

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**  
**Факультет физико-математических и естественных наук**  
**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 12**  
**Программирование в командном процессоре ОС UNIX. Ветвления и циклы**  
дисциплина: Операционные системы

Студент: Тазаева Анастасия Анатольевна  
Группа: НПИбд-02-20

МОСКВА 2021г.

### Цель работы:

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

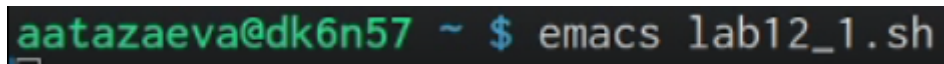
### Ход работы:

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

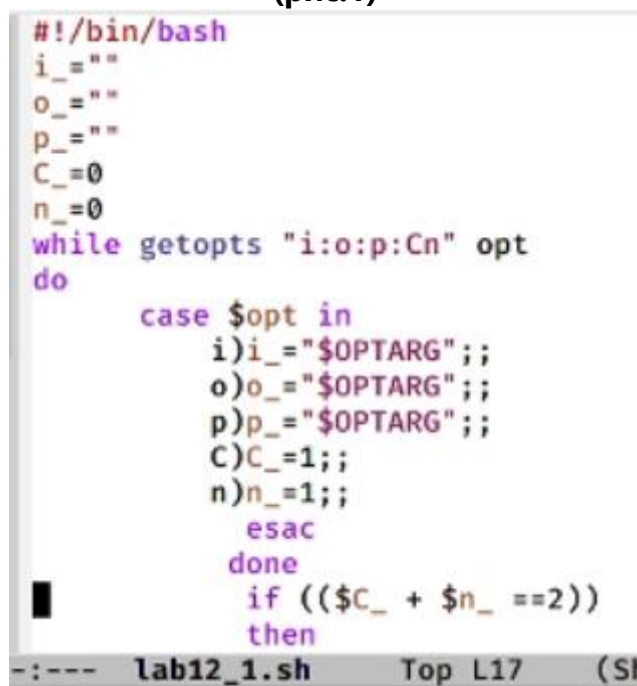
- `-iinputfile` — прочитать данные из указанного файла;
- `-ooutputfile` — вывести данные в указанный файл;
- `-rшаблон` — указать шаблон для поиска;
- `-C` — различать большие и малые буквы;
- `-n` — выдавать номера строк.

а затем ищет в указанном файле нужные строки, определяемые ключом `-r`

Для начала создала 3 файла, 1- файл, который будет читаться (i)(abc.sh рис.3), 2- файл, в который будут вводиться данные(efg.sh рис.4), 3- командный файл(рис.1), в нем и написан скрипт(рис.2). Далее выполнила файл(рис.5)

