

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №11 =====

дисциплина: Операционные системы

Студент: Койфман Кирилл Дмитриевич

Группа: НПИбд-01-21

Введение.

Цель работы.

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

Задачи.

1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами:
 - `-inputfile` — прочитать данные из указанного файла;
 - `-outputfile` — вывести данные в указанный файл;
 - `-ршаблон` — указать шаблон для поиска;
 - `-С` — 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).

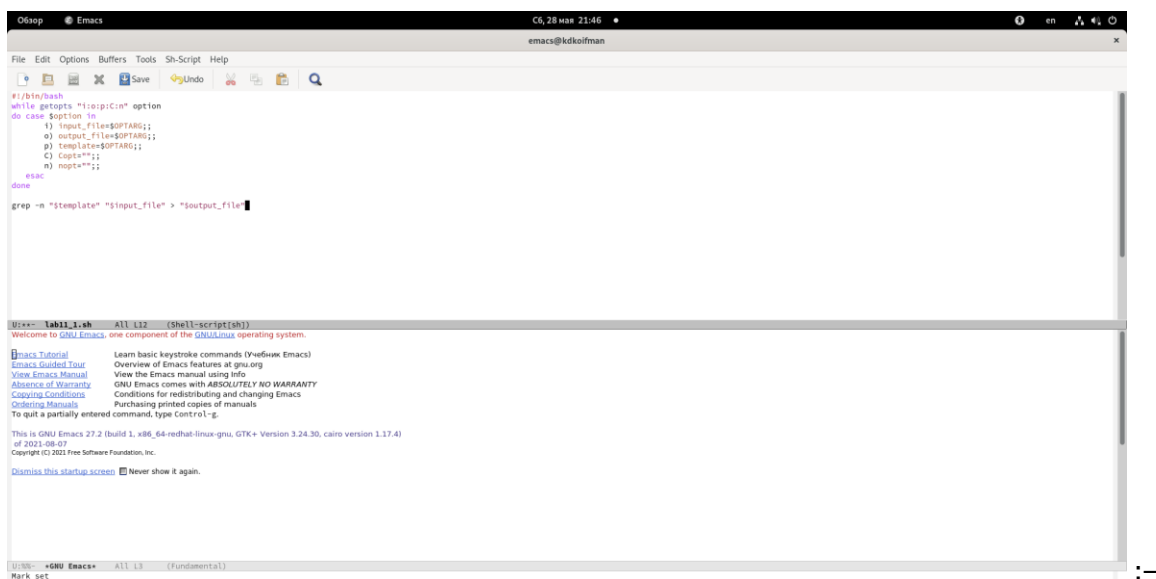
Ход работы.

1 задание.

Используя команды getopts grep, напомним командный файл, который анализирует командную строку с ключами:

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

а затем ищет в указанном файле нужные строки, определяемые ключом -р (рис.1, 2, 3)



```
#!/bin/bash
while getopts ":i:o:p:C:n" option
do case $option in
  i) input_file=$OPTARG;;
  o) output_file=$OPTARG;;
  p) template=$OPTARG;;
  C) case="";;
  n) nopt="";;
  *) ;;
esac
done

grep -n "$template" "$input_file" > "$output_file"
```

U+*** lab1_1.sh All L12 (shell-script(ish))

