

# Операционные системы

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Артём Арутюнян

18 апреля 2025

Российский университет дружбы народов, Москва, Россия

## Цели и задачи работы

---

Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

1 Выполнить 4 задания

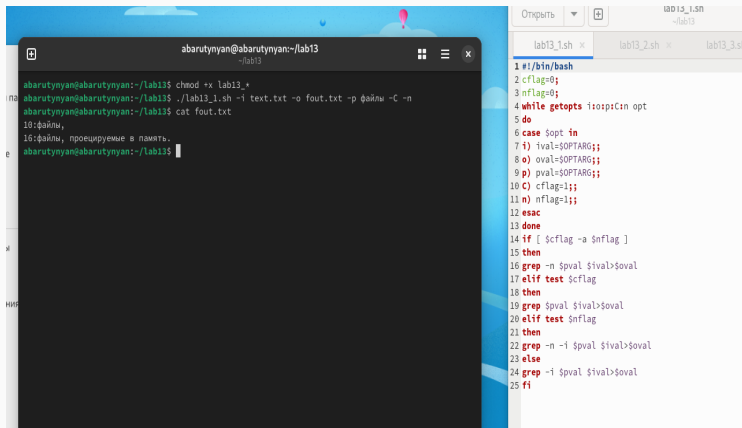
## Процесс выполнения лабораторной работы

---

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; `-p 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 on the left and a code editor on the right. The terminal window has a title bar 'abarutynyan@abarutynyan:~/lab13' and shows the following commands and output:

```
abarutynyan@abarutynyan:~/lab13$ chmod +x lab13_*
abarutynyan@abarutynyan:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n
abarutynyan@abarutynyan:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
abarutynyan@abarutynyan:~/lab13$
```

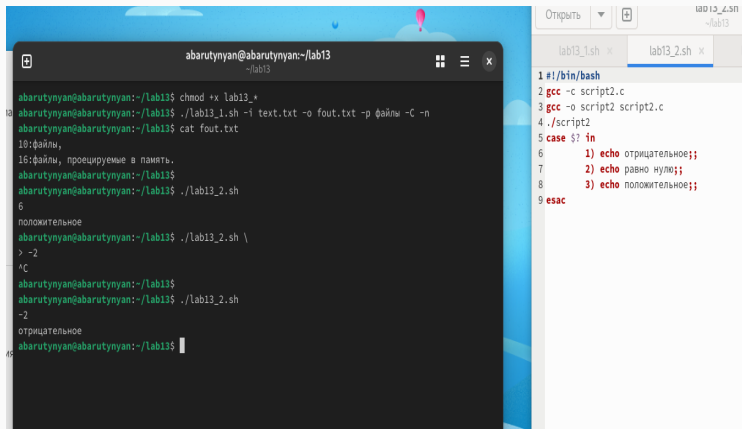
The code editor on the right shows the source code of 'lab13\_1.sh' with the following content:

```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:o:p: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
```

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





The image shows a terminal window and a code editor side-by-side. The terminal window, titled 'abarutynyan@abarutynyan:~/lab13', shows the execution of a shell script 'lab13\_2.sh'. The script uses 'cat' to read 'fout.txt' and 'echo' to output 'положительное' (positive), 'отрицательное' (negative), and '-2'. The code editor, titled 'lab13\_2.sh', shows the script's source code, which includes comments in Russian and a 'case' statement for conditional logic.

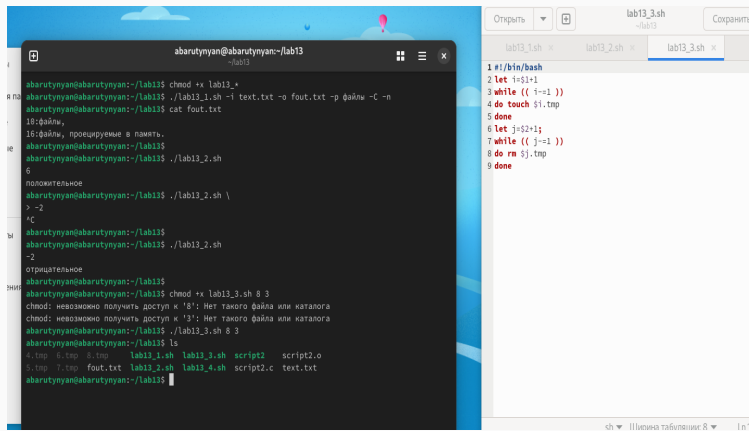
```
abarutynyan@abarutynyan:~/lab13$ chmod +x lab13_*
abarutynyan@abarutynyan:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n
abarutynyan@abarutynyan:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
abarutynyan@abarutynyan:~/lab13$ ./lab13_2.sh
6
положительное
abarutynyan@abarutynyan:~/lab13$ ./lab13_2.sh \
> -2
^C
abarutynyan@abarutynyan:~/lab13$ ./lab13_2.sh
-2
отрицательное
abarutynyan@abarutynyan:~/lab13$
```

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

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 'abarutynyan@abarutynyan:~/lab13', displays the execution of several shell scripts. It starts with 'chmod +x lab13\_\*', then runs './lab13\_1.sh' which creates 'fout.txt' and 'fout.tmp'. It then runs './lab13\_2.sh' which prints 'положительное' and '-2'. Finally, it runs './lab13\_3.sh' which prints 'отрицательное' and '-2'. The terminal also shows the execution of 'ls' and the creation of several files: '4.tmp', '6.tmp', '8.tmp', 'lab13\_1.sh', 'lab13\_3.sh', 'script2', 'script2.o', '5.tmp', '7.tmp', 'fout.txt', 'lab13\_2.sh', 'lab13\_4.sh', 'script2.c', and 'text.txt'. The file editor, titled 'lab13\_3.sh', shows the content of the script being executed in the terminal. It is a bash script that sets 'i=1', enters a 'while' loop, touches 'i.tmp', increments 'i', and then enters a 'do' loop where it increments 'j' and removes 'j.tmp' until 'j=1'.