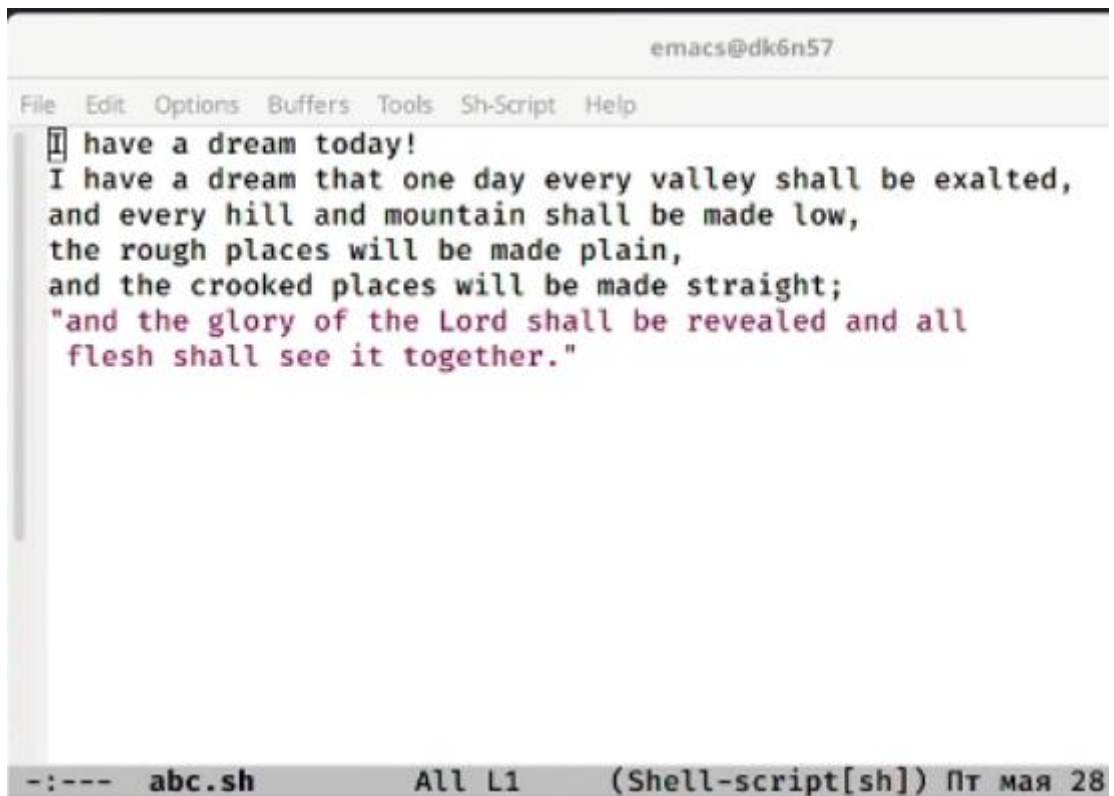


(рис.1)

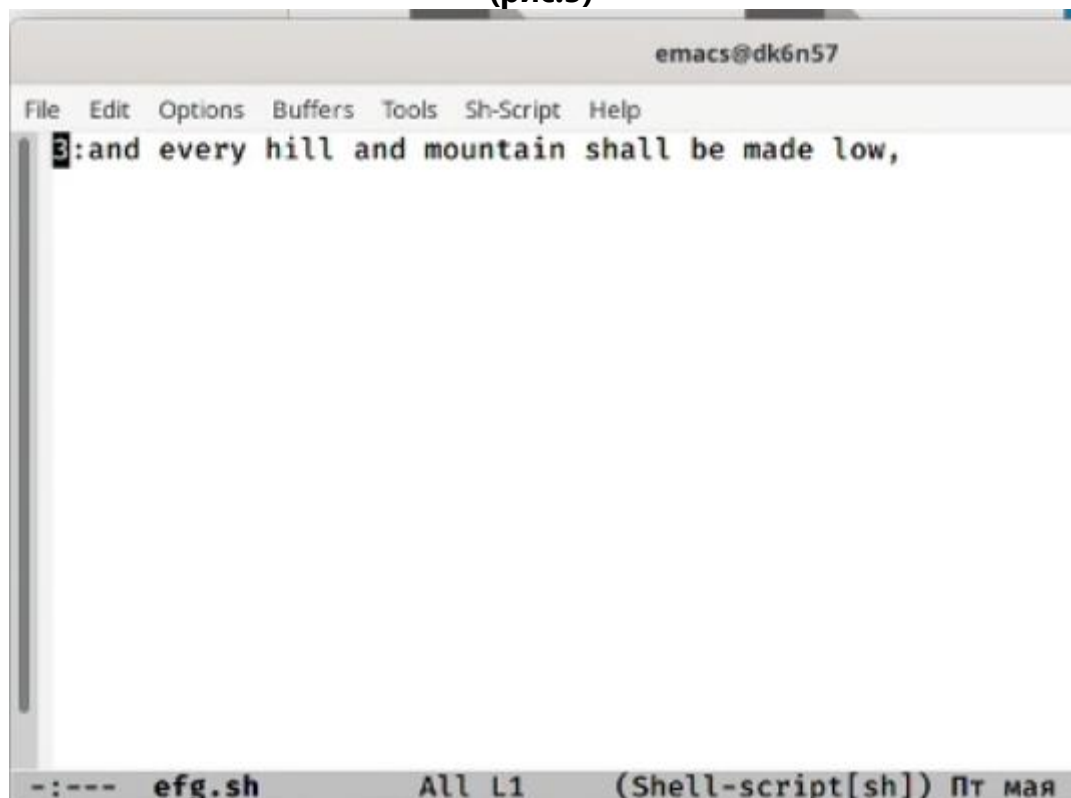


```
#!/bin/bash
i_=""
o_=""
p_=""
C_=0
n_=0
while getopts "i:o:p:Cn" opt
do
    case $opt in
        i)i_="$OPTARG";;
        o)o_="$OPTARG";;
        p)p_="$OPTARG";;
        C)C_=1;;
        n)n_=1;;
        esac
    done
    if (($C_ + $n_ == 2))
    then
```

