

# Операционные системы

Программирование в командном процессоре ОС UNIX. Включения и циклы

---

Вишняков Родион Сергеевич

19 сентября 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:  
`inputfile` — прочитать данные из указанного файла;  
`o outputfile` — вывести данные в указанный файл;  
`d` — шаблон — указать шаблон для поиска;  
`C` — различать большие и малые буквы;  
`n` — выдавать номера строк;

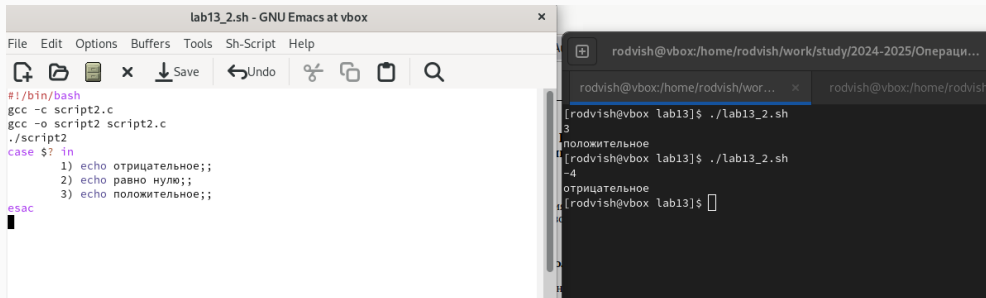
а затем ищет в указанном файле нужные строки

```
[rodvish@vbox lab13]$ chmod +x lab13_*
[rodvish@vbox lab13]$ ./lab13_1.sh -i text.txt fout.txt -p файлы -C -t
./lab13_1.sh: строка 16: $oval: неоднозначное перенаправление
[rodvish@vbox lab13]$ ./lab13_1.sh -i text.txt fout.txt -p файлы -C -n
./lab13_1.sh: строка 16: $oval: неоднозначное перенаправление
[rodvish@vbox lab13]$ ./lab13_1.sh -i text.txt fout.txt -p файлы -C -n
./lab13_1.sh: строка 16: $oval: неоднозначное перенаправление
[rodvish@vbox lab13]$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n
grep: text.txt: Нет такого файла или каталога
[rodvish@vbox lab13]$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n
[rodvish@vbox lab13]$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
[rodvish@vbox lab13]$
```

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 titled "lab13\_2.sh - GNU Emacs at vbox". The window contains the following code:

```
#!/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
```

The terminal output shows the execution of the script, which prints "3", "положительное", and "-4". The prompt is "[rodvish@vbox lab13]\$".

Рис. 2:Задание 2

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

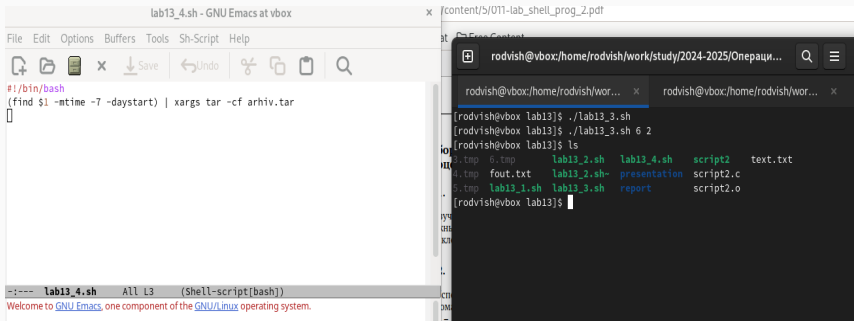
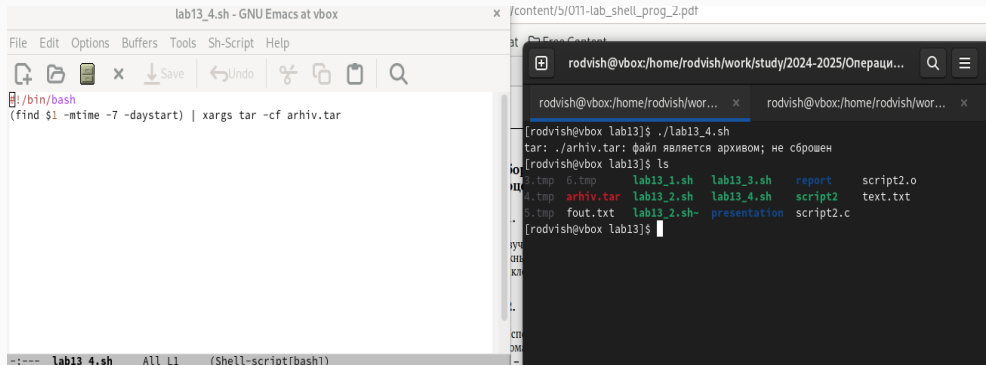


Рис. 3:Задание 3

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.



The image shows a terminal window titled "lab13\_4.sh - GNU Emacs at vbox". The terminal output is as follows:

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

[rodvish@vbox lab13]$ ./lab13_4.sh
tar: ./arhiv.tar: файл является архивом; не сброшен
[rodvish@vbox lab13]$ ls
3.tmp  6.tmp  lab13_1.sh  lab13_3.sh  report  script2.o
4.tmp  arhiv.tar  lab13_2.sh  lab13_4.sh  script2  text.txt
5.tmp  fout.txt  lab13_2.sh~  presentation  script2.c
[rodvish@vbox lab13]$
```

The terminal window also shows a menu bar with "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", and "Help". The status bar at the bottom indicates "lab13\_4.sh All L1 (Shell-script[bash])".

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.