

# Отчёт о выполнении лабораторной работы №12

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

**Российский Университет Дружбы Народов**

**Факультет Физико-Математических и Естественных Наук**

**Дисциплина:** *Операционные системы*

**Работу выполняла:** *Артеева Кристина*

*1032201673*

*НКН68-01-20*

## Цель работы

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

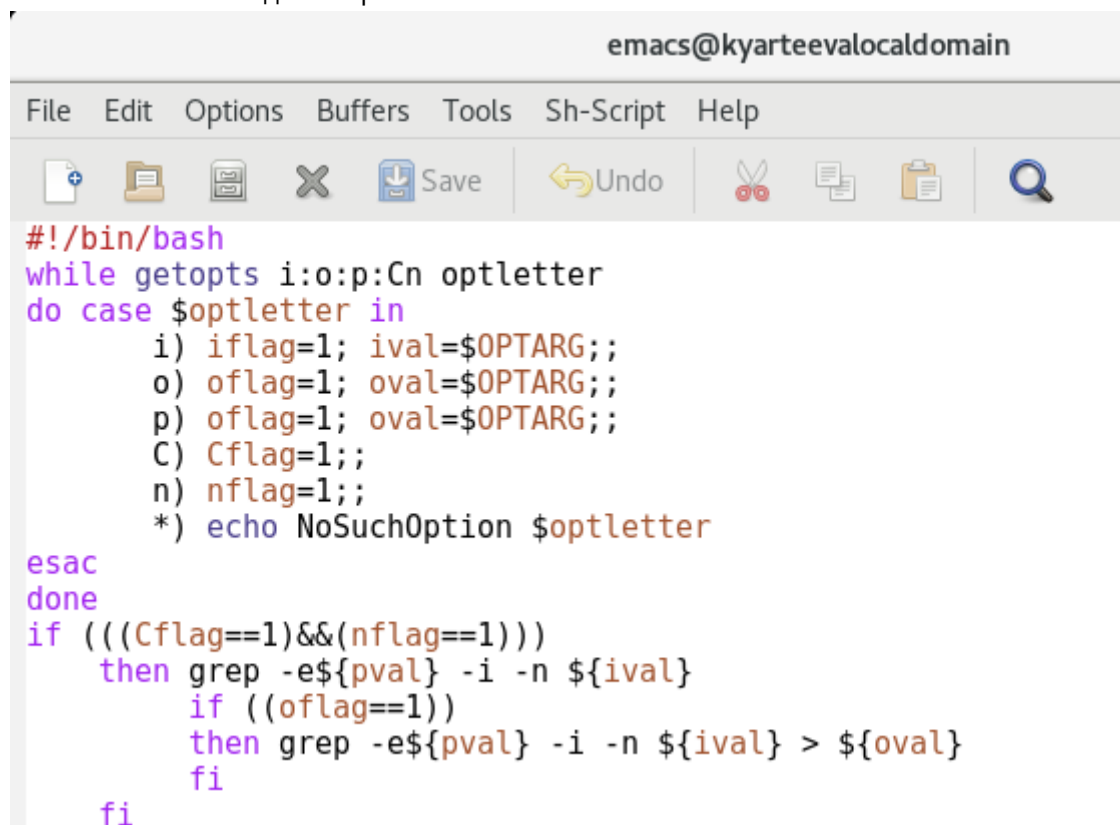
## Задание

1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами: `-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, определяемые ключом `-p`.
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 работы

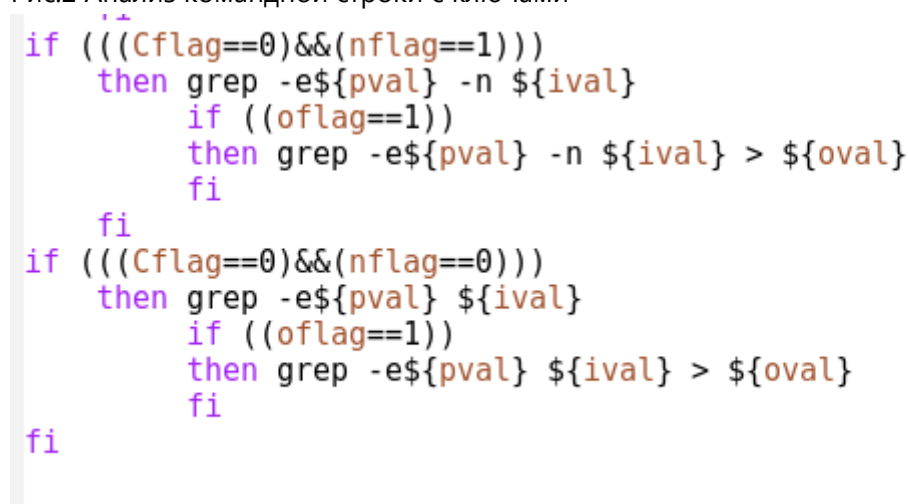
1. Используя команды `getopts` `grep`, написала командный файл, который анализирует командную строку с ключами:

