

# Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Бутерин Арсений<sup>1</sup>

29 августа, 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

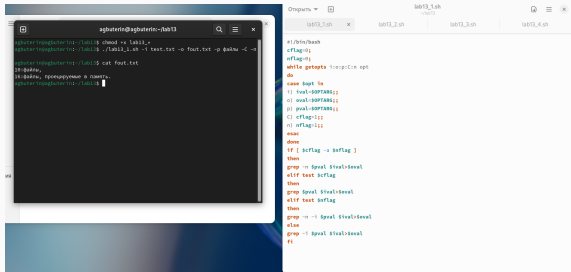
# **Процесс выполнения лабораторной работы**

---

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 displays two terminal windows. The left window, titled 'sghuterin@sghuterin:~/lab13', shows the execution of a script 'lab13.sh' which creates a file 'test.txt' and then 'fout.txt'. The right window, titled 'lab13\_1.sh', shows the source code of a script that processes flags and calls 'Spsal SivalSpsal'.

```
sghuterin@sghuterin:~/lab13
sghuterin@sghuterin:~/lab13$ chmod +x lab13.sh
sghuterin@sghuterin:~/lab13$ ./lab13_1.sh -i test.txt -o fout.txt -p fskw -C -s
sghuterin@sghuterin:~/lab13$ cat fout.txt
10-fskw,
10-fskw,
sghuterin@sghuterin:~/lab13$
```

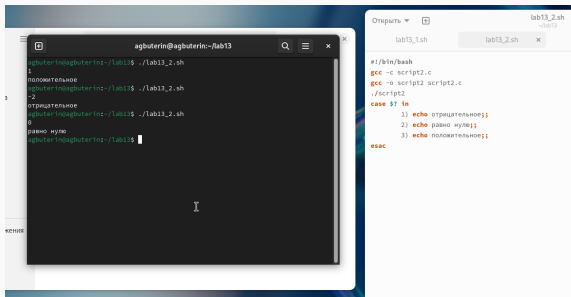
```
#!/bin/bash
cflag=0
nflag=0
while getopts "i:op:C" opt
do
    case $opt in
        i) sval=SPTAB;;
        o) oval=SPTAB;;
        p) psal=SPTAB;;
        C) cflag=1;;
        n) nflag=1;;
        *)
            echo "Error"
            exit 1
        ;;
    esac
done
if [ $cflag -eq 1 ]
then
    grep -m 1 $psal $sval
elif test $cflag
then
    grep $psal $sval
elif test $nflag
then
    grep -m 1 $psal $sval
else
    grep -l $psal $sval
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



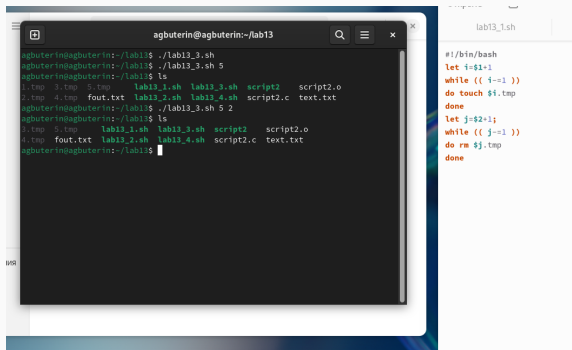
The image shows a terminal window on the left and a script file on the right. The terminal window, titled 'agbuterIn@agbuterIn:~/lab13', shows the execution of a script named 'lab13\_2.sh'. The script takes three arguments: 'положительное', '-2', and 'отрицательное'. The output of the script is '0', which is described as 'равно нулю'. The script file, titled 'lab13\_2.sh', is located in the directory 'lab13' and contains the following code:

```
#!/bin/bash
gcc -c script2.c
gcc -o script2 script2.c
./script2
case $1 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 file editor. The terminal window, titled 'agbuterin@agbuterin:~/lab13', displays the execution of a script 'lab13\_3.sh' with arguments 1 through 5. The script performs file operations: creating temporary files, copying files, and deleting files. The file editor, titled 'lab13\_1.sh', shows the content of the script being executed, which is a bash script that iterates over a range of numbers and performs file operations.

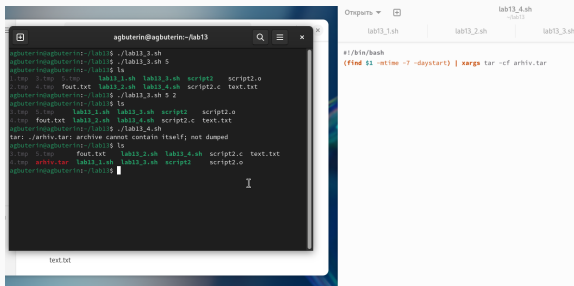
```
agbuterin@agbuterin:~/lab13$ ./lab13_3.sh
agbuterin@agbuterin:~/lab13$ ./lab13_3.sh 5
agbuterin@agbuterin:~/lab13$ ls
1.tmp 3.tmp 5.tmp lab13_1.sh lab13_3.sh script2 script2.o
2.tmp 4.tmp fout.txt lab13_2.sh lab13_4.sh script2.c text.txt
agbuterin@agbuterin:~/lab13$ ./lab13_3.sh 5 2
agbuterin@agbuterin:~/lab13$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

```
#!/bin/bash
let i=$1+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



The image shows a terminal window and a file explorer. The terminal window, titled 'agbuterin@agbuterin-lab13', displays a series of commands and their outputs. The commands include running 'lab13\_3.sh', 'ls', and 'tar'. The output shows the execution of a script that creates a file 'fout.txt' and a directory 'lab13\_4.sh'. The terminal also shows the execution of 'tar' to create an archive 'arhiv.tar'. The file explorer, titled 'lab13\_4.sh', shows the contents of the 'arhiv.tar' file, which is a bash script. The script contains a 'find' command and a 'xargs' command to create a tar archive.

```
agbuterin@agbuterin:~/lab13$ ./lab13_3.sh
agbuterin@agbuterin:~/lab13$ ./lab13_3.sh 5
agbuterin@agbuterin:~/lab13$ ls
1: 1ep 3: 1ep 5: 1ep lab13_1.sh lab13_3.sh script2 script2.o
2: 1ep 4: 1ep fout.txt lab13_2.sh lab13_4.sh script2.c text.txt
agbuterin@agbuterin:~/lab13$ ./lab13_3.sh 5 2
agbuterin@agbuterin:~/lab13$ ls
1: 1ep 3: 1ep 5: 1ep lab13_1.sh lab13_3.sh script2 script2.o
2: 1ep 4: 1ep fout.txt lab13_2.sh lab13_4.sh script2.c text.txt
agbuterin@agbuterin:~/lab13$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
agbuterin@agbuterin:~/lab13$ ls
1: 1ep 3: 1ep 5: 1ep fout.txt lab13_2.sh lab13_4.sh script2.c text.txt
2: 1ep 4: 1ep arhiv.tar lab13_1.sh lab13_3.sh script2 script2.o
agbuterin@agbuterin:~/lab13$
```

text.txt

```
#!/bin/bash
(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.