(рис.2)



(рис.3)



(рис.4)

```
aatazaeva@dk6n57 ~ $ ./lab12_1.sh -i abc.sh -o efg.sh -p made -C -n  
aatazaeva@dk6n57 ~ $ cat efg.sh  
3:and every hill and mountain shall be made low,  
4:the rough places will be made plain,  
5:and the crooked places will be made straight;
```

(рис.5)

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 о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.

Создали файл с скриптом(рис.6), который определяет, какое число(больше-меньше-равно-0). Выводит числа 0 1 2, 0 - число отрицательное, 1 - число положительное, 2 - это число нуль. Также был создан командный файл, он как раз-таки и читает этот файл с программой, и получая число 0/1/2 он выводит для каждого сообщение.(рис.7). Проверили работу(рис.8)

```
emacs@dk6n57
File Edit Options Buffers Tools C Help
#include <stdio.h>
#include <stdlib.h>
int main(){
    printf("print number\n");
    int x;
    scanf ("%d",&x);
    if (x<0) exit (0);
    if (x>0) exit (1);
    if (x==0) exit (2);
    return 0;
}

-:--- lab12_2c.c All L1 (C/*l Abbrev) Пт мая 28 18:31 0.64
Welcome to GNU Emacs, one component of the GNU/Linux operating system.

Emacs Tutorial Learn basic keystroke commands (Учебник Emacs)
Emacs Guided Tour Overview of Emacs features at gnu.org
