

Операционные системы

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Сиабу Мометоло Эстер

16 июня 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.

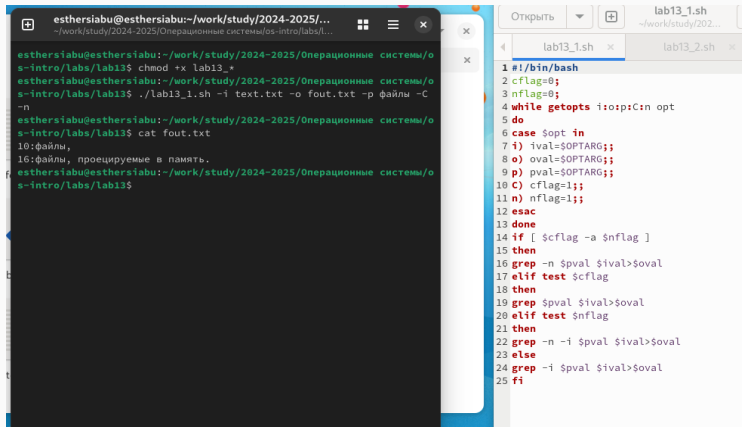
1 Выполнить 4 задания

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 displays the execution of a shell script named `lab13_1.sh`. The user `esthersiab` is in the directory `~/work/study/2024-2025/Операционные системы/os-intro/labs/L...`. The script is executed with the command `./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n`. The output of the script is displayed in the terminal, showing the contents of `text.txt` and `fout.txt`. The code editor on the right shows the source code of `lab13_1.sh`, which is a shell script that takes command-line arguments and processes them.

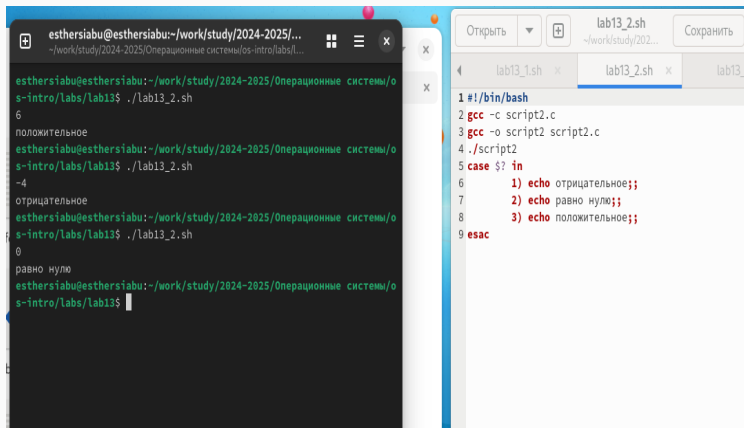
```
esthersiab@esthersiab:~/work/study/2024-2025/Операционные системы/os-intro/labs/L...  
s-intro/labs/lab13$ chmod +x lab13_*  
esthersiab@esthersiab:~/work/study/2024-2025/Операционные системы/o  
s-intro/labs/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C  
-n  
esthersiab@esthersiab:~/work/study/2024-2025/Операционные системы/o  
s-intro/labs/lab13$ cat fout.txt  
10:файлы,  
16:файлы, проецируемые в память.  
esthersiab@esthersiab:~/work/study/2024-2025/Операционные системы/o  
s-intro/labs/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

Выполнение работы



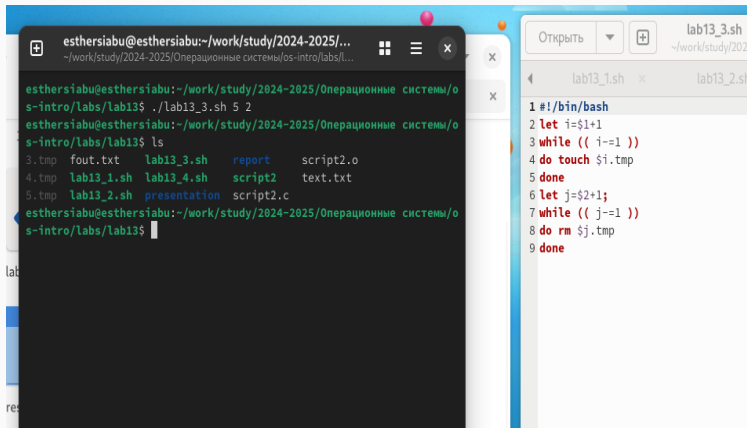
The image shows a terminal window on the left and a code editor on the right. The terminal window displays the execution of a script named `lab13_2.sh`. The user `esthersiaabu` is in the directory `~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13`. The script output shows three cases: a positive number (6), a negative number (-4), and a number equal to zero (0). The code editor on the right shows the source code of `lab13_2.sh`, which uses `gcc` to compile `script2.c` and then runs `script2` with a `case` statement to handle the input.

