

Презентация по лабораторной работе №11

Работу выполнил Шеожев Аслан Аскерович

Цель работы:

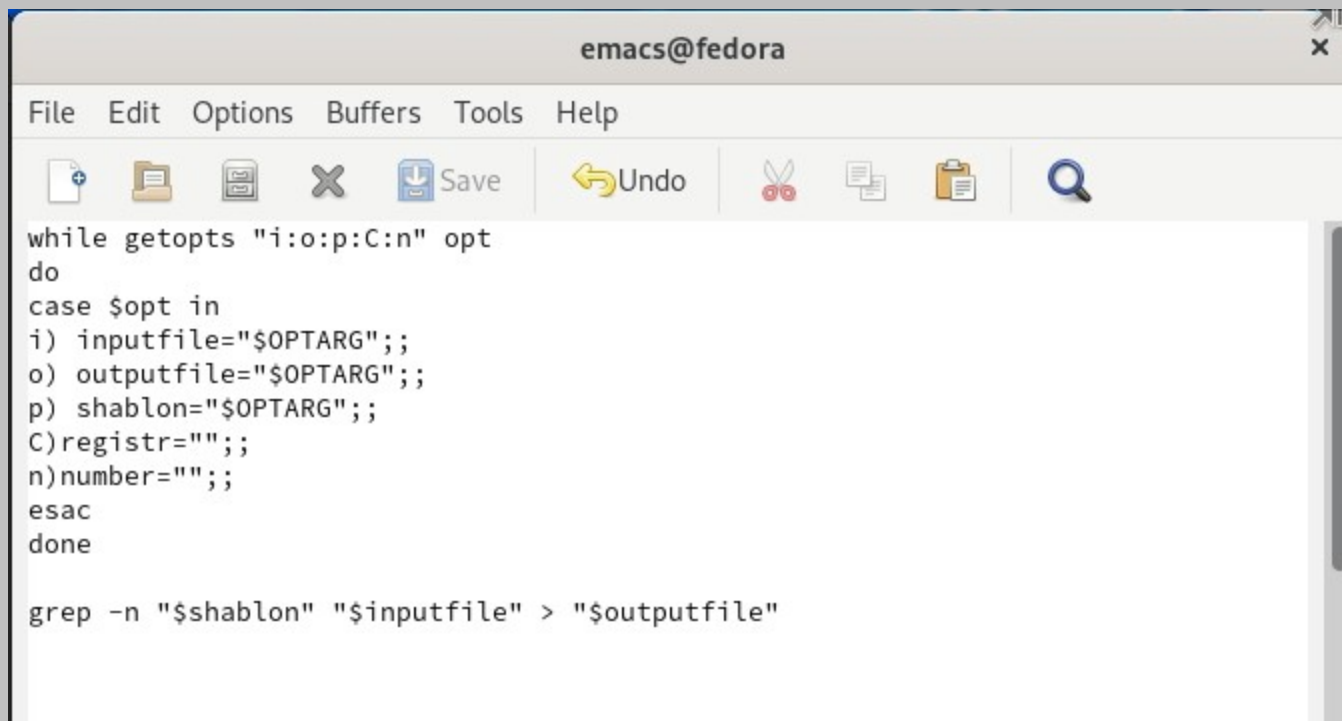
- Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Ход работы:

1. Используя команды `getopts` `grep`, написал 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.

затем ищет в указанном файле нужные строки, определяемые ключом `-r` (Рис. 1).

