

# **РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных  
наук**

**Кафедра прикладной информатики и теории  
вероятностей**

## **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 12**

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

**дисциплина: Операционные системы**

**Студент: Тасыбаева Наталья Сергеевна**

**Группа: НПИбд-02-20**

**МОСКВА 2021г.**

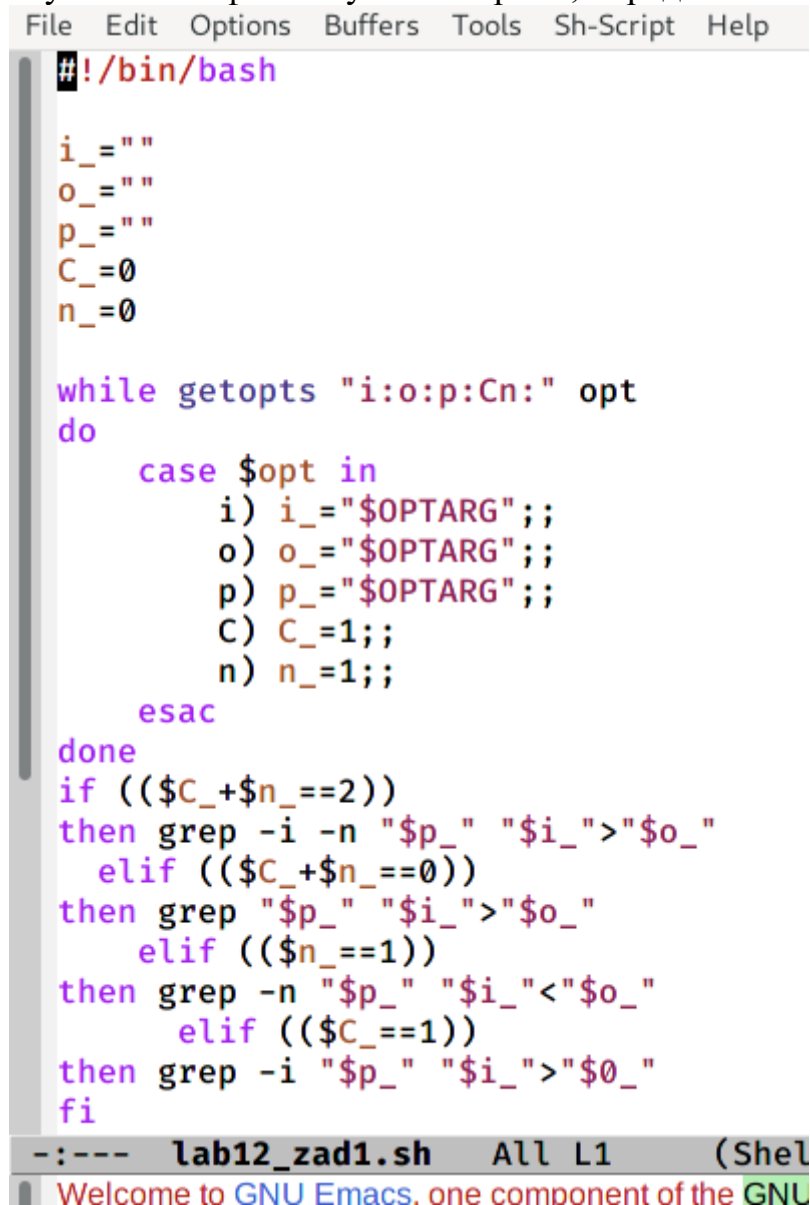
**Цель работы:** Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы.