Рис.1 Анализ командной строки с ключами



```
#!/bin/bash
while getopts i:o:p:Cn optletter
do case $optletter in
    i) iflag=1; ival=$OPTARG;;
    o) oflag=1; oval=$OPTARG;;
    p) oflag=1; oval=$OPTARG;;
    C) Cflag=1;;
    n) nflag=1;;
    *) echo NoSuchOption $optletter
esac
done
if (((Cflag==1)&&(nflag==1)))
then grep -e${pval} -i -n ${ival}
    if ((oflag==1))
    then grep -e${pval} -i -n ${ival} > ${oval}
    fi
fi
```

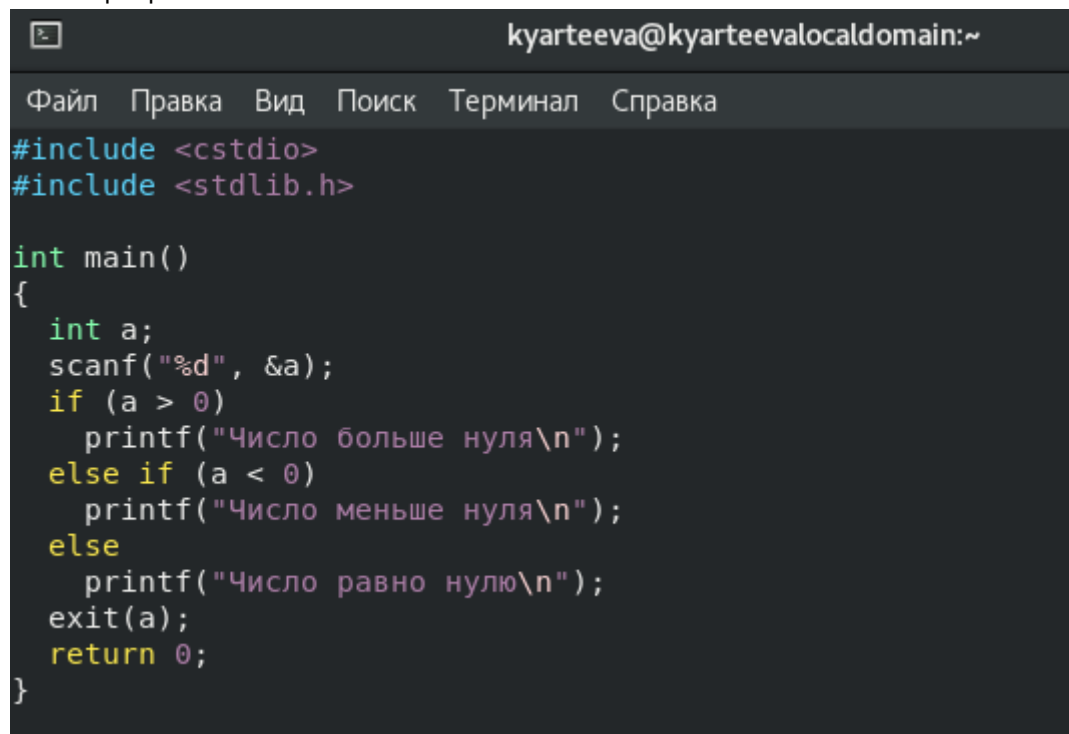
Рис.2 Анализ командной строки с ключами



```
if (((Cflag==0)&&(nflag==1)))
then grep -e${pval} -n ${ival}
    if ((oflag==1))
    then grep -e${pval} -n ${ival} > ${oval}
    fi
fi
if (((Cflag==0)&&(nflag==0)))
then grep -e${pval} ${ival}
    if ((oflag==1))
    then grep -e${pval} ${ival} > ${oval}
    fi
fi
```

- Написала 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.

