

Лабораторная работы №11

“Программирование в командном процессоре ОС UNIX. Ветвления и циклы”

Тулеев Мад

RUDN University, Moscow, Russian Federation

Цель работы:

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

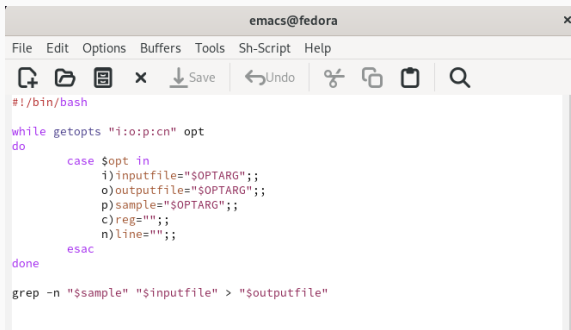
1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами:
 - `-iinputfile` — прочитать данные из указанного файла;
 - `-ooutputfile` — вывести данные в указанный файл;
 - `-р`шаблон — указать шаблон для поиска;
 - `-C` — различать большие и малые 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 ∞ (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`).

Написал скрипт, который читает данные из указанного файла, записывает их в другой, учитывая введенные опции. (рис. 1)

The image shows a screenshot of an Emacs editor window titled 'emacs@fedora'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. Below the menu is a toolbar with icons for file operations (open, save, close, delete), editing (undo, redo, cut, copy, paste), and search. The main text area contains a shell script with the following content:

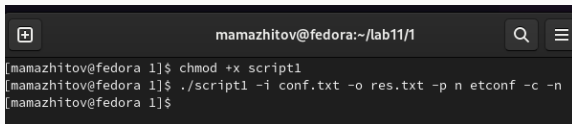
```
#!/bin/bash

while getopts "i:o:p:cn" opt
do
    case $opt in
        i)inputfile="$OPTARG";;
        o)outputfile="$OPTARG";;
        p)sample="$OPTARG";;
        c)reg="";;
        n)line="";;
        esac
    done

    grep -n "$sample" "$inputfile" > "$outputfile"
```

Figure 1: Код 1 скрипта

Запустил скрипт. (рис. 2)

A terminal window with a dark background. The title bar shows a plus icon, the username and host 'mamazhitov@fedora:~/lab11/1', a search icon, and a menu icon. The terminal content shows three lines of commands and their prompts: '[mamazhitov@fedora 1]\$ chmod +x script1', '[mamazhitov@fedora 1]\$./script1 -i conf.txt -o res.txt -p n etconf -c -n', and '[mamazhitov@fedora 1]\$'.

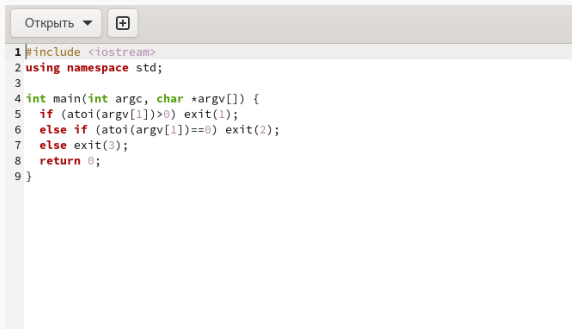
```
mamazhitov@fedora:~/lab11/1
[mamazhitov@fedora 1]$ chmod +x script1
[mamazhitov@fedora 1]$ ./script1 -i conf.txt -o res.txt -p n etconf -c -n
[mamazhitov@fedora 1]$
```


Figure 2: Работа скрипта

Проверка. (рис. ??)