### Ход работы:

1. Используя команды `getopts` `grep`, я написала командный файл (рис.1), который анализирует командную строку (рис.2) 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 `-p`



```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash

i_=""
o_=""
p_=""
C_=0
n_=0

while getopts "i:o:p:Cn:" opt
do
    case $opt in
        i) i_="$OPTARG";;
        o) o_="$OPTARG";;
        p) p_="$OPTARG";;
        C) C_=1;;
        n) n_=1;;
    esac
done
if (($C_+$n_==2))
then grep -i -n "$p_" "$i_">"$o_"
    elif (($C_+$n_==0))
then grep "$p_" "$i_">"$o_"
    elif (($n_==1))
then grep -n "$p_" "$i_"<"$o_"
    elif (($C_==1))
then grep -i "$p_" "$i_">"$o_"
fi

-:--- lab12_zad1.sh All L1 (Shell)
Welcome to GNU Emacs. one component of the GNU
```

Рис.1

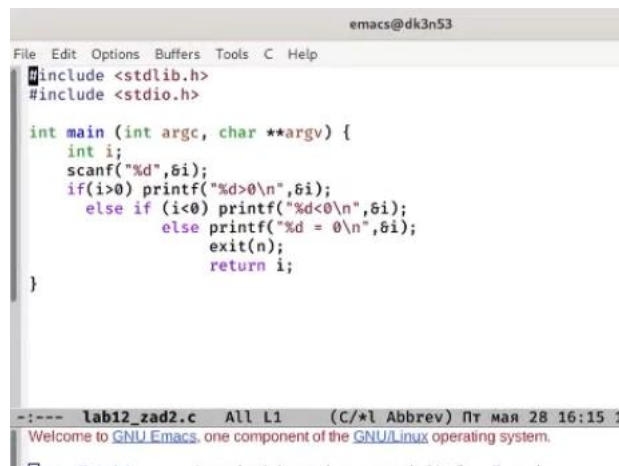
```

nstashbaeva@dk3n53 ~ $ emacs
nstashbaeva@dk3n53 ~ $ emacs lab12_zad1.sh
nstashbaeva@dk3n53 ~ $ chmod u+x lam12_zad1.sh
chmod: невозможно получить доступ к 'lam12_zad1.sh': Нет такого файла
или каталога
nstashbaeva@dk3n53 ~ $ chmod u+x lab12_zad1.sh
nstashbaeva@dk3n53 ~ $ touch lab12_1.txt
nstashbaeva@dk3n53 ~ $ emacs lab12_1.txt
nstashbaeva@dk3n53 ~ $ cat lab 12_1.txt
cat: lab: Нет такого файла или каталога
cat: 12_1.txt: Нет такого файла или каталога
nstashbaeva@dk3n53 ~ $ cat lab12_1.txt
capitan america is the best hero
my favorite serial is "game of Throne's"
i"m never gonna dance agian
nstashbaeva@dk3n53 ~ $ emacs
nstashbaeva@dk3n53 ~ $ cat lab12_1-1.txt
cat: lab12_1-1.txt: Нет такого файла или каталога
nstashbaeva@dk3n53 ~ $ cat lab12_1_1.txt
nstashbaeva@dk3n53 ~ $ ./lab12_zad1.sh -i lab12_1.txt -o lab12_1_1.tx
t -p dance
./lab12_zad1.sh: строка 28: синтаксическая ошибка рядом с неожиданным
маркером «fi»
./lab12_zad1.sh: строка 28: `fi'
nstashbaeva@dk3n53 ~ $ emacs lab12_zad1.sh
nstashbaeva@dk3n53 ~ $ ./lab12_zad1.sh -i lab12_1.txt -o lab12_1_1.tx
t -p dance
nstashbaeva@dk3n53 ~ $ cat lab12_1_1.txt
i"m never gonna dance agian

```

Рис.2

2. Написала на языке Си программу (рис.3), 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. Однако моя программа всё время выдаёт ошибку, исправить которую у меня не вышло.(рис.4)