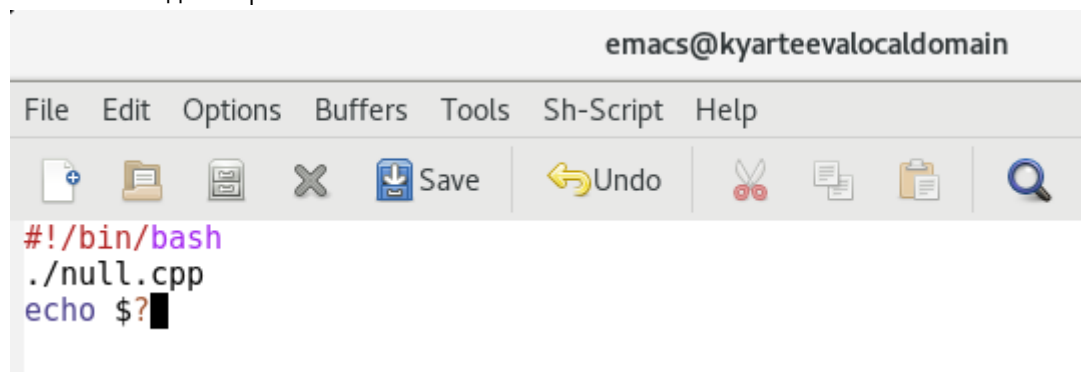
Рис.3 Программа на Си

A screenshot of a terminal window with a dark background. The title bar shows 'kyarteeva@kyarteevalocaldomain:~'. The menu bar includes 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', and 'Справка'. The code is written in C and checks if an input number is greater than, less than, or equal to zero, printing the result in Russian. The code is as follows:

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

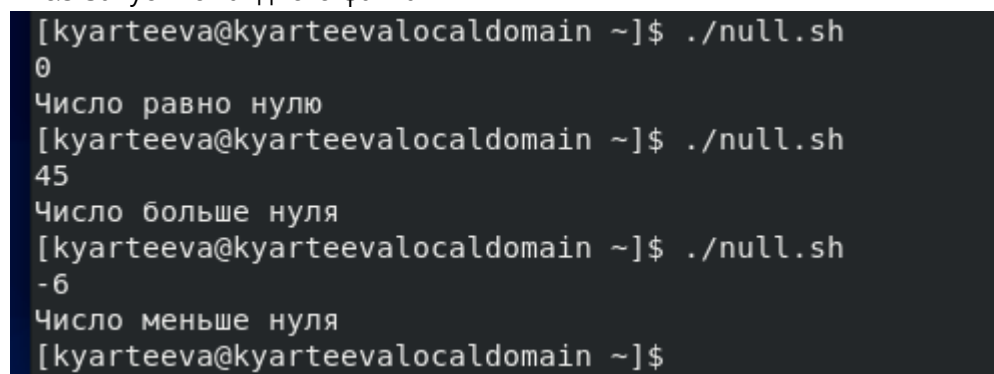
int main()
{
    int a;
    scanf("%d", &a);
    if (a > 0)
        printf("Число больше нуля\n");
    else if (a < 0)
        printf("Число меньше нуля\n");
    else
        printf("Число равно нулю\n");
    exit(a);
    return 0;
}
```

Рис.4 Командный файл

A screenshot of the Emacs editor interface. The title bar shows 'emacs@kyarteevalocaldomain'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The toolbar contains icons for file operations and editing. The text in the editor is a shell script:

```
#!/bin/bash
./null.cpp
echo $?
```

Рис.5 Запуск командного файла

A screenshot of a terminal window showing the execution of a script named 'null.sh'. The user enters the command './null.sh' three times, and the script outputs the result of the previous command's exit status in Russian. The terminal output is as follows:

```
[kyarteeva@kyarteevalocaldomain ~]$ ./null.sh
0
Число равно нулю
[kyarteeva@kyarteevalocaldomain ~]$ ./null.sh
45
Число больше нуля
[kyarteeva@kyarteevalocaldomain ~]$ ./null.sh
-6
Число меньше нуля
[kyarteeva@kyarteevalocaldomain ~]$
```

3. 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 умеет удалять все созданные им файлы (если они существуют).

Рис.6 Создание файлов

```

kyarteeva@kyarteevalocaldomain:~
Файл  Правка  Вид  Поиск  Терминал  Справка
#!/bin/bash
name=1
echo "Сколько файлов создать?"
read n
while (($name!=($n+1)))
do
    touch $name.tmp
    ((name+=1))
done
echo "Файлы созданы. Удалить их? ( 1 - да, 0 - нет)"
name=1
read d
if ((d==1))
then while (($name!=($n+1)))
do
    rm $name.tmp
    echo "Файл ${name}.tmp удалён"
    ((name+=1))
done
else if ((d==0)) then
    echo "Не удалено"
fi
fi
"file" "ch" 221  4266

```




Рис.7 Запуск командного файла