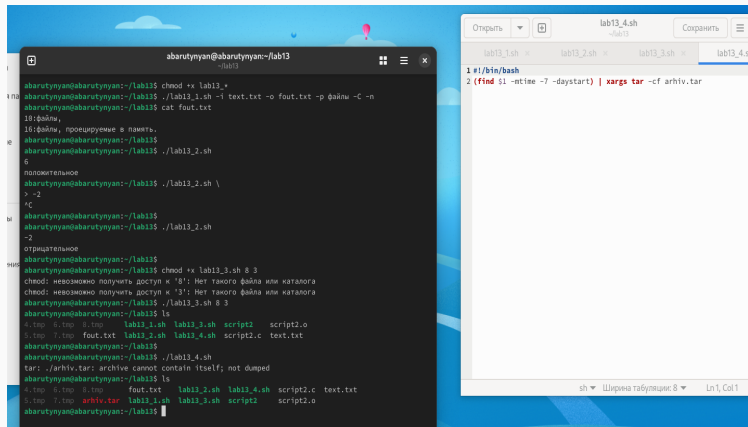
```
abarutynyan@abarutynyan:~/lab13$ chmod +x lab13_*
abarutynyan@abarutynyan:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n
abarutynyan@abarutynyan:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
abarutynyan@abarutynyan:~/lab13$ ./lab13_2.sh
6
положительное
abarutynyan@abarutynyan:~/lab13$ ./lab13_2.sh \
> -2
^C
abarutynyan@abarutynyan:~/lab13$ ./lab13_2.sh
-2
отрицательное
abarutynyan@abarutynyan:~/lab13$ ./lab13_3.sh
abarutynyan@abarutynyan:~/lab13$ chmod +x lab13_3.sh 8 3
chmod: невозможно получить доступ к '8': Нет такого файла или каталога
chmod: невозможно получить доступ к '3': Нет такого файла или каталога
abarutynyan@abarutynyan:~/lab13$ ./lab13_3.sh 8 3
abarutynyan@abarutynyan:~/lab13$ ls
4.tmp 6.tmp 8.tmp lab13_1.sh lab13_3.sh script2 script2.o
5.tmp 7.tmp fout.txt lab13_2.sh lab13_4.sh script2.c text.txt
abarutynyan@abarutynyan:~/lab13$
```

```
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
```

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 editor. The terminal window, titled 'abarutynyan@abarutynyan:~/lab13', displays the execution of several shell scripts: 'lab13\_1.sh', 'lab13\_2.sh', and 'lab13\_3.sh'. It shows file creation, directory manipulation, and an attempt to create a tar archive 'arhiv.tar' which fails with the message 'tar: ./arhiv.tar: archive cannot contain itself; not dumped'. The file editor, titled 'lab13\_4.sh', shows the content of the script being executed in the terminal, which includes a 'find' command to locate files and a 'tar' command to create the archive.

```
abarutynyan@abarutynyan:~/lab13$ chmod +x lab13_1.sh
abarutynyan@abarutynyan:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n
abarutynyan@abarutynyan:~/lab13$ cat fout.txt
10:файлы,
16:файлы, прощиряемые в память.
abarutynyan@abarutynyan:~/lab13$ ./lab13_2.sh
6
положительное
abarutynyan@abarutynyan:~/lab13$ ./lab13_2.sh \
> -2
^C
abarutynyan@abarutynyan:~/lab13$ ./lab13_2.sh
-2
отрицательное
abarutynyan@abarutynyan:~/lab13$ chmod +x lab13_3.sh 8 3
chmod: невозможно получить доступ к '8': Нет такого файла или каталога
chmod: невозможно получить доступ к '3': Нет такого файла или каталога
abarutynyan@abarutynyan:~/lab13$ ./lab13_3.sh 8 3
abarutynyan@abarutynyan:~/lab13$ ls
4.tmp 6.tmp 8.tmp  lab13_1.sh  lab13_3.sh  script2  script2.o
5.tmp 7.tmp  fout.txt  lab13_2.sh  lab13_4.sh  script2.c  text.txt
abarutynyan@abarutynyan:~/lab13$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
abarutynyan@abarutynyan:~/lab13$ ls
4.tmp 6.tmp 8.tmp  fout.txt  lab13_2.sh  lab13_4.sh  script2.c  text.txt
5.tmp 7.tmp  arhiv.tar  lab13_1.sh  lab13_3.sh  script2  script2.o
abarutynyan@abarutynyan:~/lab13$
```

```
1 #!/bin/bash
2 (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.