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

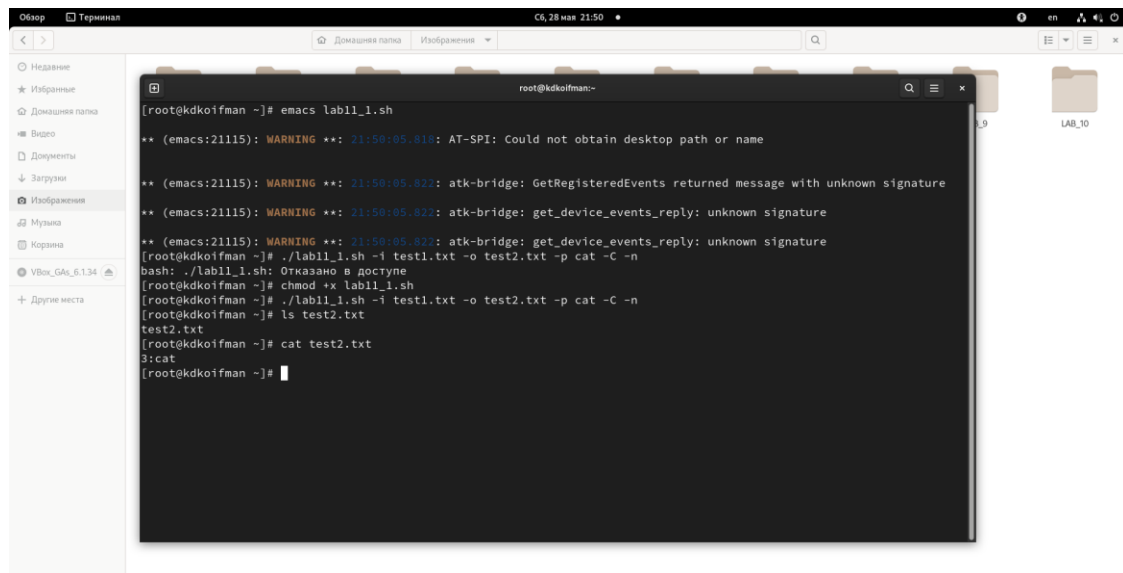
[Emacs Tutorial](#) Learn basic keystroke commands (YieBieW Emacs)
[Emacs Guided Tour](#) Overview of Emacs features at gnu.org
[View Emacs Manual](#) View the Emacs manual using Info
[Absence of Warranty](#) GNU Emacs comes with ABSOLUTELY NO WARRANTY
[Copying Conditions](#) Conditions for redistributing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

This is GNU Emacs 27.2 (build 1, x86_64-redhat-linux-gnu, GTK+ Version 3.24.30, cairo version 1.17.4) of 2021-08-07.
Copyright (C) 2021 Free Software Foundation, Inc.

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

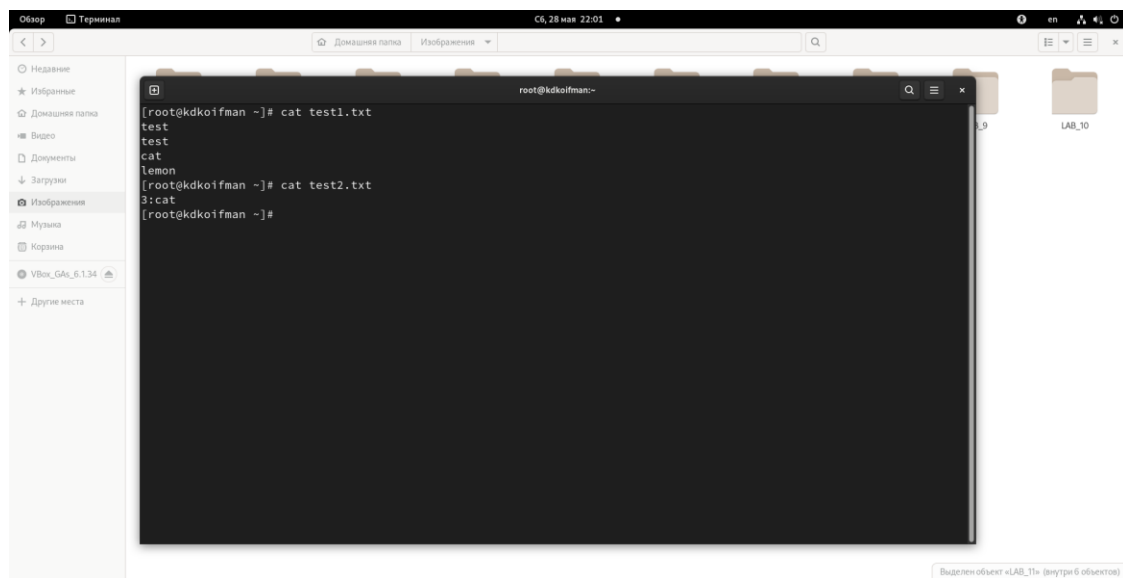
U+*** GNU Emacs All L12 (Fundamental)
Mark set