Проверка работы скрипта

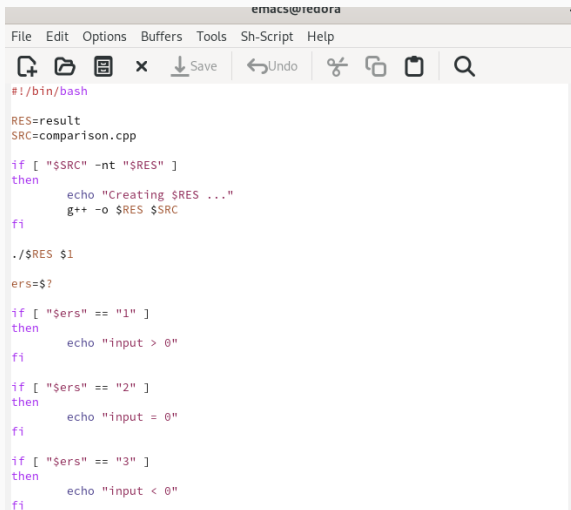
Написал командный файл и программу на языке C++, которые получает на входе число и выводит больше, меньше или равно “0”.(рис. 3, 4)



```
Открыть ▼ 
```

```
1 #include <iostream>
2 using namespace std;
3
4 int main(int argc, char *argv[]) {
5     if (atoi(argv[1])>0) exit(1);
6     else if (atoi(argv[1])==0) exit(2);
7     else exit(3);
8     return 0;
9 }
```

Figure 3: Код 2 скрипта



The image shows a screenshot of an Emacs editor window titled "emacs@redora". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", and "Help". The toolbar contains icons for opening a file, saving, undo, redo, and search. The main text area contains a shell script with the following content:

```
#!/bin/bash

RES=result
SRC=comparison.cpp

if [ "$SRC" -nt "$RES" ]
then
    echo "Creating $RES ..."
    g++ -o $RES $SRC
fi

./$RES $1

ers=$?

if [ "$ers" == "1" ]
then
    echo "input > 0"
fi

if [ "$ers" == "2" ]
then
    echo "input = 0"
fi

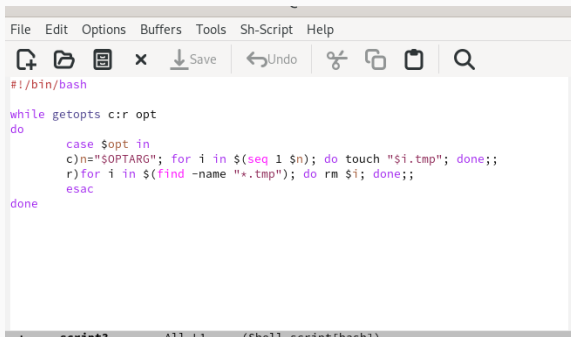
if [ "$ers" == "3" ]
then
    echo "input < 0"
fi
```

Figure 4: Код C++ файла

Запустил скрипт.(рис. 5)

```
[mamazhitov@fedora 2]$ chmod +x script2
[mamazhitov@fedora 2]$ ./script2 3
Creating result ...
input > 0
[mamazhitov@fedora 2]$ ./script2 0
input = 0
[mamazhitov@fedora 2]$ ./script2 -1
input < 0
[mamazhitov@fedora 2]$
```

Открыл в emacs файл *script3* и написал программу, которая в зависимости от введенных опций либо создает определенное кол-во файлов, либо удаляет их.(рис. 6)

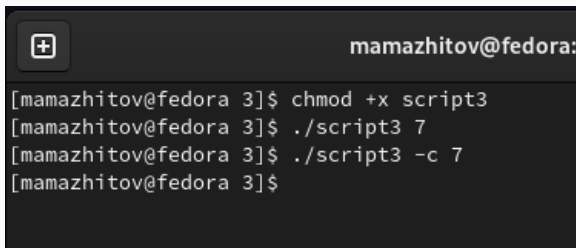


```
#!/bin/bash

while getopts c:r opt
do
    case $opt in
        c)n="$OPTARG"; for i in $(seq 1 $n); do touch "$i.tmp"; done;;
        r)for i in $(find -name "*.tmp"); do rm $i; done;;
        esac
    done
```

Figure 6: Код 3 скрипта

Запустил скрипт для создания файлов.(рис. 7)

A terminal window with a dark background. The title bar shows a window icon with a plus sign and the text 'mamazhitov@fedora:'. The terminal content shows four lines of commands and their prompts: '[mamazhitov@fedora 3]\$ chmod +x script3', '[mamazhitov@fedora 3]\$./script3 7', '[mamazhitov@fedora 3]\$./script3 -c 7', and '[mamazhitov@fedora 3]\$'.

```
mamazhitov@fedora:
[mamazhitov@fedora 3]$ chmod +x script3
[mamazhitov@fedora 3]$ ./script3 7
[mamazhitov@fedora 3]$ ./script3 -c 7
[mamazhitov@fedora 3]$
```