```

[kyarteeva@kyarteevalocaldomain ~]$ ./file.sh
Сколько файлов создать?
3
Файлы созданы. Удалить их? ( 1 - да, 0 - нет)
1
Файл 1.tmp удалён
Файл 2.tmp удалён
Файл 3.tmp удалён
[kyarteeva@kyarteevalocaldomain ~]$ ./file.sh
Сколько файлов создать?
2
Файлы созданы. Удалить их? ( 1 - да, 0 - нет)
0
Не удалено

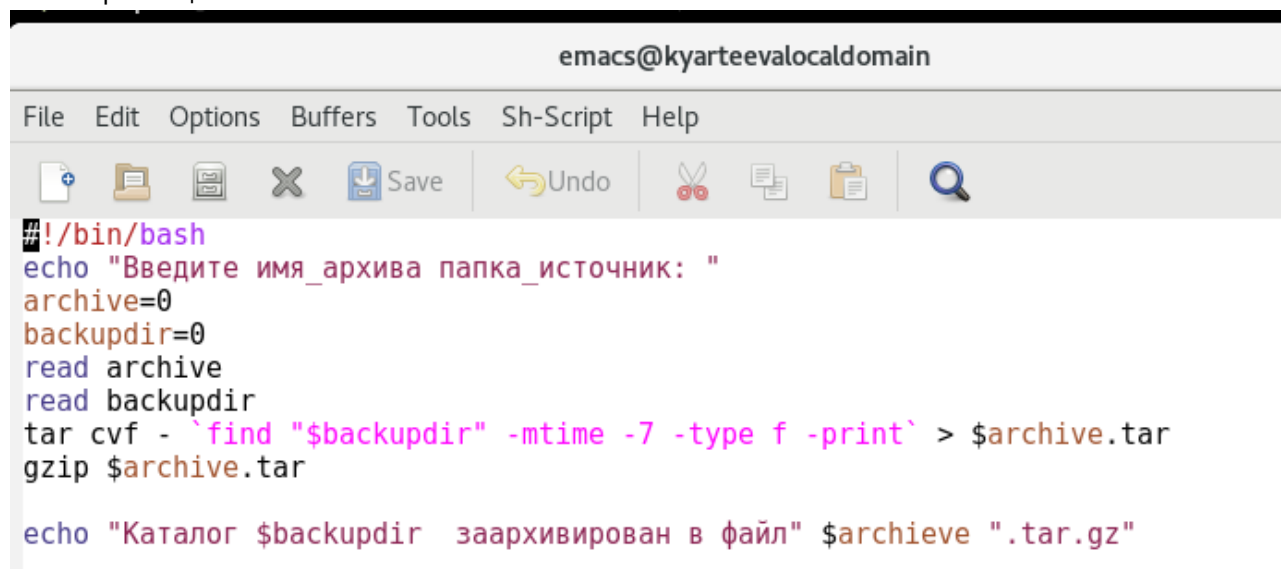
```

Рис.8 Результат

Имя	Размер	Дата изменения
 1.tmp	0 байт	17:31
 2.tmp	0 байт	17:31
 3.tmp	0 байт	17:31

4. Написала 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.

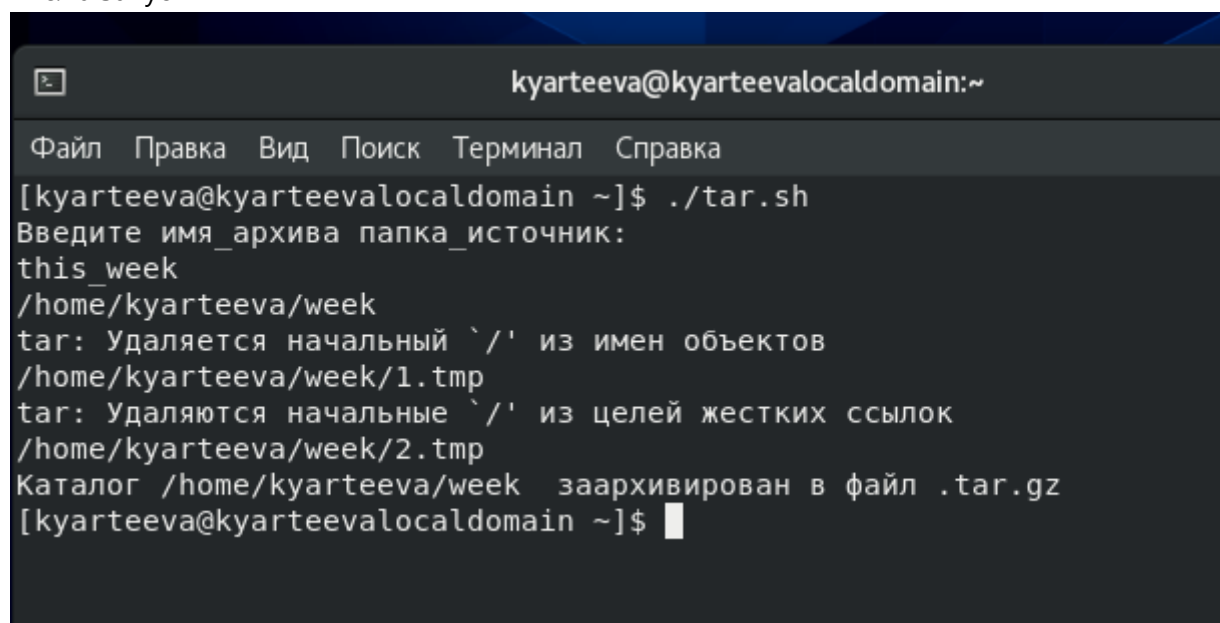
Рис.9 Архивация



