

# Лабораторная работа №12

---

Егорова Александра<sup>1</sup>

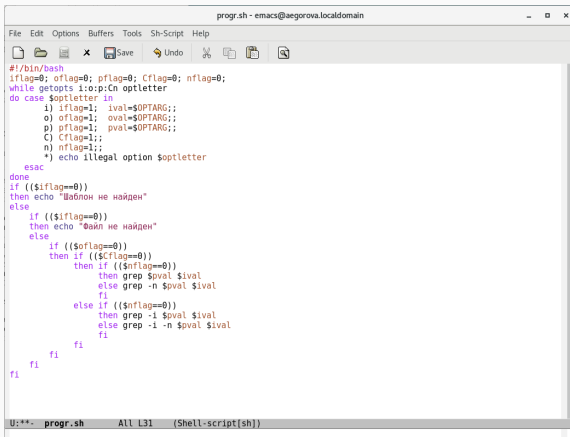
27 мая, 2021, Москва, Россия

<sup>1</sup>RUDN University, Moscow, Russian Federation

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

# Выполнение лабораторной работы

- 1) Используя команды getopts grep, написала командный файл, который анализирует командную строку с ключами (рис. -fig. 1)

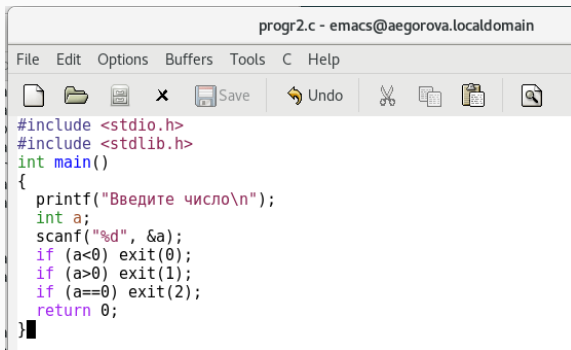
The image shows a screenshot of an Emacs editor window. The title bar at the top reads 'progr.sh - emacs@aegorova.localdomain'. Below the title bar is a menu bar with 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. A toolbar with various icons is located below the menu bar. The main editing area contains a shell script with the following content:

```
#!/bin/bash
iflag=0; oflag=0; pflag=0; Cflag=0; nflag=0;
while getopts i:op:Cn optletter
do case $optletter in
  i) iflag=1; ival=$OPTARG;;
  o) oflag=1; oval=$OPTARG;;
  p) pflag=1; pval=$OPTARG;;
  C) Cflag=1;;
  n) nflag=1;;
  *) echo illegal option $optletter
  esac
done
if (($iflag==0))
then echo "Шаблон не найден"
else
  if (($oflag==0))
  then echo "Файл не найден"
  else
    if (($soflag==0))
    then if (($Cflag==0))
    then if (($nflag==0))
    then grep $pval $ival
    else grep -n $pval $ival
    fi
    else if (($nflag==0))
    then grep -i $pval $ival
    else grep -i -n $pval $ival
    fi
    fi
  fi
fi
```

The status bar at the bottom of the window displays 'U:\*\*\* progr.sh All L31 (Shell-script[sh])'.

Figure 1: Первая программа

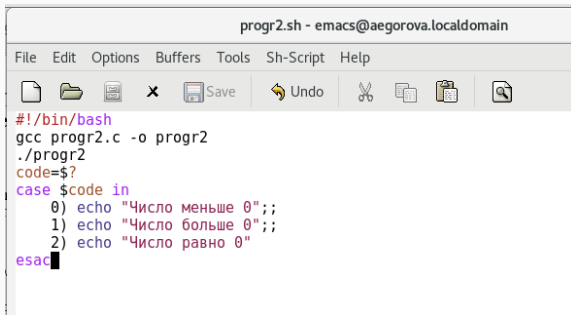
- 2) Написала на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. (рис. -fig. 2)



```
progr2.c - emacs@aegorova.localdomain
File Edit Options Buffers Tools C Help
[Icons: New, Open, Save, Undo, Cut, Copy, Paste, Find]
#include <stdio.h>
#include <stdlib.h>
int main()
{
    printf("Введите число\n");
    int a;
    scanf("%d", &a);
    if (a<0) exit(0);
    if (a>0) exit(1);
    if (a==0) exit(2);
    return 0;
}
```

Figure 2: Вторая программа на с

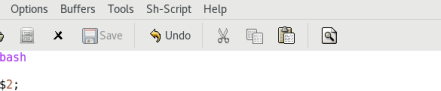
Для данной задачи я создала 2 файла: progr2.c и progr2.sh и написала соответствующие скрипты. (рис. -fig. 3)

The image shows a screenshot of an Emacs editor window. The title bar at the top reads 'progr2.sh - emacs@aegorova.localdomain'. Below the title bar is a menu bar with 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. Underneath the menu bar is a toolbar with icons for creating a new file, opening a file, saving, closing, and other standard editing functions. The main text area contains the following shell script code:

```
#!/bin/bash
gcc progr2.c -o progr2
./progr2
code=$?
case $code in
  0) echo "Число меньше 0" ;;
  1) echo "Число больше 0" ;;
  2) echo "Число равно 0" ;;
esac
```

Figure 3: Вторая программа sh

- 3) Написала командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N. 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. -fig. 4)



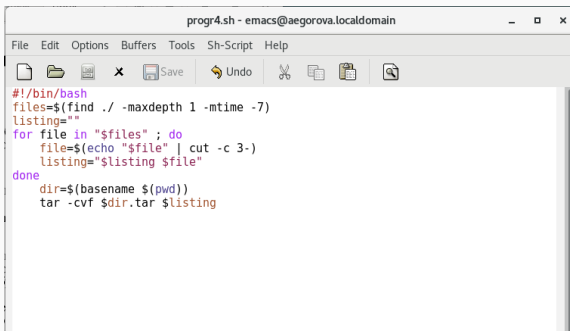
```
progr3.sh - emacs@aegorova.localdomain

File Edit Options Buffers Tools Sh-Script Help

[Icons: New File, Open File, Save, Undo, Cut, Copy, Paste, Find]

#!/bin/bash
opt=$1;
format=$2;
format=$3;
function Files()
{
    for (( i=1; i<=$number; i++)) do
        file=$(echo $format | tr '#' "$i")
        if [ $opt == "-r" ]
        then
            rm -f $file
        elif [ $opt == "-c" ]
        then
            touch $file
        fi
    done
}
Files
```

- 4) Написала командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировала его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад. (рис. -fig. 5)

The image shows a screenshot of an Emacs editor window. The title bar reads 'progr4.sh - emacs@aegorova.localdomain'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. Below the menu is a toolbar with icons for file operations like 'Save', 'Undo', 'Cut', 'Copy', 'Paste', and 'Find'. The main text area contains a shell script with the following content:

```
#!/bin/bash
files=$(find ./ -maxdepth 1 -mtime -7)
listing=""
for file in "$files" ; do
    file=$(echo "$file" | cut -c 3-)
    listing="$listing $file"
done
dir=$(basename $(pwd))
tar -cvf $dir.tar $listing
```

Figure 5: Четвертая программа

В ходе выполнения данной лабораторной работы я изучила основы программирования в оболочке ОС UNIX и научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.