

Операционные системы

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Фатима Халилова

2025-11-14

Содержание i

1. Цели и задачи работы
2. Процесс выполнения лабораторной работы
3. Выводы по проделанной работе

1. Цели и задачи работы

1.1 Цель лабораторной работы

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

1.2 Задачи лабораторной работы

1 Выполнить 4 задания

2. Процесс выполнения лабораторной работы

2.1 Выполнение работы

1. Используя команды getopt 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; -r шаблон – указать шаблон для поиска; -C – различать большие и малые буквы; -n – выдавать номера строк;

а затем ищет в указанном файле нужные строки

2.2 Выполнение работы

The image shows a terminal window and a code editor side-by-side.

Terminal Window:

```
frhalilova@frhalilova:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab1
$ chmod +x lab13_*
frhalilova@frhalilova:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab1
$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n
frhalilova@frhalilova:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab1
$ cat fout.txt
18:файлы,
16:файлы, проецируемые в память.
frhalilova@frhalilova:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab1
$
```

Code Editor:

```
Open lab13_1.sh lab13_2.sh lab13_3.sh
1#!/bin/bash
2cflag=0;
3nflag=0;
4while getopts i:o:p:C:n opt
5do
6case $opt in
7i) ival=$OPTARG;;
8o) oval=$OPTARG;;
9p) pval=$OPTARG;;
10C) cflag=1;;
11n) nflag=1;;
12esac
13done
14if [ $cflag -a $nflag ]
15then
16grep -n $pval $ival>$oval
17elif test $cflag
18then
19grep $pval $ival>$oval
20elif test $nflag
21then
22grep -n -i $pval $ival>$oval
23else
24grep -i $pval $ival>$oval
25fi
```

Рисунок 1: Задание 1

2.3 Выполнение работы

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

2.4 Выполнение работы

The screenshot shows a terminal window with the following session:

```
frhalilova@frhalilova:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab1$ ./lab13_2.sh
3
+
положительное
frhalilova@frhalilova:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab1$ ./lab13_2.sh
-1
-
отрицательное
frhalilova@frhalilova:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab1$ ./lab13_2.sh
0
=
равно нулю
frhalilova@frhalilova:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab1$
```

To the right of the terminal, there is a code editor window titled "lab13_2.sh" which contains the following script:

```
#!/bin/bash
gcc -c script2.c
gcc -o script2 script2.c
./script2
case $? in
    1) echo отрицательное;;
    2) echo равно нулю;;
    3) echo положительное;;
esac
```

Рисунок 2: Задание 2

2.5 Выполнение работы

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

2.6 Выполнение работы

The image shows two windows side-by-side. The left window is a terminal window titled 'Home' with the path 'frhalilova@frhalilova:~/work/study/2025-2026/Операционные системы/os...'. It displays the command 'frhalilova@frhalilova:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab1\$./lab13_3.sh 5 2' followed by the output of the script. The right window is a code editor titled 'lab13_3.sh' with the path 'frhalilova@frhalilova:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab1\$'. It contains the following bash script code:

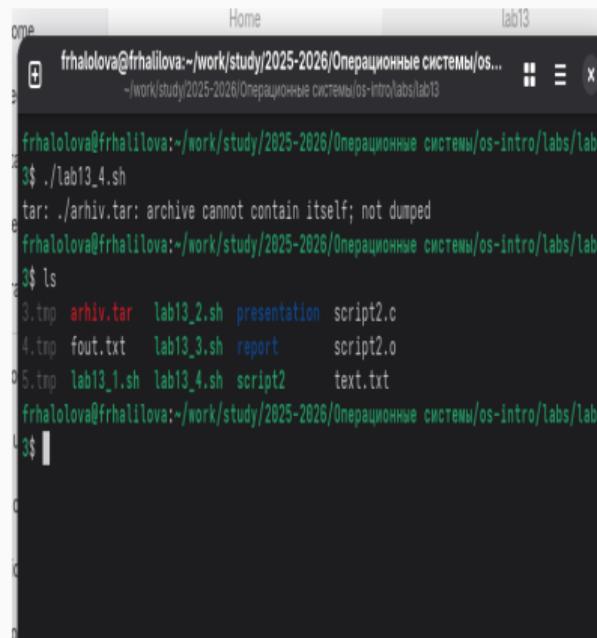
```
#!/bin/bash
let i=$1+1
while (( i=1 ))
do touch $i.tmp
done
let j=$2+1;
while (( j=1 ))
do rm $j.tmp
done
```

Рисунок 3: Задание 3

2.7 Выполнение работы

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.

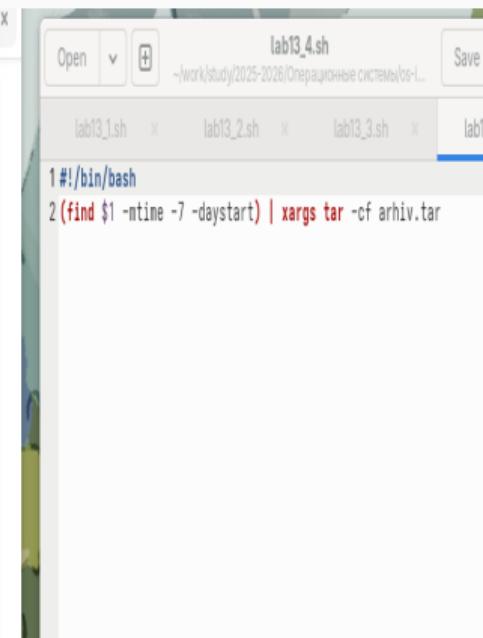
2.8 Выполнение работы



The terminal window shows the following session:

```
frhalilova@frhalilova:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab1$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
frhalilova@frhalilova:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab1$ 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
frhalilova@frhalilova:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab1$
```

The terminal window title is "Home" and the tab title is "lab13".



The code editor shows the following script content:

```
#!/bin/bash
2 (find $1 -mtime -7 -daystart) | xargs tar -cf arhiv.tar
```

The file is titled "lab13_4.sh" and is part of a project structure with other files like "lab13_1.sh", "lab13_2.sh", and "lab13_3.sh". A save dialog is visible.

Рисунок 4: Задание 4

3. Выводы по проделанной работе

3.1 Вывод

В данной работе мы изучили основы программирования в оболочке ОС UNIX и писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.