

# Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Сидельников Андрей Владимирович<sup>1</sup>

18 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

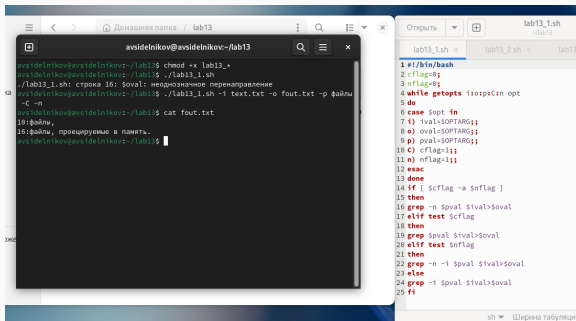
# **Процесс выполнения лабораторной работы**

---

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 terminal window and a script file. The terminal window, titled 'avsidelnikov@avsidelnikov:~/lab13', displays the following commands and output:

```
avsidelnikov@avsidelnikov:~/lab13$ chmod +x lab13_*
avsidelnikov@avsidelnikov:~/lab13$ ./lab13_1.sh
./lab13_1.sh: строка 16: $oval: неоднозначное перенаправление
avsidelnikov@avsidelnikov:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файл
-C -n
avsidelnikov@avsidelnikov:~/lab13$ cat fout.txt
10:файл,
16:файл, проецируемые в память.
avsidelnikov@avsidelnikov:~/lab13$
```

The script file, 'lab13\_1.sh', contains the following code:

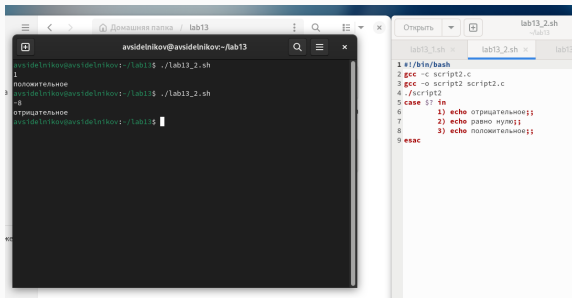
```
#!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts isopiCin opt
5 do
6 case $opt in
7 i) ival=$OPTARG;;
8 o) oval=$OPTARG;;
9 p) pval=$OPTARG;;
10 C) cflag=1;;
11 n) nflag=1;;
12 esac
13 done
14 if [ $cflag -a $nflag ]
15 then
16 grep -n $pval $ival>$oval
17 elif test $cflag
18 then
19 grep $pval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

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 code editor on the right. The terminal window, titled 'avsideInikov@avsideInikov:~/lab13', shows the execution of a script named 'lab13\_2.sh'. The script takes an argument and prints 'положительное' (positive) or 'отрицательное' (negative) based on the argument's value. The code editor on the right shows the source code of 'lab13\_2.sh'.

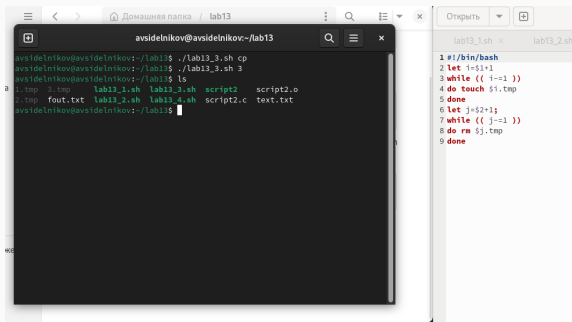
```
avsideInikov@avsideInikov:~/lab13$ ./lab13_2.sh
1
положительное
avsideInikov@avsideInikov:~/lab13$ ./lab13_2.sh
-8
отрицательное
avsideInikov@avsideInikov:~/lab13$
```

```
#!/bin/bash
1 gcc -c script2.c
2 gcc -o script2 script2.c
3 ./script2
4 case $? in
5     1) echo отрицательное;;
6     2) echo равно нулю;;
7     3) echo положительное;;
8 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 side-by-side. The terminal window, titled 'avsideInikov@avsideInikov:~/lab13', displays the execution of a script 'lab13\_3.sh'. The script performs a series of file operations: copying 'lab13\_3.sh' to itself, running 'ls', and then executing 'script2' to create 'script2.o' and 'script2.c'. It also creates 'fout.txt' and 'text.txt'. The file editor, titled 'lab13\_1.sh' and 'lab13\_2.sh', shows the source code of a shell script. The code includes a shebang, variable initialization, and two loops: a 'while' loop that increments 'i' and a 'while' loop that increments 'j' and removes files.

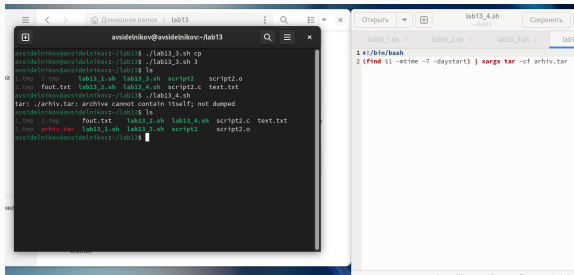
```
avsideInikov@avsideInikov:~/lab13$ ./lab13_3.sh cp
avsideInikov@avsideInikov:~/lab13$ ./lab13_3.sh 3
avsideInikov@avsideInikov:~/lab13$ ls
1.tmp  3.tmp  lab13_1.sh  lab13_3.sh  script2  script2.o
2.tmp  fout.txt lab13_2.sh  lab13_4.sh  script2.c  text.txt
avsideInikov@avsideInikov:~/lab13$
```

```
1#!/bin/bash
2let i=61i=1
3while (( i-->1 ))
4do touch $i.tmp
5done
6let j=52i=1;
7while (( j-->1 ))
8do rm $j.tmp
9done
```

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



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

The screenshot shows a terminal window with a dark background and a file manager window in the background. The terminal displays the execution of a script named lab13\_3.sh, which performs a 'cp' command and then lists files. This is followed by the execution of lab13\_4.sh, which attempts to create a tar archive but fails with the message 'archive cannot contain itself; not dumped'. After running lab13\_4.sh again, the terminal shows the successful creation of 'arhiv.tar' and lists the updated directory contents.

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.