

# Лабораторная работа №13

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

---

Юсупова Ксения Равиловна

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Юсупова Ксения Равилевна
- Российский университет дружбы народов
- Номер студенческого билета- 1132247531
- [1132247531@pfur.ru]

## Вводная часть

---

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 с ключами:
  - `-iinputfile` — прочитать данные из указанного файла;
  - `-ooutputfile` — вывести данные в указанный файл;
  - `-р`шаблон — указать шаблон для поиска;
  - `-C` — различать большие и малые буквы;
  - `-n` — выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом `-р`.
2. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в `о` коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.

3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до `N` (например 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).
4. Написать командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду `find`).

## Выполнение лабораторной работы

---

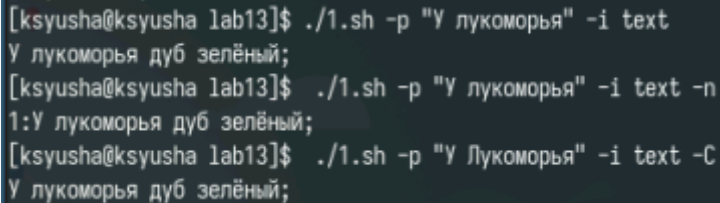


## Выполнение лабораторной работы

Напишем код для первой программы (Используя команды getoptс grep, написать командный файл, который анализирует командную строку с необходимыми ключами)

```
2.c x 2.sh x 3.sh x 4.sh x 1.sh x
1 #!/bin/bash
2
3 while getoptс ":i:o:p:Cn" opt; do
4     case ${opt} in
5         i )
6             inputfile=$OPTARG
7             ;;
8         o )
9             outputfile=$OPTARG
10            ;;
11        p )
12            pattern=$OPTARG
13            ;;
14        C )
15            case_sensitive=true
16            ;;
17        n )
18            line_numbers=true
19            ;;
20        \?)
21            echo "Неверный параметр: $OPTARG" 1>&2
22            exit 1
23            ;;
24        :)
25            echo "Отсутствует значение для параметра: $OPTARG" 1>&2
26            exit 1
27            ;;
28    esac
29done
```

Проверили код на работу

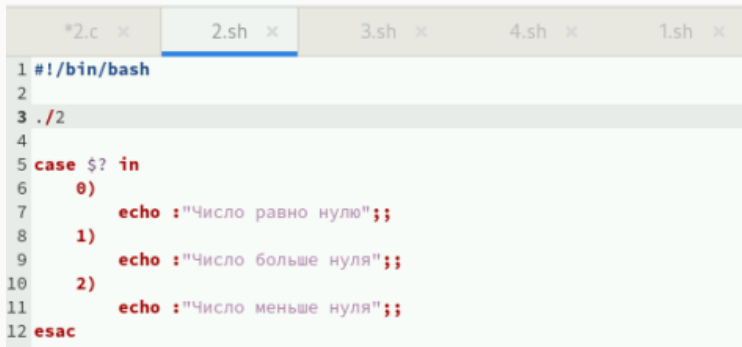


```
[ksyusha@ksyusha lab13]$ ./1.sh -p "У лукоморья" -i text
У лукоморья дуб зелёный;
[ksyusha@ksyusha lab13]$ ./1.sh -p "У лукоморья" -i text -n
1:У лукоморья дуб зелёный;
[ksyusha@ksyusha lab13]$ ./1.sh -p "У Лукоморья" -i text -C
У лукоморья дуб зелёный;
```

Рис. 2: проверили первый код

## Выполнение лабораторной работы

Напишем код для второй программы (Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю.)



```
*2.c x 2.sh x 3.sh x 4.sh x 1.sh x
1 #!/bin/bash
2
3 ./2
4
5 case $? in
6     0)
7         echo : "Число равно нулю";;
8     1)
9         echo : "Число больше нуля";;
10    2)
11        echo : "Число меньше нуля";;
12 esac
```

