

Операционные системы

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Микаил Кандакжи

21 апреля 2025

Российский университет дружбы народов, Москва, Россия

Цели и задачи работы

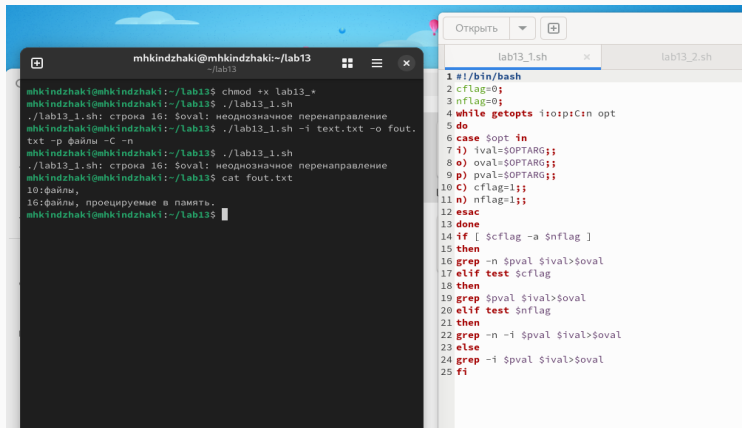
Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

1 Выполнить 4 задания

Процесс выполнения лабораторной работы

1. Используя команды `getopts` `grep` напомним 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



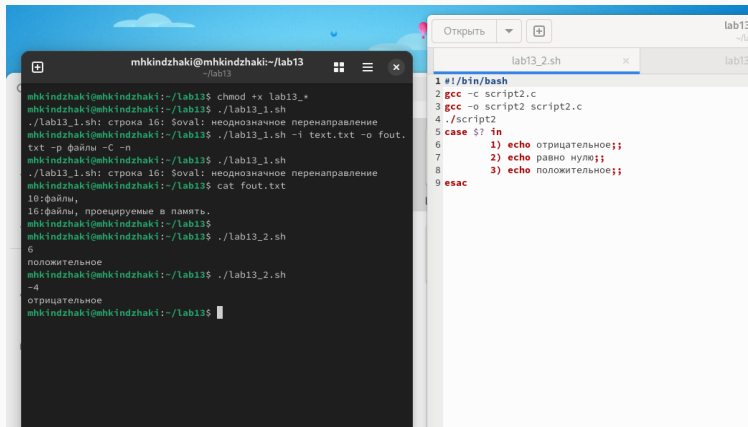
The image shows a terminal window and a code editor. The terminal window, titled 'mhkindzhaki@mhkindzhaki:~/lab13', shows the execution of a script 'lab13.sh' and its output. The code editor, titled 'lab13_1.sh', shows the source code of the script.

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

```
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



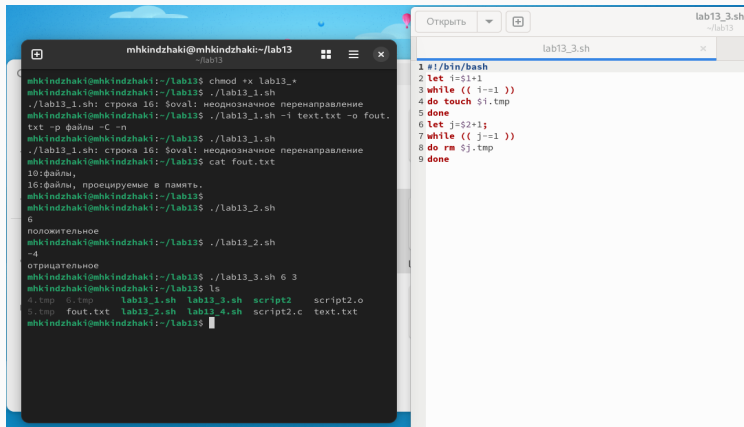
The image shows a terminal window and a code editor side-by-side. The terminal window, titled 'mhkindzhaki@mhkindzhaki:~/lab13', shows the execution of two shell scripts, 'lab13_1.sh' and 'lab13_2.sh'. The first script 'lab13_1.sh' is run with the command './lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n'. It outputs a message about an ambiguous redirection and then lists files. The second script 'lab13_2.sh' is run with the command './lab13_2.sh' and outputs '6', 'положительное', '-4', and 'отрицательное'. The code editor, titled 'lab13_2.sh', shows the source code of the second script, which is a shell script that compiles 'script2.c' and runs it with arguments '1)', '2)', and '3)' to echo negative, zero, and positive values respectively.

```
mhkindzhaki@mhkindzhaki:~/lab13$ chmod +x lab13_*
mhkindzhaki@mhkindzhaki:~/lab13$ ./lab13_1.sh
./lab13_1.sh: строка 16: $oval: неоднозначное перенаправление
mhkindzhaki@mhkindzhaki:~/lab13$ ./lab13_1.sh -i text.txt -o fout.
txt -p файлы -C -n
mhkindzhaki@mhkindzhaki:~/lab13$ ./lab13_1.sh
./lab13_1.sh: строка 16: $oval: неоднозначное перенаправление
mhkindzhaki@mhkindzhaki:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
mhkindzhaki@mhkindzhaki:~/lab13$
mhkindzhaki@mhkindzhaki:~/lab13$ ./lab13_2.sh
6
положительное
mhkindzhaki@mhkindzhaki:~/lab13$ ./lab13_2.sh
-4
отрицательное
mhkindzhaki@mhkindzhaki:~/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



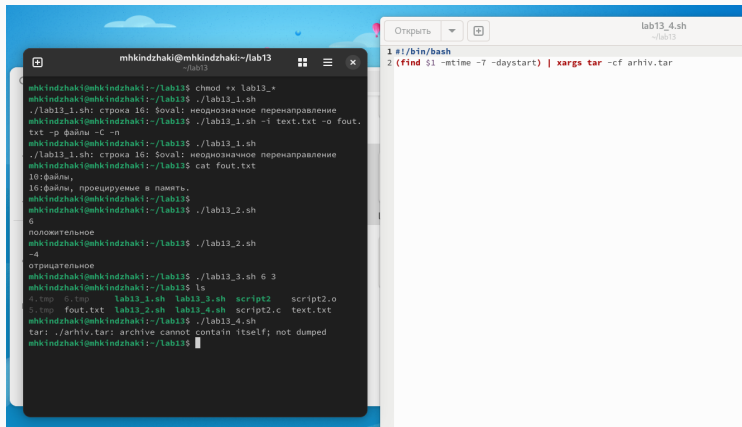
The image shows a terminal window on the left and a file editor on the right. The terminal window, titled 'mhkindzhaki@mhkindzhaki:~/lab13', shows the execution of several shell scripts: 'lab13_1.sh' (which creates 'fout.txt'), 'lab13_2.sh' (which prints '6' and '-4'), and 'lab13_3.sh' (which lists files). The file editor, titled 'lab13_3.sh', shows the source code of the script, which is a bash script that uses a while loop to create files named 'i.tmp' and 'j.tmp'.

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



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

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.