View Emacs Manual View the Emacs manual using Info
Absence of Warranty GNU Emacs comes with ABSOLUTELY NO WARRANTY
Copying Conditions Conditions for redistributing and changing Emacs
Ordering Manuals Purchasing printed copies of manuals
To quit a partially entered command, type Control-g.

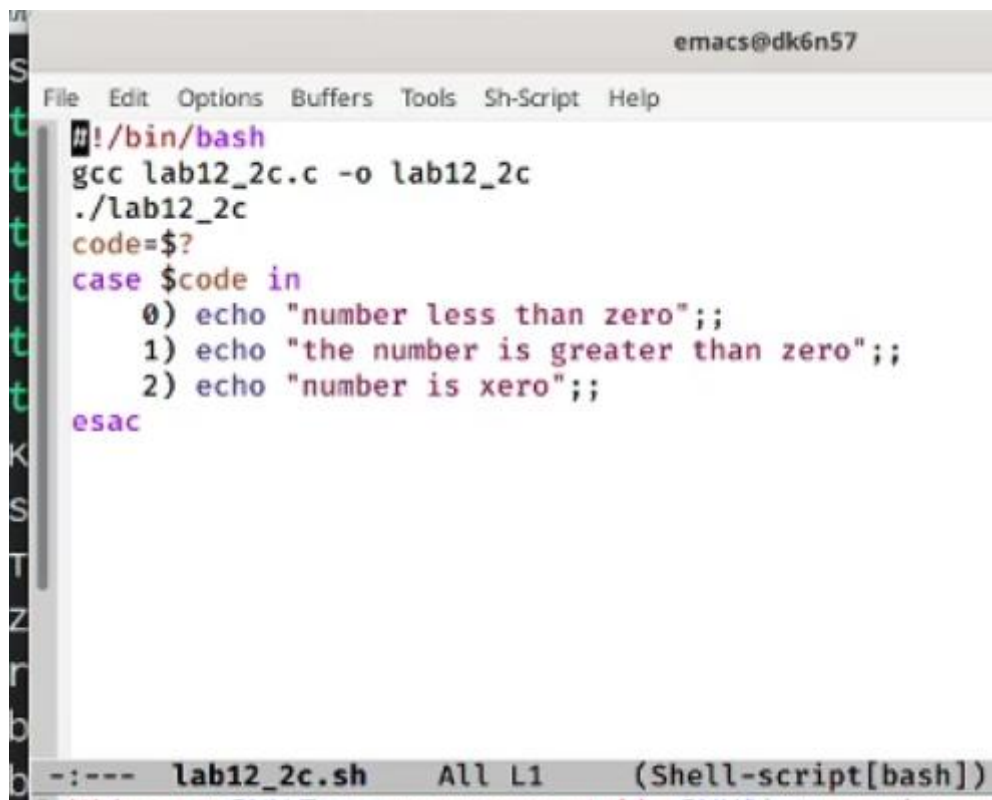
This is GNU Emacs 27.1 (build 1, x86_64-pc-linux-gnu, GTK+ Version 3.24.24, cairo version 1.16.
of 2021-03-11
Copyright (C) 2020 Free Software Foundation, Inc.

Dismiss this startup screen ☐ Never show it again.

U;%%- *GNU Emacs* All L3 (Fundamental) Пт мая 28 18:31 0.64
For information about GNU Emacs and the GNU system, type C-h C-a.

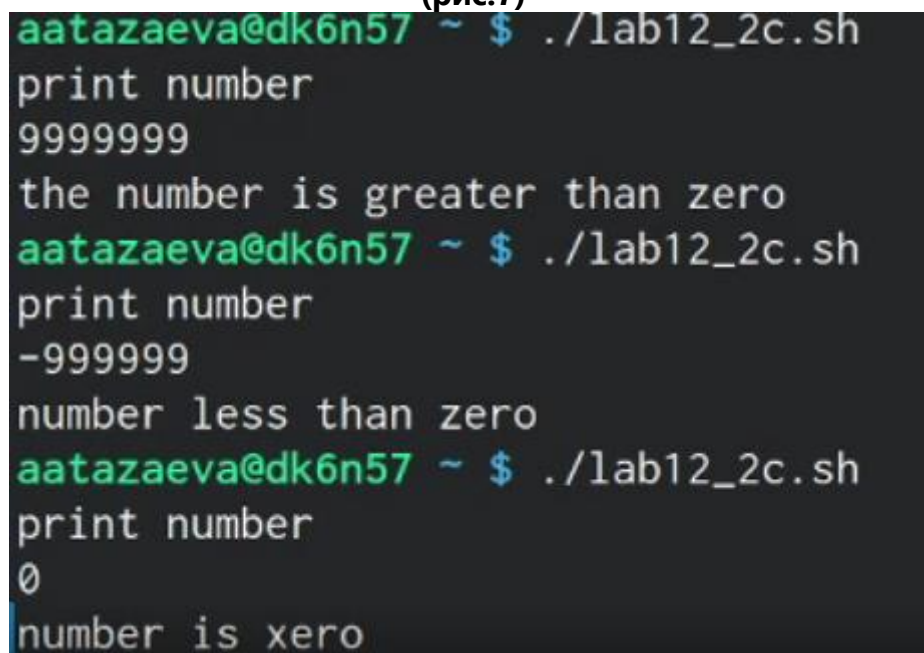
:and every hill and mountain shall be made low,
:the rough places will be made plain,
:and the crooked places will be made straight;
atazaeva@dk6n57 ~ $ emacs lab12_2c.c
```

(рис.6)



```
emacs@dk6n57
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
gcc lab12_2c.c -o lab12_2c
./lab12_2c
code=$?
case $code in
  0) echo "number less than zero";;
  1) echo "the number is greater than zero";;
  2) echo "number is xero";;
esac
-:--- lab12_2c.sh All L1 (Shell-script[bash])
```

(рис.7)