Рис. 3: код для второй программы

## Выполнение лабораторной работы

Написали второй код для второй программы

```
*2.c x 2.sh x 3.sh x 4.sh x 1.sh x
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5     int number;
6
7     printf("Введите число: ");
8     scanf("%d", &number);
9
10    if (number > 0) {
11        printf("Число больше нуля\n");
12        exit(1);
13    } else if (number < 0) {
14        printf("Число меньше нуля\n");
15        exit(2);
16    } else {
17        printf("Число равно нулю\n");
18        exit(0);
19    }
20 }
```

## Выполнение лабораторной работы

Проверили код на работу

```
[ksyusha@ksyusha lab13]$ ./2
Введите число: 4
Число больше нуля
[ksyusha@ksyusha lab13]$ ./2.sh
Введите число: 10
Число больше нуля
:Число больше нуля
[ksyusha@ksyusha lab13]$ ./2.sh
Введите число: -5
Число меньше нуля
:Число меньше нуля
[ksyusha@ksyusha lab13]$ ./2.sh
Введите число: 0
Число равно нулю
:Число равно нулю
```

## Выполнение лабораторной работы

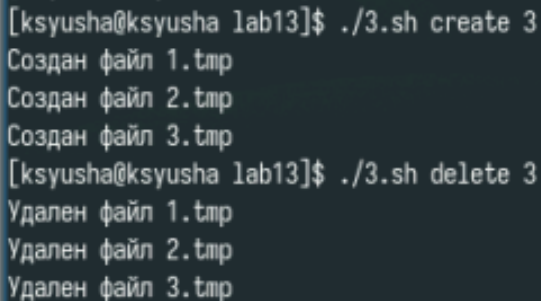
Написали код для третьей программы (Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до  $n$  (например 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.))

```
#!/bin/bash
create_files() {
    local count=$1
    for ((i=1; i<=$count; i++)); do
        touch "$i.tmp"
        echo "Создан файл $i.tmp"
    done
}

delete_files() {
    local count=$1
    for ((i=1; i<=$count; i++)); do
        if [ -e "$i.tmp" ]; then
            rm "$i.tmp"
            echo "Удален файл $i.tmp"
        fi
    done
}

if [ $# -eq 0 ]; then
    echo "Не указано количество файлов для создания"
    exit 1
fi
```

Проверили код на работу



```
[ksyusha@ksyusha lab13]$ ./3.sh create 3
Создан файл 1.tmp
Создан файл 2.tmp
Создан файл 3.tmp
[ksyusha@ksyusha lab13]$ ./3.sh delete 3
Удален файл 1.tmp
Удален файл 2.tmp
Удален файл 3.tmp
```

Рис. 7: Проверили код на работу

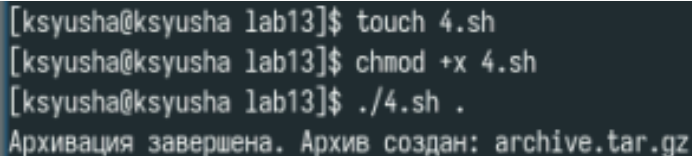
## Выполнение лабораторной работы

Написали код для четвертой программы (Написать командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории.)

```
*2.c x    2.sh x    3.sh x    4.sh x    1.sh x
1 #!/bin/bash
2
3 directory=$1
4 output_archive="archive.tar.gz"
5 threshold_days=7
6
7 if [ -z "$directory" ]; then
8     echo "Укажите директорию в качестве аргумента"
9     exit 1
10 fi
11
12 if [ ! -d "$directory" ]; then
13     echo "Указанная директория не существует"
14     exit 1
15 fi
16
17 # Используем find для поиска файлов, измененных менее чем
18     threshold_days назад,
```



Проверили код на работу

A terminal window with a dark background and light-colored text. It shows a user named ksyusha in a directory named lab13. The user runs three commands: 'touch 4.sh', 'chmod +x 4.sh', and './4.sh'. The output of the third command is 'Архивация завершена. Архив создан: archive.tar.gz'.

```
[ksyusha@ksyusha lab13]$ touch 4.sh  
[ksyusha@ksyusha lab13]$ chmod +x 4.sh  
[ksyusha@ksyusha lab13]$ ./4.sh .  
Архивация завершена. Архив создан: archive.tar.gz
```

Рис. 9: Проверили код на работу

## Выводы

---

В ходе лабораторной работы мы изучили основы программирования в оболочке ОС UNIX. Научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.