

Операционные системы

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Шамес Эддин Хамза

22 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.

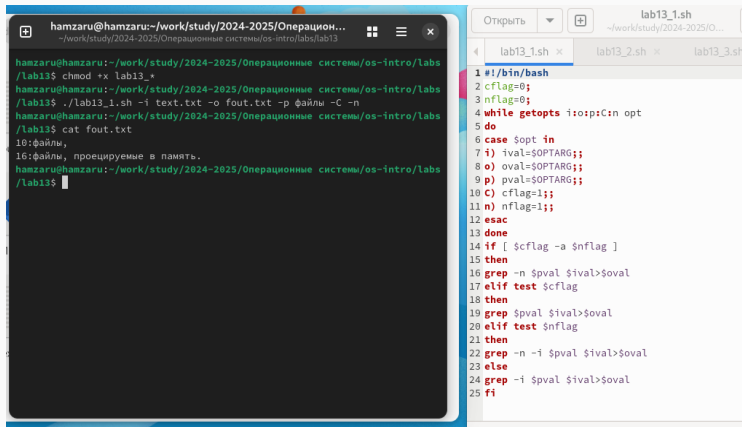
1 Выполнить 4 задания

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 script editor on the right. The terminal window displays the execution of a script named `lab13_1.sh`. The script sets permissions for `lab13_*` files, runs `./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n`, and then displays the contents of `fout.txt`. The output of the script is:

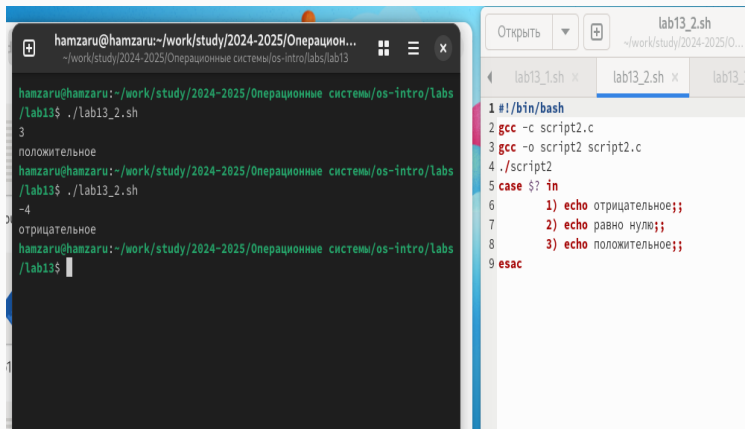
```
10:файлы,  
16:файлы, проецируемые в память.
```

The script editor on the right shows the source code of `lab13_1.sh`. The script is a shell script that uses `getopts` to parse command-line options and `grep` to search for patterns in the input file.

```
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 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 runs the script twice, and the output shows the script checking for positive and negative values. The code editor shows the source code of `lab13_2.sh`, which is a shell script that compiles a C program, runs it, and checks the exit status to print a message.

```
hamzaru@hamzaru:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$ ./lab13_2.sh
3
положительное
hamzaru@hamzaru:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$ ./lab13_2.sh
-4
отрицательное
hamzaru@hamzaru:~/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 on the left and a file editor on the right. The terminal window has a title bar with the text "hamzaru@hamzaru:~/work/study/2024-2025/Операцион...". The terminal content shows the user running a script and listing files:

```
hamzaru@hamzaru:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$ ./lab13_3.sh 6 2
hamzaru@hamzaru:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$ ls
3.tmp  6.tmp      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
hamzaru@hamzaru:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$
```

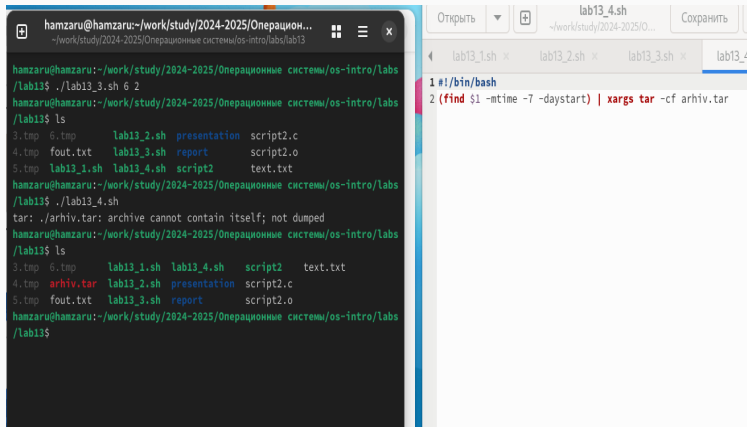
The file editor on the right has a title bar with "lab13_3.sh" and a tab labeled "lab13_3.sh". The editor content shows the following shell script code:

```
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 manager on the right. The terminal window displays the following commands and output:

```
hamzaru@hamzaru:~/work/study/2024-2025/Операцион...  
~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13  
  
hamzaru@hamzaru:~/work/study/2024-2025/Операционные системы/os-intro/labs  
/lab13$ ./lab13_3.sh 6 2  
hamzaru@hamzaru:~/work/study/2024-2025/Операционные системы/os-intro/labs  
/lab13$ ls  
3.tmp 6.tmp 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  
hamzaru@hamzaru:~/work/study/2024-2025/Операционные системы/os-intro/labs  
/lab13$ ./lab13_4.sh  
tar: ./arhiv.tar: archive cannot contain itself; not dumped  
hamzaru@hamzaru:~/work/study/2024-2025/Операционные системы/os-intro/labs  
/lab13$ ls  
3.tmp 6.tmp lab13_1.sh lab13_4.sh script2 text.txt  
4.tmp arhiv.tar lab13_2.sh presentation script2.c  
5.tmp fout.txt lab13_3.sh report script2.o  
hamzaru@hamzaru:~/work/study/2024-2025/Операционные системы/os-intro/labs  
/lab13$
```

The file manager on the right shows a tab for 'lab13_4.sh' with the following content:

```
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.