рис.1(код командного файла)



```
[root@kdkoifman ~]# emacs lab11_1.sh
** (emacs:21115): WARNING **: 21:50:05.818: AT-SPI: Could not obtain desktop path or name
** (emacs:21115): WARNING **: 21:50:05.822: atk-bridge: GetRegisteredEvents returned message with unknown signature
** (emacs:21115): WARNING **: 21:50:05.822: atk-bridge: get_device_events_reply: unknown signature
** (emacs:21115): WARNING **: 21:50:05.822: atk-bridge: get_device_events_reply: unknown signature
[root@kdkoifman ~]# ./lab11_1.sh -i test1.txt -o test2.txt -p cat -C -n
bash: ./lab11_1.sh: Отказано в доступе
[root@kdkoifman ~]# chmod +x lab11_1.sh
[root@kdkoifman ~]# ./lab11_1.sh -i test1.txt -o test2.txt -p cat -C -n
[root@kdkoifman ~]# ls test2.txt
test2.txt
[root@kdkoifman ~]# cat test2.txt
3:cat
[root@kdkoifman ~]#
```

рис.2(выполнение командного файла)



```
[root@kdkoifman ~]# cat test1.txt
test
test
cat
lemon
[root@kdkoifman ~]# cat test2.txt
3:cat
[root@kdkoifman ~]#
```

рис.3(содержимое файлов test1.txt и test2.txt)

2 задание.

Напишем 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 код завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено (рис.4, 5, 6, 7)

```
GNU Emacs C6, 28 May 21:58 emacs@kdkolifman
File Edit Options Buffers Tools C Help
#include<stdio.h>

int main()
{
    int input_value=0;
    scanf("%d", &input_value);
    if (input_value==0) return 1;
    if (input_value>0) return 2;
    if (input_value<0) return 3;
    else return 0;
}

U:---- Lab1_22.c All L11 (C++1 Abbrev)
Welcome to GNU Emacs, one component of the GNU/Linux operating system.
[ Emacs Tutorial Learn basic keystroke commands (Y&eS&u&w 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 27.2 (build 1, x86_64-redhat-linux-gnu, GTK+ Version 3.24.30, cairo version 1.17.4)
of 2021-08-07.
Copyright (C) 2021 Free Software Foundation, Inc.
Dismiss this startup screen [N] Never show it again.

U:W:-- GNU Emacs+ All L3 (Fundamental)
Wrote /root/Lab1_22.c
```

рис.4(код из C-файла)

```
GNU Emacs C6, 28 May 21:57 emacs@kdkolifman
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash

echo "Input value: "
./lab1_2
case $? in
  1) echo "The value is equal 0";
  2) echo "The value is more than 0";
  3) echo "The value is less than 0";
  0) echo "Try again";
esac

U:---- Lab1_2.sh All L4 (Shell-script(bash))
Welcome to GNU Emacs, one component of the GNU/Linux operating system.
[ Emacs Tutorial Learn basic keystroke commands (Y&eS&u&w 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 27.2 (build 1, x86_64-redhat-linux-gnu, GTK+ Version 3.24.30, cairo version 1.17.4)
of 2021-08-07.
Copyright (C) 2021 Free Software Foundation, Inc.
Dismiss this startup screen [N] Never show it again.

U:W:-- GNU Emacs+ All L3 (Fundamental)
Wrote /root/Lab1_2.sh
```

рис.5(код командного файла)