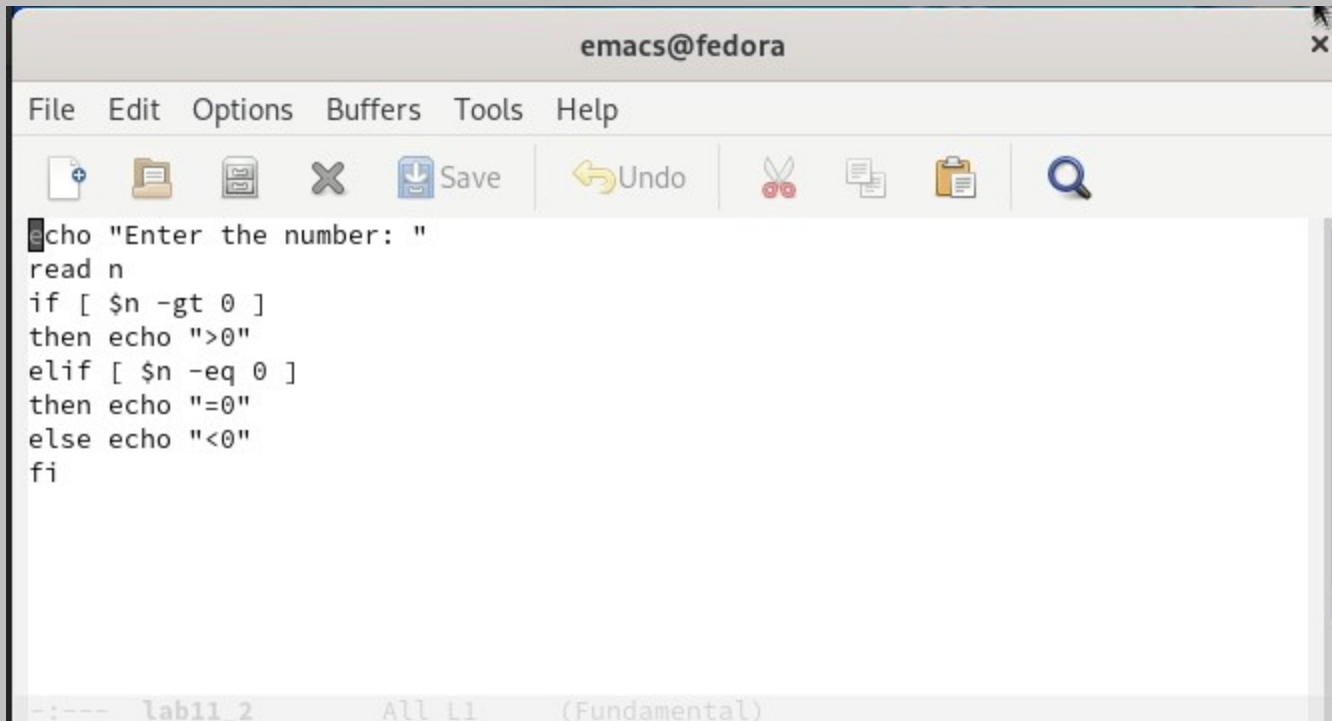


```
while getopts "i:o:p:C:n" opt
do
case $opt in
i) inputfile="$OPTARG";;
o) outputfile="$OPTARG";;
p) shablon="$OPTARG";;
C)registr="";;
n)number="";;
esac
done

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

Рис. 1 (Программа 1)

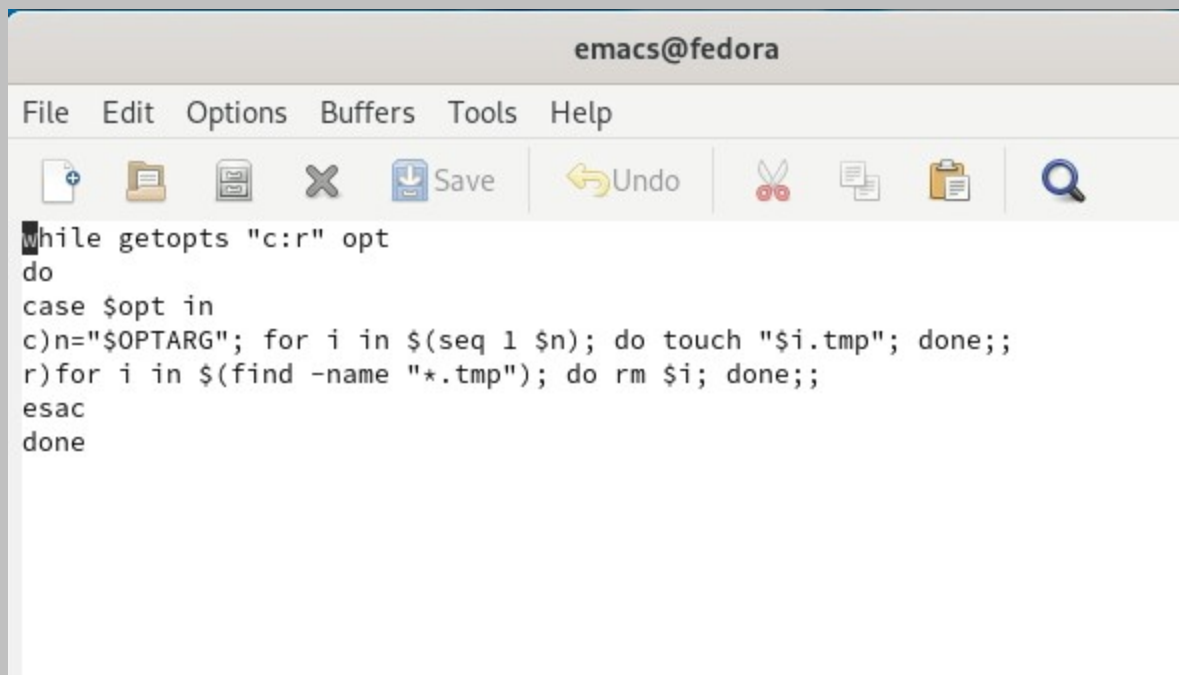
2. Написал программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю (Рис. 2).

The image shows a screenshot of the Emacs text editor window. The title bar at the top reads "emacs@fedora". Below the title bar is a menu bar with "File", "Edit", "Options", "Buffers", "Tools", and "Help". Underneath the menu bar is a toolbar with icons for opening a file, saving, undo, redo, and search. The main text area contains a shell script. The status bar at the bottom shows the file name "lab11_2", the line and column "All 11", and the buffer name "(Fundamental)".

