

# Операционные системы

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Данил Артёмов

16 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

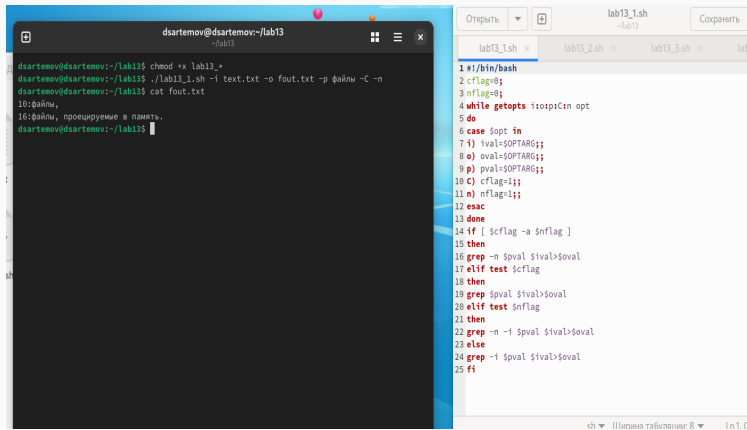
## Процесс выполнения лабораторной работы

---

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, titled 'dsartemov@dsartemov:~/lab13', shows the execution of a script 'lab13\_1.sh' with arguments '-i text.txt -o fout.txt -p файлы -C -n'. The output of the script is displayed in the terminal, showing the creation of a file 'файлы' and its contents. The code editor on the right, titled 'lab13\_1.sh', shows the source code of the script, which is a shell script that takes arguments and processes them.

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

```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:osp: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 on the left and a file editor on the right, both within a desktop environment.

**Terminal Window:** The title bar reads "dsartemov@dsartemov:~/lab13". The terminal shows the following commands and output:

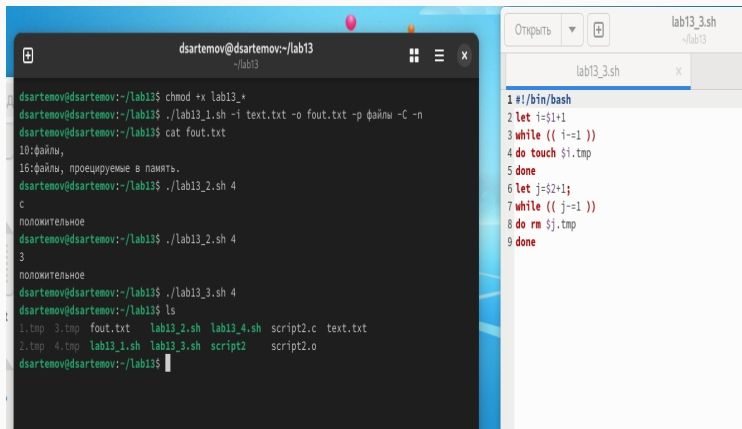
```
dsartemov@dsartemov:~/lab13$ chmod +x lab13_*
dsartemov@dsartemov:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n
dsartemov@dsartemov:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
dsartemov@dsartemov:~/lab13$ ./lab13_2.sh 4
с
положительное
dsartemov@dsartemov:~/lab13$ ./lab13_2.sh 4
3
положительное
dsartemov@dsartemov:~/lab13$
```

**File Editor:** The editor shows the content of "lab13\_2.sh". The title bar indicates the file is located at "~/lab13". The code is as follows:

```
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
```

Рис. 2: Задание 2

3. Напишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N



The image shows a terminal window and a file editor side-by-side. The terminal window, titled 'dsartemov@dsartemov:~/lab13', shows the execution of several commands: 'chmod +x lab13\_\*', './lab13\_1.sh -i text.txt -o fout.txt -p файлы -C -n', 'cat fout.txt' (outputting '10:файлы, 16:файлы, проецируемые в память.'), './lab13\_2.sh 4' (outputting 'с положительное'), './lab13\_2.sh 4' (outputting '3 положительное'), './lab13\_3.sh 4', and 'ls' (outputting a list of files including '1.tmp', '3.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: a shebang line, a loop increment, a while loop with a touch command, and a cleanup loop with a rm command.

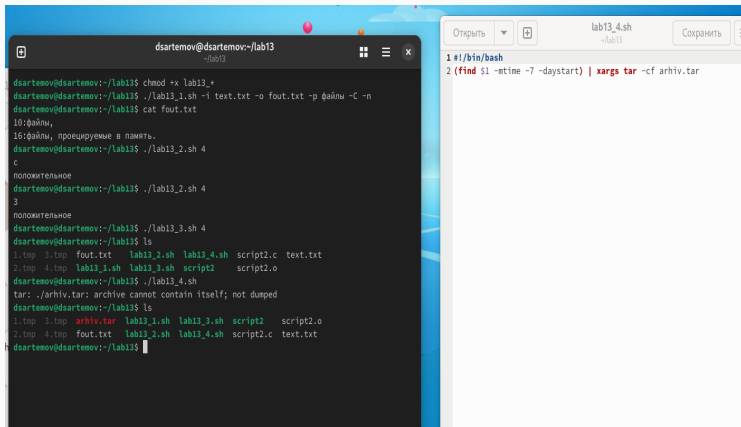
```
dsartemov@dsartemov:~/lab13$ chmod +x lab13_*
dsartemov@dsartemov:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n
dsartemov@dsartemov:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
dsartemov@dsartemov:~/lab13$ ./lab13_2.sh 4
с
положительное
dsartemov@dsartemov:~/lab13$ ./lab13_2.sh 4
3
положительное
dsartemov@dsartemov:~/lab13$ ./lab13_3.sh 4
dsartemov@dsartemov:~/lab13$ ls
1.tmp  3.tmp  fout.txt  lab13_2.sh  lab13_4.sh  script2.c  text.txt
2.tmp  4.tmp  lab13_1.sh  lab13_3.sh  script2    script2.o
dsartemov@dsartemov:~/lab13$
```

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



The image shows a terminal window and a file editor side-by-side. The terminal window, titled 'dsartemov@dsartemov:~/lab13', shows the execution of several shell scripts. The file editor, titled 'lab13\_4.sh', shows the contents of a script that uses 'find' and 'xargs' to create an archive.

```
dsartemov@dsartemov:~/lab13$ chmod +x lab13.*
dsartemov@dsartemov:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n
dsartemov@dsartemov:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
dsartemov@dsartemov:~/lab13$ ./lab13_2.sh 4
с
положительное
dsartemov@dsartemov:~/lab13$ ./lab13_2.sh 4
3
положительное
dsartemov@dsartemov:~/lab13$ ./lab13_3.sh 4
dsartemov@dsartemov:~/lab13$ ls
1.tmp 3.tmp fout.txt lab13_2.sh lab13_4.sh script2.c text.txt
2.tmp 4.tmp lab13_1.sh lab13_3.sh script2 script2.o
dsartemov@dsartemov:~/lab13$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
dsartemov@dsartemov:~/lab13$ ls
1.tmp 3.tmp arhiv.tar 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
dsartemov@dsartemov:~/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.