

# Презентация по лабораторной работе №13

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Барето Вилиан Мануел

05 мая 2025

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Барето Вилиан Мануел
- НКА 03-24
- факультет физико математических и естественных наук
- Российский университет дружбы народов
- 1032239248@pfur.ru
- [https://github.com/WMBarreto/study\\_2024-2025\\_os-intro](https://github.com/WMBarreto/study_2024-2025_os-intro)

.....  
.....

## Цель работы

---

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

## Задание

---

1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку
2. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю.
3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N
4. Написать командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории.

## Выполнение лабораторной работы

---



Создаю файл `file1` и в нем написала код, который анализирует командную строку с ключами `-i` (прочитать данные из указанного файла), `-o` (вывести данные в указанный файл), `-p` (указать шаблон для поиска), `-C` (различать большие и малые буквы), `-n` (выдавать номера строк) используя команды `getopts` `grep`:

## командный файл, который анализирует командную строку

```
while getopts "i:o:p:C:n" opt
do
case $opt in
i)inputfile="$OPTARG";;
o)outputfile="$OPTARG";;
p)template="$OPTARG";;
c)register="$OPTARG";;
n)number="";;
esac
done
```

```
grep -n "$template" "$inputfile.txt" > "$outputfile.txt"
```

```
willianmanuelbarreto@willianbarreto:~$ ./file1.txt -i file1 -o output -p n etconf -C -  
n  
willianmanuelbarreto@willianbarreto:~$ cat output.txt  
willianmanuelbarreto@willianbarreto:~$
```

Рис. 1: Запуск file1

программа, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю.

Программа завершается с помощью функции `exit(n)`, передавая информацию о коде завершения в оболочку:

```
#include <stdlib.h>
#include <stdio.h>
```

```
int main()
{
    int n;
    printf("Enter a number: ");
    scanf("%d", &n);
    if(n>0)
```

программа, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю.

```
{  
    exit(1);  
}  
  
else if (n==0) {  
    exit(0);}   
else  
{  
    exit(2);  
}  
}
```

программа, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю.

Далее создала командный файл который вызывает эту программу и, проанализировав с помощью команды \$?, выдает сообщение о том, какое число было введено:

```
gcc -o cprog file2.c  
./cprog
```

```
case $? in  
0) echo "равно нулю";;  
1) echo "больше нуля";;  
2) echo "меньше нуля";;
```

```
esac
```

программа, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю.

Создала исполняемый файл и запустила:

```
willianmanuelbarreto@willianbarreto:~$ gedit file2.c
willianmanuelbarreto@willianbarreto:~$ gedit command_file.sh
willianmanuelbarreto@willianbarreto:~$ chmod +x command_file.sh
willianmanuelbarreto@willianbarreto:~$ ./command_file.sh
Enter a number: 7
больше нуля
willianmanuelbarreto@willianbarreto:~$
```

Рис. 2: Результаты программы

## командный файл, создающий указанное число файлов

Я написала командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N. Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют):

```
for((i=1; i<=$*; i++))
do
if test -f "$i".tmp
then rm "$i".tmp
else touch "$i.tmp"
fi
done
```



## командный файл, создающий указанное число файлов

Создала исполняемый файл и запустила:

```
willianmanuelbarreto@willianbarreto:~$ gedit file3.sh
willianmanuelbarreto@willianbarreto:~$ chmod +x file3.sh
willianmanuelbarreto@willianbarreto:~$ ./file3.sh 3
willianmanuelbarreto@willianbarreto:~$ ls
1.tmp          Documents      may            play
2.tmp          Downloads     memos          play_old
3.tmp          file1         misk           project1.sh
abc1           file1.txt     mkdir          project2.sh
'Área de Trabalho' file2.c       Modelos        project3.sh
australia      file3.sh      monthly       project.sh
backup         file.txt      Música         reports
bin            fun           'New Directory' ski.plases
blog           git-extended  output         ski.plases_backup.c
command_file.sh helloworld.cpp output.txt     test2.txt
config.txt     Imagens      parentdir      text.txt
conf.txt       lab07.sh     parentdir1     Transferências
cprog         letters      parentdir2     work
Desktop        LICENSE      parentdir3     work.tar.gz
Documentos     logfile      Pictures
```

Рис. 3: Создание файлов с помощью командного файла

командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории.

создала командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировала его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find).

```
find $* -mtime -7 -mtime +0 -type f > FILES.txt  
tar -cf archive.tar -T FILES.txt
```

командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории.

```
willianmanuelbarreto@willianbarreto:~$ gedit file4.sh
willianmanuelbarreto@willianbarreto:~$ chmod +x file4.sh
willianmanuelbarreto@willianbarreto:~$ ./file4.sh /home/willianbarreto/work
find: '/home/willianbarreto/work': Permissão recusada
willianmanuelbarreto@willianbarreto:~$ ./file4.sh /home/willianmanuelbarreto/work
tar: A remover "/" iniciais para nomes de membros
tar: A remover "/" iniciais de alvos rigidamente ligados
willianmanuelbarreto@willianbarreto:~$ ls ~/work
arch-pc  blog  os  study
willianmanuelbarreto@willianbarreto:~$ ls
1.tmp          Documents      logfile        play
2.tmp          Downloads     may            play_old
3.tmp          file1         memos          project1.sh
abc1           file1.txt     misk           project2.sh
archive.tar    file2.c       mkdir          project3.sh
'Área de Trabalho' file3.sh     Modelos        project.sh
australia     file4.sh     monthly        reports
backup        FILES.txt    Música          ski.plases
bin           file.txt     'New Directory' ski.plases_backup.c
blog          fun          output          test2.txt
command_file.sh git-extended  output.txt     text.txt
config.txt    helloworld.cpp parentdir       Transferências
conf.txt      Imagens      parentdir1     work
cprog        lab07.sh     parentdir2     work.tar.gz
Desktop       letters      parentdir3
Documentos    LICENSE     Pictures
willianmanuelbarreto@willianbarreto:~$
```

## Выводы

---

При выполнении проделанной работы я научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.