

Операционные системы

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Ксения Святashова

19 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.

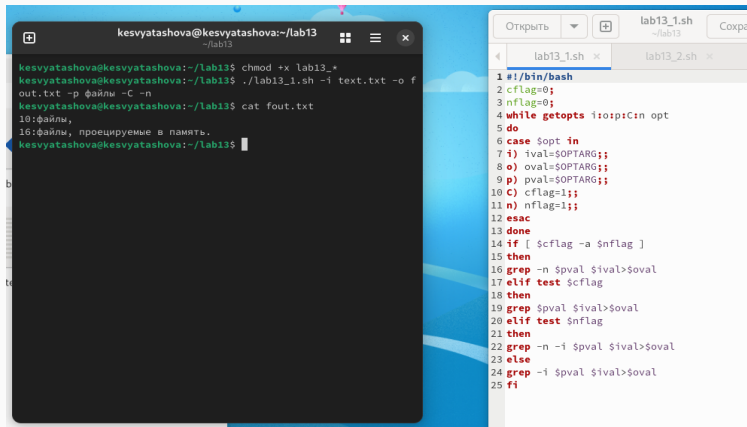
1 Выполнить 4 задания

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 two windows side-by-side. The left window is a terminal with the following content:

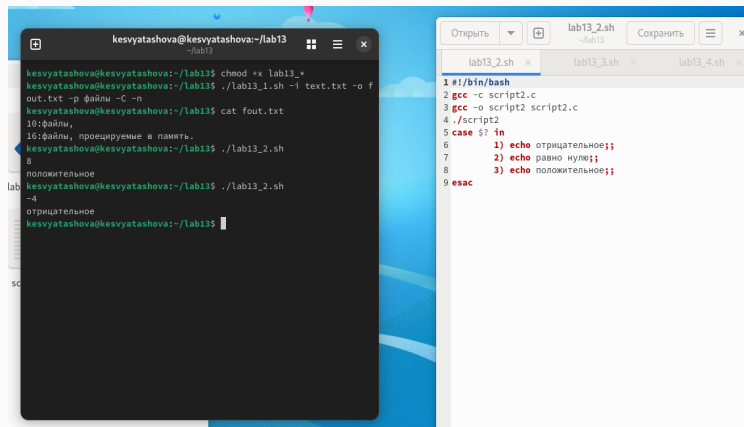
```
kesvyatashova@kesvyatashova:~/lab13
kesvyatashova@kesvyatashova:~/lab13$ chmod +x lab13.*
kesvyatashova@kesvyatashova:~/lab13$ ./lab13_1.sh -i text.txt -o f
out.txt -p файлы -C -n
kesvyatashova@kesvyatashova:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
kesvyatashova@kesvyatashova:~/lab13$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

The right window is a script editor showing the code for `lab13_1.sh`:

```
#!/bin/bash
cflag=0;
nflag=0;
while getopts i:op:C:n opt
do
case $opt in
i) ival=$OPTARG;;
o) oval=$OPTARG;;
p) pval=$OPTARG;;
C) cflag=1;;
n) nflag=1;;
esac
done
if [ $cflag -a $nflag ]
then
grep -n $pval $ival>$oval
elif test $cflag
then
grep $pval $ival>$oval
elif test $nflag
then
grep -n -i $pval $ival>$oval
else
grep -i $pval $ival>$oval
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 code editor on the right, both set against a blue background with a hot air balloon.

Terminal Window:

```
kesvyatashova@kesvyatashova:~/lab13
kesvyatashova@kesvyatashova:~/lab13$ chmod +x lab13_*
kesvyatashova@kesvyatashova:~/lab13$ ./lab13_1.sh -i text.txt -o f
out.txt -p файлы -C -n
kesvyatashova@kesvyatashova:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
kesvyatashova@kesvyatashova:~/lab13$ ./lab13_2.sh
8
положительное
kesvyatashova@kesvyatashova:~/lab13$ ./lab13_2.sh
-4
отрицательное
kesvyatashova@kesvyatashova:~/lab13$
```