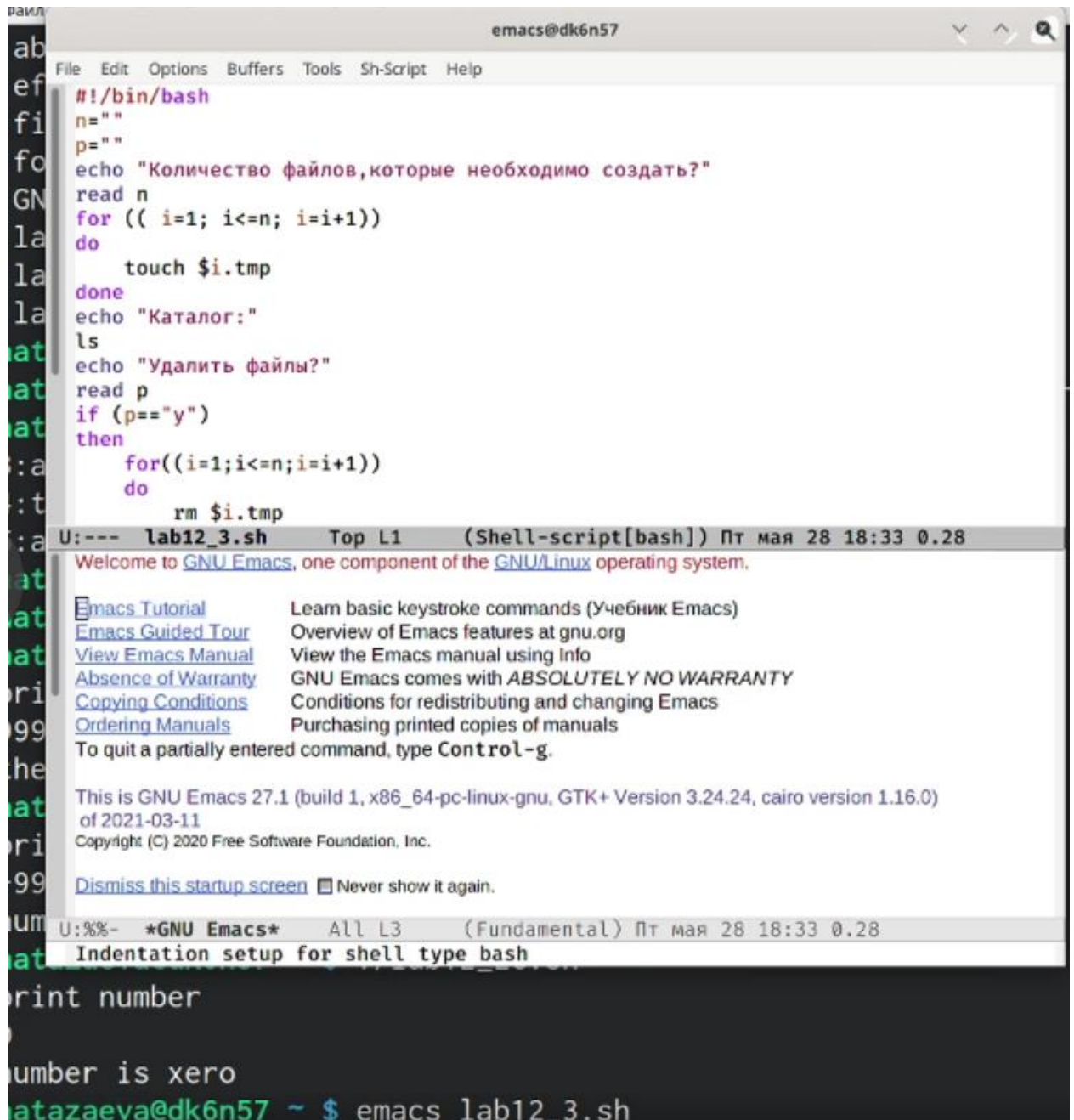
```
aatazaeva@dk6n57 ~ $ ./lab12_2c.sh
print number
9999999
the number is greater than zero
aatazaeva@dk6n57 ~ $ ./lab12_2c.sh
print number
-9999999
number less than zero
aatazaeva@dk6n57 ~ $ ./lab12_2c.sh
print number
0
number is xero
```

(рис.8)

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).