Figure 7: Создание файлов

Результат.(рис. 8)

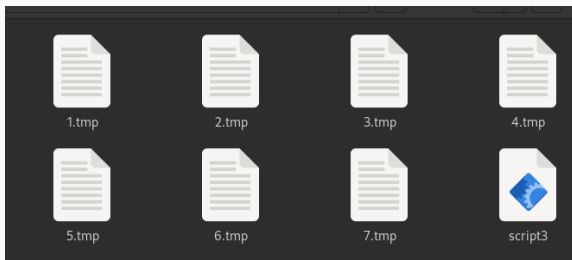
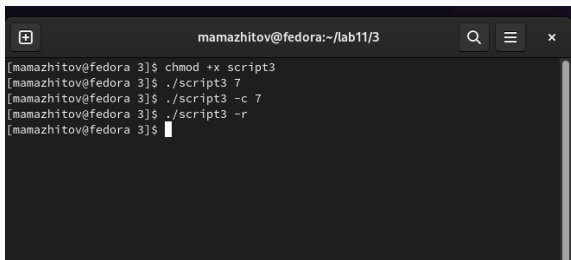


Figure 8: Результат создания

Запустил скрипт для удаления файлов.(рис. 9)

A terminal window with a dark background. The title bar shows 'mamazhitov@fedora:~/lab11/3'. The terminal contains the following text:

```
[mamazhitov@fedora 3]$ chmod +x script3  
[mamazhitov@fedora 3]$ ./script3 7  
[mamazhitov@fedora 3]$ ./script3 -c 7  
[mamazhitov@fedora 3]$ ./script3 -r  
[mamazhitov@fedora 3]$
```

Figure 9: Удаление файлов

Результат.(рис. 10)

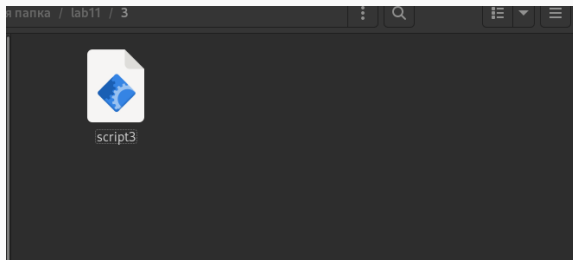
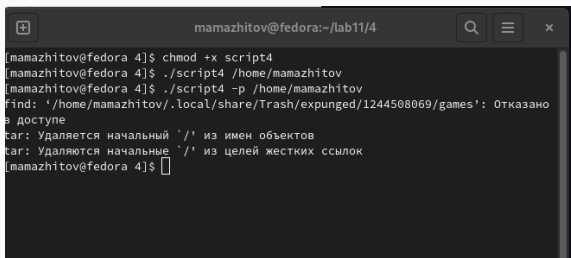


Figure 10: Результат удаления

Открыл в emacs файл *script4*. Написал программу, которая с помощью команды `tar` запаковывает в архив все файлы в указанной директории. Модифицировал его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (рис. 11)

A screenshot of a terminal window with a dark background. The window title is 'mamazhitov@fedora:~/lab11/4'. The terminal shows the following commands and output:

```
[mamazhitov@fedora 4]$ chmod +x script4
[mamazhitov@fedora 4]$ ./script4 /home/mamazhitov
[mamazhitov@fedora 4]$ ./script4 -p /home/mamazhitov
find: '/home/mamazhitov/.local/share/Trash/expunged/1244508069/games': Отказано
в доступе
tar: Удаляется начальный '/' из имен объектов
tar: Удаляются начальные '/' из целей жестких ссылок
[mamazhitov@fedora 4]$
```

Figure 11: Код 4 скрипта

Проверил работу скрипта. (рис. 12)

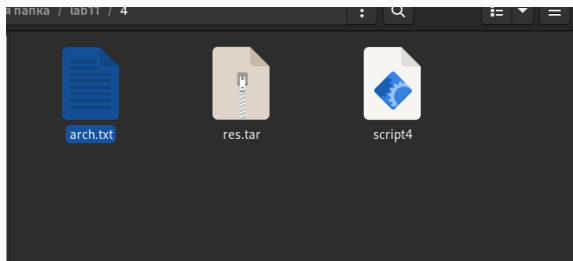


Figure 12: Проверка

Мы научились писать более сложные командные файлы.