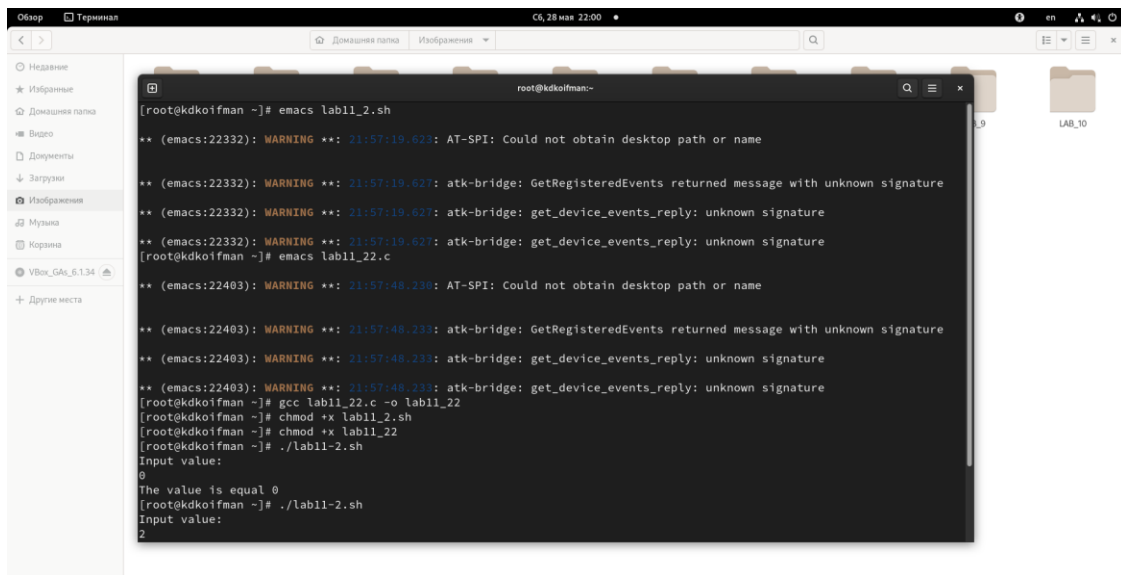
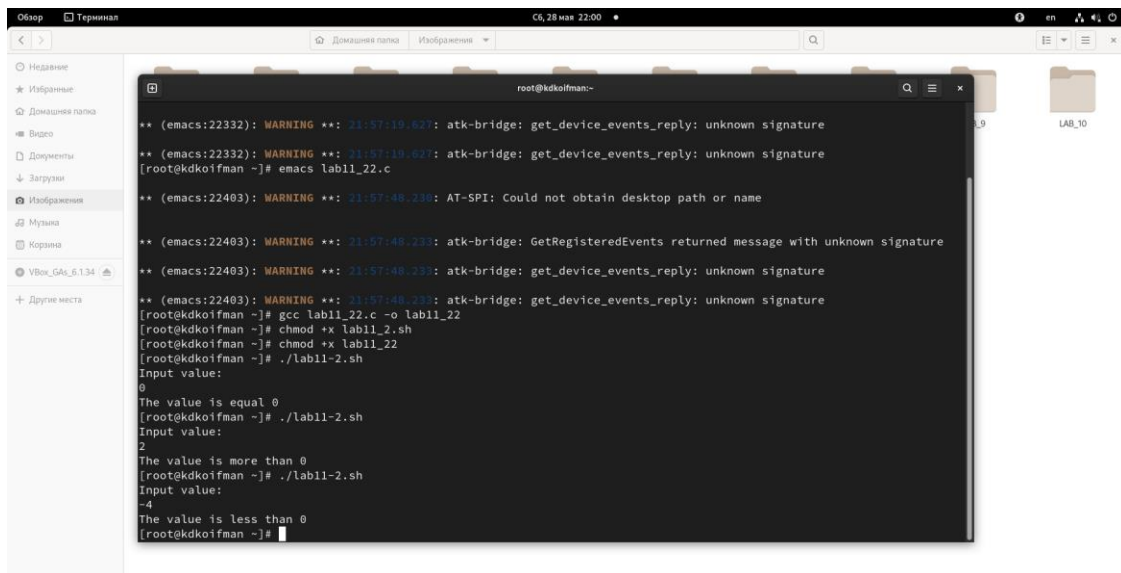


рис.6(выполнение командного файла)



:-: рис.7

3 задание.

Напишем 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)(рис.8,9)