```
emacs@kyarteevalocaldomain
File Edit Options Buffers Tools Sh-Script Help
[Icons: New, Open, Save, Undo, Redo, Find, etc.]
#!/bin/bash
echo "Введите имя_архива папка_источник: "
archive=0
backupdir=0
read archive
read backupdir
tar cvf - `find "$backupdir" -mtime -7 -type f -print` > $archive.tar
gzip $archive.tar

echo "Каталог $backupdir заархивирован в файл" $archive ".tar.gz"
```

Рис.10 Запуск



```
kyarteeva@kyarteevalocaldomain:~
Файл Правка Вид Поиск Терминал Справка
[kyarteeva@kyarteevalocaldomain ~]$ ./tar.sh
Введите имя_архива папка_источник:
this_week
/home/kyarteeva/week
tar: Удаляется начальный '/' из имен объектов
/home/kyarteeva/week/1.tmp
tar: Удаляются начальные '/' из целей жестких ссылок
/home/kyarteeva/week/2.tmp
Каталог /home/kyarteeva/week заархивирован в файл .tar.gz
[kyarteeva@kyarteevalocaldomain ~]$
```

## Контрольные вопросы

1. Каково предназначение команды getopts?

Она осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных.

2. Какое отношение метасимволы имеют к генерации имён файлов?

При генерации имен файлов используют метасимволы:

- произвольная (возможно пустая) последовательность символов;

? один произвольный символ;

[...] любой из символов, указанных в скобках перечислением и/или с указанием диапазона;

`cat f*` выдаст все файлы каталога, начинающиеся с "f";

`cat f` выдаст все файлы, содержащие "f";

`cat program.?` выдаст файлы данного каталога с однобуквенными расширениями, скажем "program.c" и "program.o", но не выдаст "program.com";

`cat [a-d]*` выдаст файлы, которые начинаются с "a", "b", "c", "d". Аналогичный эффект дадут и команды `"cat [abcd]"` и `"cat [bdac]"`.

### 3. Какие операторы управления действиями вы знаете?

for, case, if, while

### 4. Какие операторы используются для прерывания цикла?

Break, continue

### 5. Для чего нужны команды false и true?

Команды ОС UNIX возвращают код завершения, значение которого может быть использовано для принятия решения о дальнейших действиях. Команда test, например, создана специально для использования в командных файлах. Единственная функция этой команды заключается в выработке кода завершения.

### 6. Что означает строка `if test -f man$s/$i.$s`, встреченная в командном файле?

Означает условие существования файла `man$s/$i.$s`

### 7. Объясните различия между конструкциями while и until.

Если речь идет о 2-х параллельных действиях, то это while. когда мы показываем, что сначала делается 1-е действие. потом оно заканчивается при наступлении 2-го действия, применяем until.

## Вывод

В ходе лабораторной работы я получила изучила основы программирования в оболочке ОС UNIX/Linux, научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.