```
esthersiaabu@esthersiaabu:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$ ./lab13_2.sh
6
положительное
esthersiaabu@esthersiaabu:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$ ./lab13_2.sh
-4
отрицательное
esthersiaabu@esthersiaabu:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$ ./lab13_2.sh
0
равно нулю
esthersiaabu@esthersiaabu:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$
```

```
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 'esthersiaabu@esthersiaabu:~/work/study/2024-2025/...', displays the execution of a script 'lab13_3.sh' with arguments '5' and '2'. The script's output lists five temporary files: '3.tmp' (fout.txt, lab13_3.sh, report, script2.o), '4.tmp' (lab13_1.sh, lab13_4.sh, script2, text.txt), and '5.tmp' (lab13_2.sh, presentation, script2.c). The file editor, titled 'lab13_3.sh', shows the script's content, which is a bash script that iterates from i=1 to 5, creating files 'i.tmp' and then removing them in a second loop.

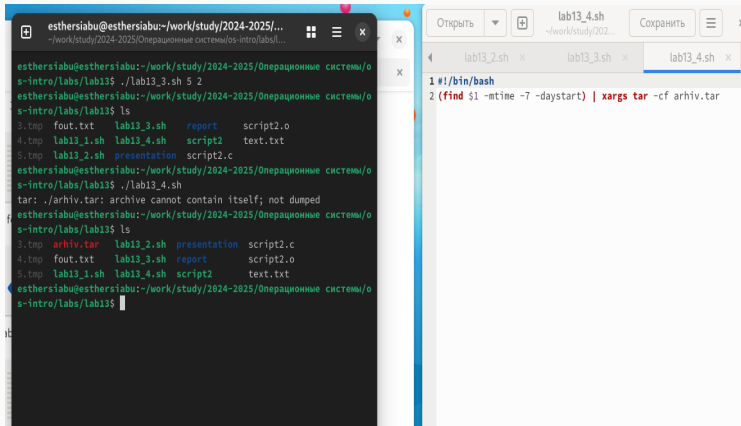
```
esthersiaabu@esthersiaabu:~/work/study/2024-2025/...  
s-intro/labs/lab13$ ./lab13_3.sh 5 2  
esthersiaabu@esthersiaabu:~/work/study/2024-2025/Операционные системы/o  
s-intro/labs/lab13$ ls  
3.tmp  fout.txt  lab13_3.sh  report    script2.o  
4.tmp  lab13_1.sh  lab13_4.sh  script2   text.txt  
5.tmp  lab13_2.sh  presentation  script2.c  
esthersiaabu@esthersiaabu:~/work/study/2024-2025/Операционные/o  
s-intro/labs/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 on the left and a file editor on the right. The terminal window displays the following commands and output:

```
esthersiaabu@esthersiaabu:~/work/study/2024-2025/Операционные системы/ос-intro/labs/...  
s-intro/labs/Lab13$ ./lab13_3.sh 5 2  
esthersiaabu@esthersiaabu:~/work/study/2024-2025/Операционные системы/ос-intro/labs/Lab13$ ls  
3.tmp  fout.txt  lab13_3.sh  report  script2.o  
4.tmp  lab13_1.sh  lab13_4.sh  script2  text.txt  
5.tmp  lab13_2.sh  presentation script2.c  
esthersiaabu@esthersiaabu:~/work/study/2024-2025/Операционные системы/ос-intro/labs/Lab13$ ./lab13_4.sh  
tar: ./arhiv.tar: archive cannot contain itself; not dumped  
esthersiaabu@esthersiaabu:~/work/study/2024-2025/Операционные системы/ос-intro/labs/Lab13$ ls  
3.tmp  arhiv.tar  lab13_2.sh  presentation script2.c  
4.tmp  fout.txt  lab13_3.sh  report  script2.o  
5.tmp  lab13_1.sh  lab13_4.sh  script2  text.txt  
esthersiaabu@esthersiaabu:~/work/study/2024-2025/Операционные системы/ос-intro/labs/Lab13$
```

The file editor on the right shows the contents of the file `lab13_4.sh`:

```
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.