Создали файл со скриптом (рис.9 и 10). Файл выводит на экран какое кол-во необходимо создать файлов, с клавиатуры вводим число. Создаются файлы от 1 до n, сделали так, чтобы программа выводила содержимое директории домашней. Далее по необходимости удаляются файлы, и снова выводится содержимое дом.дир.(рис.11)



```
#!/bin/bash
n=""
p=""
echo "Количество файлов, которые необходимо создать?"
read n
for (( i=1; i<=n; i=i+1))
do
    touch $i.tmp
done
echo "Каталог:"
ls
echo "Удалить файлы?"
read p
if (p=="y")
then
    for((i=1;i<=n;i=i+1))
    do
        rm $i.tmp
    done
fi
```

U:--- lab12\_3.sh Top L1 (Shell-script[bash]) Пт мая 28 18:33 0.28

Welcome to [GNU Emacs](#), one component of the [GNU/Linux](#) operating system.

<a href="#">Emacs Tutorial</a>	Learn basic keystroke commands (Учебник Emacs)
<a href="#">Emacs Guided Tour</a>	Overview of Emacs features at gnu.org
<a href="#">View Emacs Manual</a>	View the Emacs manual using Info
<a href="#">Absence of Warranty</a>	GNU Emacs comes with <b>ABSOLUTELY NO WARRANTY</b>
<a href="#">Copying Conditions</a>	Conditions for redistributing and changing Emacs
<a href="#">Ordering Manuals</a>	Purchasing printed copies of manuals

To quit a partially entered command, type **Control-g**.

This is GNU Emacs 27.1 (build 1, x86\_64-pc-linux-gnu, GTK+ Version 3.24.24, cairo version 1.16.0) of 2021-03-11  
Copyright (C) 2020 Free Software Foundation, Inc.

[Dismiss this startup screen](#) ☐ Never show it again.

U:%%- \*GNU Emacs\* All L3 (Fundamental) Пт мая 28 18:33 0.28

Indentation setup for shell type `bash`

print number

umber is xero

atazaeva@dk6n57 ~ \$ emacs lab12\_3.sh

(рис.9)

```
File Edit Options Buffers Tools Sh-Script Help
do
    rm $i.tmp
done
echo "Каталог:"
ls
fi
```

(рис.10)

```
aatazaeva@dk6n57 ~ $ ./lab12_3.sh
Количество файлов, которые необходимо создать?
10
Каталог:
10.tmp    abc.sh~      lab12-2c.cpp~  public
1.tmp     efg.sh       lab12_2c.cpp   public_html
2.tmp     first_task.sh~ lab12_2c.sh    tmp
3.tmp     fourth_task.sh~ lab12_2c.sh~   Видео
4.tmp     GNUstep      lab12_3.sh     Документы
5.tmp     lab12_1.sh   lab12_3.sh~    Загрузки
6.tmp     lab12_1.sh~  lab12_4.sh     Изображения
7.tmp     lab12_2c     lab12_4.sh~    Музыка
8.tmp     lab12_2c.c   my_os          Общедоступные
9.tmp     lab12_2c.c~ nastya.tar      'Рабочий стол'
abc.sh    lab12-2c.cpp OS              Шаблоны
Удалить файлы?
y
Каталог:
abc.sh      lab12_2c.c    lab12_4.sh     Документы
abc.sh~     lab12_2c.c~   lab12_4.sh~    Загрузки
efg.sh      lab12-2c.cpp  my_os          Изображения
first_task.sh~ lab12-2c.cpp~ nastya.tar      Музыка
fourth_task.sh~ lab12_2c.cpp  OS             Общедоступные
GNUstep     lab12_2c.sh   public         'Рабочий стол'
lab12_1.sh  lab12_2c.sh~ public_html     Шаблоны
lab12_1.sh~ lab12_3.sh    tmp            Видео
lab12_2c    lab12_3.sh~   Видео
```

(рис.11)

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*).

Написала командный файл, который с помощью команды *tar* запаковывает в архив все файлы в указанной директории. Запаковываются только те файлы, которые были изменены менее недели тому назад (использовал команду *find*). (рис.12) и проверила(рис.13)



The screenshot shows the Emacs editor interface. The top menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The main window displays a shell script named `lab12_4.sh` with the following content:

```
#!/bin/bash
archive=""
echo "Какой создать архив?"
read archive
directory=""
echo "А теперь директорию?"
read directory
cd $dir
find . -mtime -7 -type f -print0 | xargs -0 tar -czf ${archive}.tar
ls
```

Below the script, the Emacs startup screen is visible, displaying the following text:

U:--- lab12\_4.sh All L1 (Shell-script[bash]) Пт мая 28 18:35

Welcome to [GNU Emacs](#), one component of the [GNU/Linux](#) operating system.

