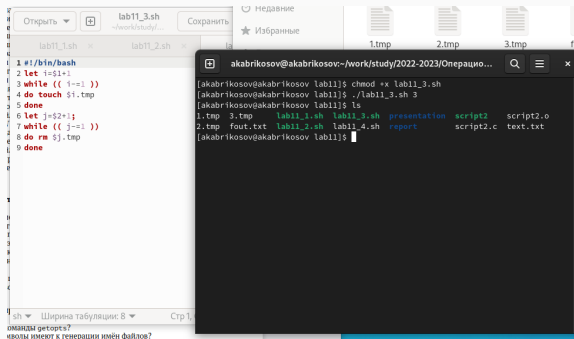
```
lab11_1.sh lab11_2.sh
1 #!/bin/bash
2 gcc -c script2.c
3 gcc -o script2 script2.c
4 ./script2
5 case $? in
6     1) echo отрицательное;;
7     2) echo равно нулю;;
8     3) echo положительное;;
9 esac

akabrikosov@akabrikosov:~/work/study/2022-2023/Операцио...
[akabrikosov@akabrikosov lab11]$ chmod +x lab11_2.sh
[akabrikosov@akabrikosov lab11]$ ./lab11_2.sh
3
положительное
[akabrikosov@akabrikosov lab11]$ ./lab11_2.sh
-2
отрицательное
[akabrikosov@akabrikosov lab11]$
```

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 with a shell script being executed. The script is named `lab11_3.sh` and is located in the directory `~/work/study/2022-2023/Операцио...`. The script's content is as follows:

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

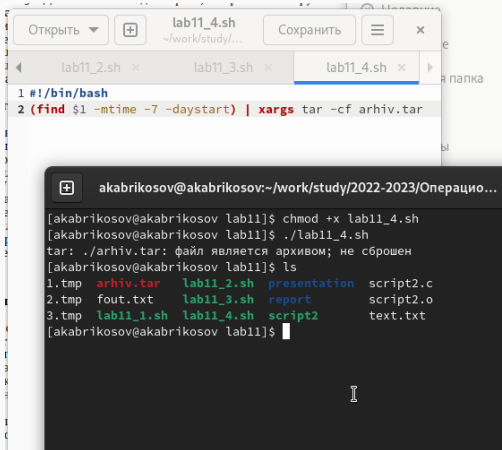
The terminal output shows the execution of the script with arguments `3` and `3`. The user runs `chmod +x lab11_3.sh` and then `./lab11_3.sh 3`. The output of the script is:

```
1.tmp 3.tmp lab11_1.sh lab11_3.sh presentation script2 script2.o
2.tmp fout.txt lab11_2.sh lab11_4.sh report script2.c text.txt
```

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 light theme. The title bar indicates the file 'lab11\_4.sh' is open. The terminal content shows the execution of a script that creates a tar archive. Below the terminal window, a dark-themed terminal window shows the user performing file operations and listing the directory contents.

```
1 #!/bin/bash
2 (find $1 -mtime -7 -daystart) | xargs tar -cf arhiv.tar
```

```
[akabrikosov@akabrikosov lab11]$ chmod +x lab11_4.sh
[akabrikosov@akabrikosov lab11]$ ./lab11_4.sh
tar: ./arhiv.tar: файл является архивом; не сброшен
[akabrikosov@akabrikosov lab11]$ ls
1.tmp  arhiv.tar  lab11_2.sh  presentation  script2.c
2.tmp  fout.txt  lab11_3.sh  report        script2.o
3.tmp  lab11_1.sh lab11_4.sh  script2       text.txt
[akabrikosov@akabrikosov lab11]$
```

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.