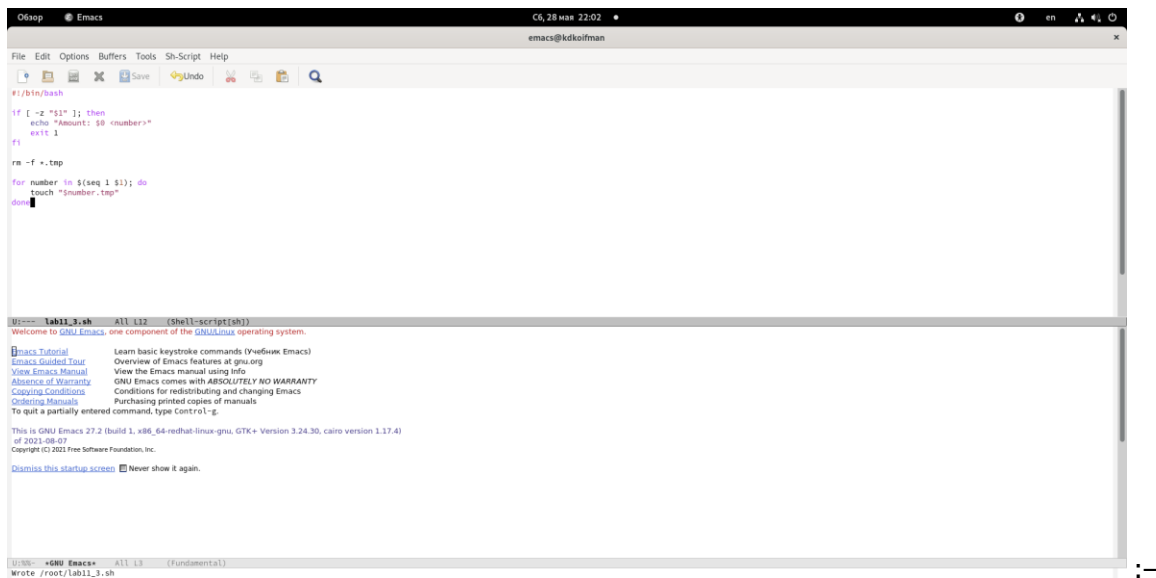


рис.8(код командного файла)

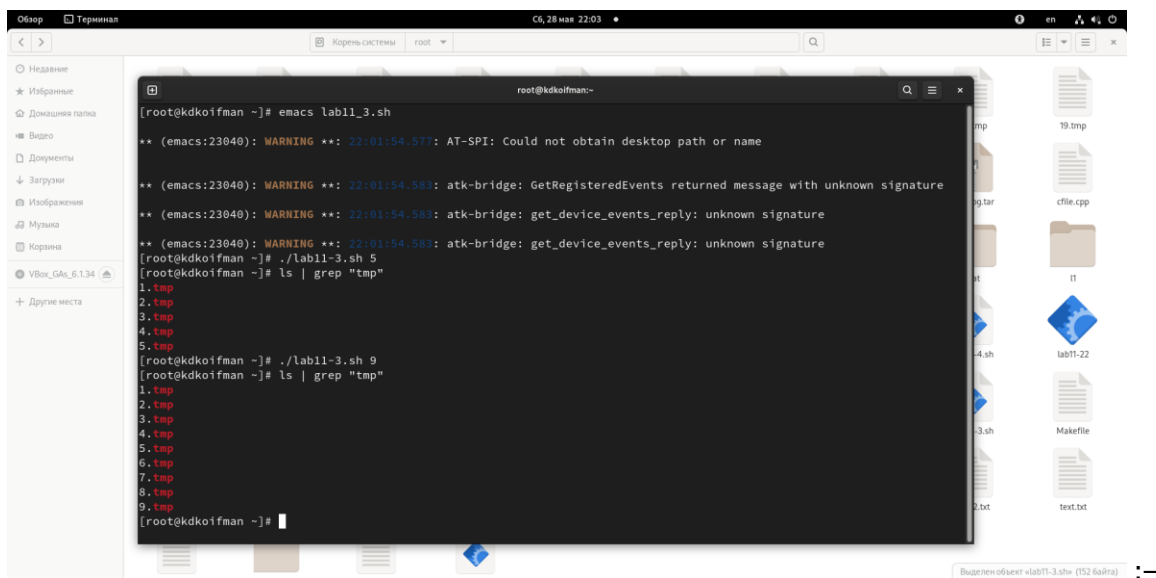


рис.9(выполнение командного файла)

4 задание.

Напишем командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицируем его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find)(рис.10,11,12,13)