```
echo "Enter the number: "  
read n  
if [ $n -gt 0 ]  
then echo ">0"  
elif [ $n -eq 0 ]  
then echo "=0"  
else echo "<0"  
fi
```

Рис. 2 (Программа 2)

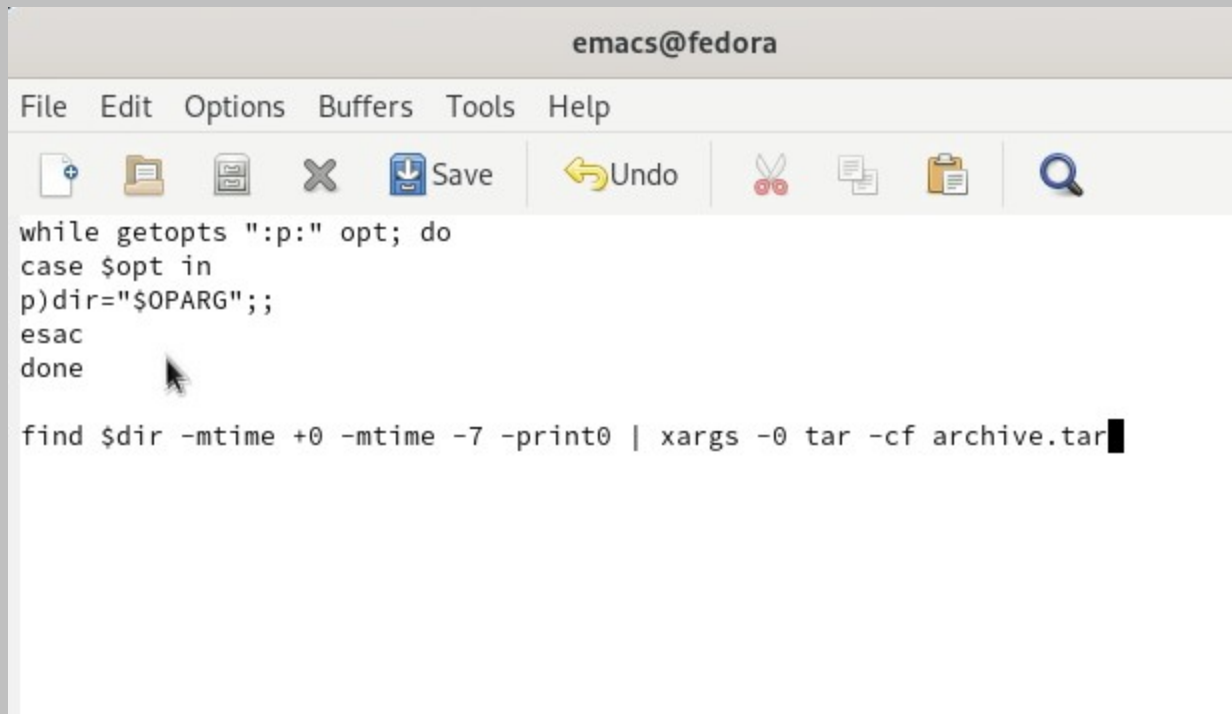
3. Написал командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N (например 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл умеет удалять все созданные им файлы (Рис. 3).

The image shows a screenshot of the Emacs text editor window. The title bar at the top reads "emacs@fedora". Below the title bar is a menu bar with the following items: "File", "Edit", "Options", "Buffers", "Tools", and "Help". Underneath the menu bar is a toolbar containing several icons: a document with a plus sign, a folder, a document with a checkmark, a document with an 'X', a floppy disk labeled "Save", a curved arrow labeled "Undo", a pair of scissors, a document with a plus sign, a document with a checkmark, and a magnifying glass. The main text area of the editor contains a shell script written in a monospaced font. The script is as follows:

```
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
```

Рис. 3 (Программа 3)

4. Написал командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. (Рис. 4)



The image shows a screenshot of the Emacs text editor window. The title bar at the top reads "emacs@fedora". Below the title bar is a menu bar with the following items: "File", "Edit", "Options", "Buffers", "Tools", and "Help". Underneath the menu bar is a toolbar containing several icons: a document with a plus sign, a folder, a document with a checkmark, a document with an 'X', a floppy disk labeled "Save", a yellow arrow pointing left labeled "Undo", a pair of scissors, two overlapping documents, a clipboard, and a magnifying glass. The main editing area contains a shell script. The script starts with a "while" loop that iterates over command-line options starting with "p:". Inside the loop, it uses a "case" statement to set a variable "dir" to the value of the option's argument. After the loop, it uses "find" to locate files in the specified directory that are newer than 7 days old, and pipes the results to "xargs" to create a tar archive named "archive.tar".

```
while getopts ":p:" opt; do
case $opt in
p)dir="$OPTARG";;
esac
done

find $dir -mtime +0 -mtime -7 -print0 | xargs -0 tar -cf archive.tar
```

Рис. 4 (Программа 4)

Вывод:

- Я изучил основы программирования в оболочке ОС UNIX. Научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.