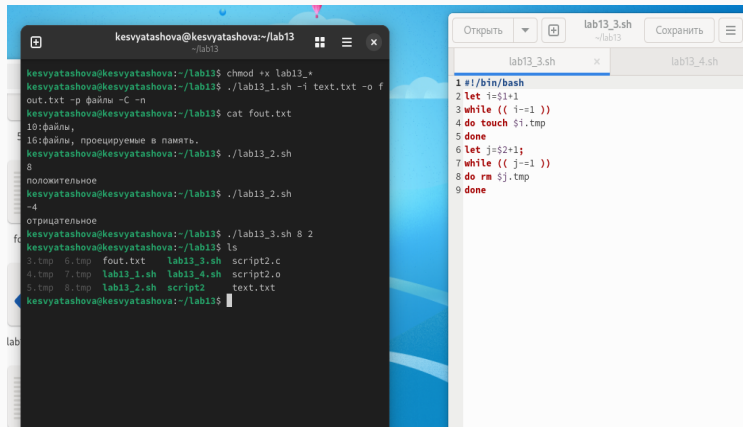
Code Editor:

lab13_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
```

Рис. 2: Задание 2

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



The image shows a terminal window on the left and a file editor on the right, both displaying shell script execution and code.

Terminal Window (Left):

```
kesvyatashova@kesvyatashova:~/lab13
~/lab13

kesvyatashova@kesvyatashova:~/lab13$ chmod +x lab13_*
kesvyatashova@kesvyatashova:~/lab13$ ./lab13_1.sh -i text.txt -o f
out.txt -p файлы -C -n
kesvyatashova@kesvyatashova:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
kesvyatashova@kesvyatashova:~/lab13$ ./lab13_2.sh
8
положительное
kesvyatashova@kesvyatashova:~/lab13$ ./lab13_2.sh
-4
отрицательное
kesvyatashova@kesvyatashova:~/lab13$ ./lab13_3.sh 8 2
kesvyatashova@kesvyatashova:~/lab13$ ls
3.tmp  6.tmp  fout.txt  lab13_3.sh  script2.c
4.tmp  7.tmp  lab13_1.sh  lab13_4.sh  script2.o
5.tmp  8.tmp  lab13_2.sh  script2    text.txt
kesvyatashova@kesvyatashova:~/lab13$
```

File Editor (Right):

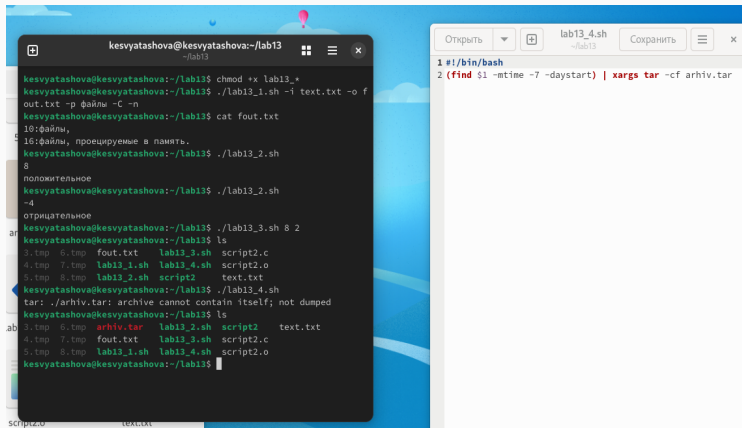
The file editor shows the content of `lab13_3.sh`:

```
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 manager window. The terminal window, titled 'kesvyatashova@kesvyatashova:~/lab13', displays the following commands and output:

```
kesvyatashova@kesvyatashova:~/lab13$ chmod +x lab13_*
kesvyatashova@kesvyatashova:~/lab13$ ./lab13_1.sh -i text.txt -o f
out.txt -r файлы -C -n
kesvyatashova@kesvyatashova:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
kesvyatashova@kesvyatashova:~/lab13$ ./lab13_2.sh
8
положительное
kesvyatashova@kesvyatashova:~/lab13$ ./lab13_2.sh
~4
отрицательное
kesvyatashova@kesvyatashova:~/lab13$ ./lab13_3.sh 8 2
kesvyatashova@kesvyatashova:~/lab13$ ls
3.tmp 6.tmp  fout.txt  lab13_3.sh  script2.c
4.tmp 7.tmp  lab13_1.sh  lab13_4.sh  script2.o
5.tmp 8.tmp  lab13_2.sh  script2    text.txt
kesvyatashova@kesvyatashova:~/lab13$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
kesvyatashova@kesvyatashova:~/lab13$ ls
3.tmp 6.tmp  arhiv.tar  lab13_2.sh  script2    text.txt
4.tmp 7.tmp  fout.txt  lab13_3.sh  script2.c
5.tmp 8.tmp  lab13_1.sh  lab13_4.sh  script2.o
kesvyatashova@kesvyatashova:~/lab13$
```

The file manager window, titled 'lab13_4.sh', shows the following commands:

```
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.