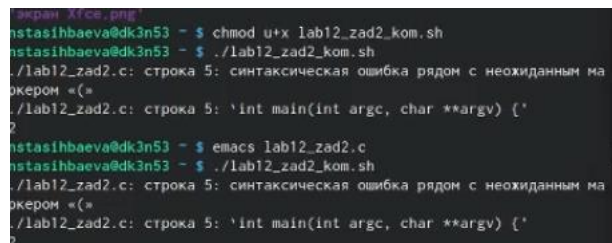


```
emacs@dk3n53
File Edit Options Buffers Tools C Help
#include <stdlib.h>
#include <stdio.h>

int main (int argc, char **argv) {
    int i;
    scanf("%d",&i);
    if(i>0) printf("%d>0\n",&i);
    else if (i<0) printf("%d<0\n",&i);
    else printf("%d = 0\n",&i);
    exit(n);
    return i;
}

-:--- lab12_zad2.c All L1 (C/*l Abbrev) Пт мая 28 16:15
Welcome to GNU Emacs, one component of the GNU/Linux operating system.
```

Рис.3



```
экран_47ce.png
stasihaeva@dk3n53 ~$ chmod u+x lab12_zad2_kom.sh
stasihaeva@dk3n53 ~$ ./lab12_zad2_kom.sh
./lab12_zad2.c: строка 5: синтаксическая ошибка рядом с неожиданным ма
кером «(»
./lab12_zad2.c: строка 5: 'int main(int argc, char **argv) {'
2
stasihaeva@dk3n53 ~$ emacs lab12_zad2.c
stasihaeva@dk3n53 ~$ ./lab12_zad2_kom.sh
./lab12_zad2.c: строка 5: синтаксическая ошибка рядом с неожиданным ма
кером «(»
./lab12_zad2.c: строка 5: 'int main(int argc, char **argv) {'
2
```

Рис.4

3. Написала командный файл (рис.5), 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)(рис.6)

```
#!/bin/bash
name=1
k=0
read k
while (($name!=($k+1)))
do
    for i in $name
    do
        touch $i.tmp
    done
    ((name+=1))
done
echo "Удалить файлы? (0/1)"
name=1
r=0
read r
if(r==1)
then
    while (($name!=($k+1)))
    do
        for i in $name
        do
            rm $i.tmp
        done
        ((name+=1))
    done
fi
```

Рис.5

```
2
Удалить файлы? (0/1)
0
```

Рис.6

4. Написала командный файл (рис.7), который с помощью команды tar запаковывает в архив все файлы в указанной директории.(рис.8)

```
#!/bin/bash
archive=0
BACKUPDIR=0
read archive
read BACKUPDIR
find "$BACKUPDIR" -type f print0 | xargs -0 tar rvf "$archive.tar" > /dev/null
gzip $archive
```

Рис.7

```
lab9sh.sh
/afs/.dk.sci.pfu.edu.ru/home/i/a/iakirichenko/
find: пути должны быть перед выражением: print0
Использование: find [-H] [-L] [-P] [-Oуровень] [-D help|tree|search|stat|rates|opt|exec] [путь...]
gzip: lab9sh.sh: No such file or directory
```

Рис.8

## Вывод:

Я изучила основы программирования в оболочке ОС UNIX, научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

### **Ответы на контрольные вопросы:**

1. Каково предназначение команды `getopts`?

Команда `getopts` используется для разбора позиционных параметров и проверки опций на допустимость.

2. Какое отношение метасимволы имеют к генерации имён файлов?

Метасимволы отвечают за параметры выдачи файлов.

3. Какие операторы управления действиями вы знаете?

`for`, `case`, `while`, `until`, `break`, `continue`, `if`

4. Какие операторы используются для прерывания цикла?

`break`

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

Аналог логических значений (0/1) для функций.

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

В конструкции `while` проверка условия выхода выполняется вначале, а не в конце цикла, если условие не удовлетворяется до начала выполнения цикла, то управление передается оператору стоящему сразу за телом цикла.

Условие выхода удовлетворяется, если выражение, определяющее условие выхода, ложно.

Между зарезервированными словами `until` может размещаться несколько операторов не применяя операторные скобки, когда как в конструкции `while` только один. Условие выхода удовлетворяется, если выражение, определяющее условие выхода, истинно