[Emacs Tutorial](#) Learn basic keystroke commands (Учебник Emacs)  
[Emacs Guided Tour](#) Overview of Emacs features at gnu.org  
[View Emacs Manual](#) View the Emacs manual using Info  
[Absence of Warranty](#) GNU Emacs comes with **ABSOLUTELY NO WARRANTY**  
[Copying Conditions](#) Conditions for redistributing and changing Emacs  
[Ordering Manuals](#) Purchasing printed copies of manuals

To quit a partially entered command, type **Control-g**.

This is GNU Emacs 27.1 (build 1, x86\_64-pc-linux-gnu, GTK+ Version 3.24.24, cairo version 1.16. of 2021-03-11  
Copyright (C) 2020 Free Software Foundation, Inc.

[Dismiss this startup screen](#) ☐ Never show it again.

U:%%- \*GNU Emacs\* All L3 (Fundamental) Пт мая 28 18:35 0.59

Indentation setup for shell\_type bash

lab12\_1.sh lab12\_2c.sh~ public\_html Шабло  
lab12\_1.sh~ lab12\_3.sh tmp  
lab12\_2c lab12\_3.sh~ Видео

aatazaeva@dk6n57 ~ \$ emacs lab12\_4.sh

(рис.12)

```
atazaeva@dk6n57 ~ $ ./lab12_4.sh
Какой создать архив?
globulus
А тепеь директорию?
earth
abc.sh          lab12_2c        lab12_3.sh~     Видео
abc.sh~         lab12_2c.c      lab12_4.sh      Документы
efg.sh          lab12_2c.c~     lab12_4.sh~     Загрузки
first_task.sh~  lab12-2c.cpp    my_os           Изображения
fourth_task.sh~ lab12-2c.cpp~   nastya.tar      Музыка
globulus.tar    lab12_2c.cpp    OS              Общедоступные
GNUstep         lab12_2c.sh     public          'Рабочий стол'
lab12_1.sh      lab12_2c.sh~    public_html     Шаблоны
lab12_1.sh~     lab12_3.sh      tmp
```

(рис.13)

### Контрольные вопросы:

1. Каково предназначение команды `getopts`?  
Команда `getopts` используется для разбора параметров и проверки опций на допустимость.  
Осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных.
2. Какое отношение метасимволы имеют к генерации имён файлов?  
Метасимволы отвечают за параметры выдачи файлов, а следовательно могут вносить иной смысл, нежели прямое значение как символа.  
Например:
  - - — соответствует произвольной, в том числе и пустой строке;
  - ? — соответствует любому одному символу;
  - [c1-c1] — соответствует любому символу, лексикографически находящемуся между символами c1 и c2.
  - `echo *` — выведет имена всех файлов текущего каталога, что представляет собой простейший аналог команды `ls`;
  - `ls *.c` — выведет все файлы с последними двумя символами, равными `.c`.
  - `echo prog.?` — выдаст все файлы, состоящие из пяти или шести символов, первыми пятью символами которых являются `prog.` .
  - [a-z]\* — соответствует произвольному имени файла в текущем каталоге, начинающемуся с любой строчной буквы латинского алфавита.

3. Какие операторы управления действиями вы знаете?  
for, break , while, until, case ,continue, if, else
4. Какие операторы используются для прерывания цикла?  
break
5. Для чего нужны команды false и true?  
Это логические значения (0 и 1 соответственно). Можно использовать как более понятный вариант
6. Что означает строка `if test -f man$s/$i.$s`, встречаемая в командном файле?  
Условие существования файла `man$s/$i.$s`
7. Объясните различия между конструкциями `while` и `until`  
`while` - проверка условия затем выполнение тела цикла  
`until` - выполнение тела цикла затем проверка условия

### **ВЫВОД:**

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