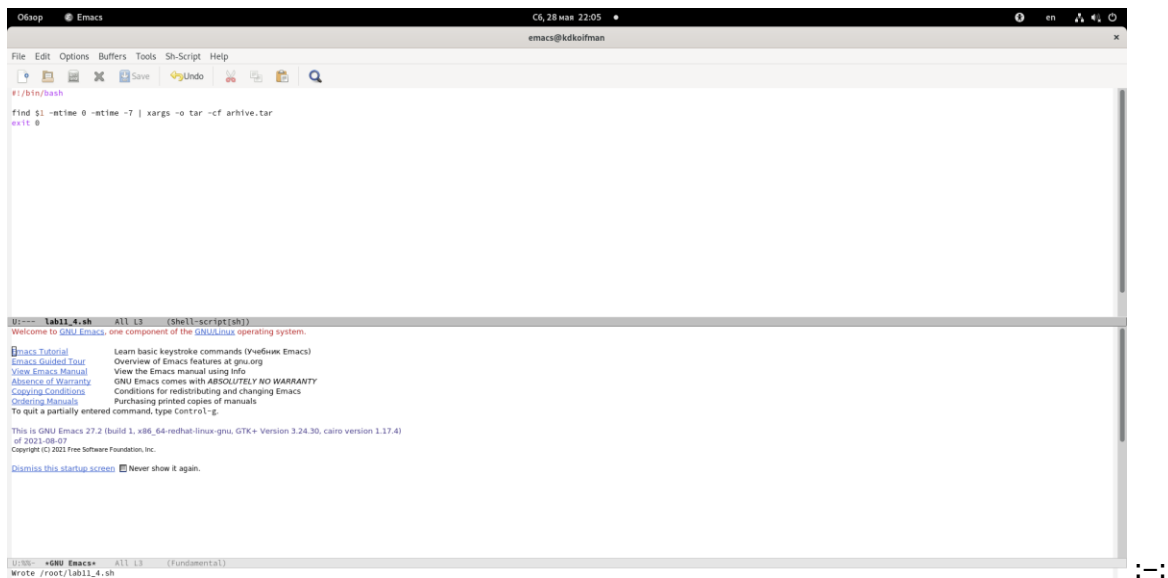


рис.10(код командного файла)

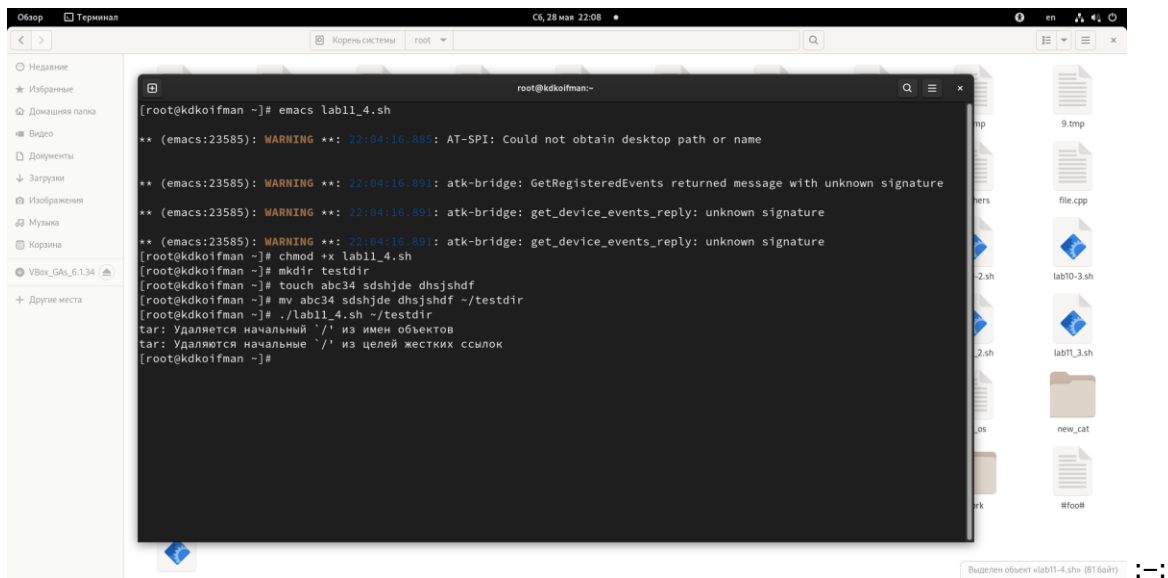


рис.11(выполнение командного файла)

2. Какое отношение метасимволы имеют к генерации имён файлов?
Метасимволы обрабатываются оболочкой в качестве списка строк, имена которых соответствуют именам файлов, а им в свою очередь соответствует определённая последовательность метасимволов.
3. Какие операторы управления действиями вы знаете? Операторы управления: for, while, if, else, case, switch, then, fi, done, esac.
4. Какие операторы используются для прерывания цикла? Операторы break, return и continue могут прерывать выполнение циклов.
5. Для чего нужны команды false и true? Для работы с условными операторами, циклами и логическими переменными.
6. Что означает строка if test -f mans/i.\$s, встреченная в командном файле?
7. Объясните различия между конструкциями while и until. Оба оператора создают цикл. Однако цикл while будет выполняться до тех пор, пока выражение истинно, а until выполняется пока выражение ложно.