

Презентация к лабораторной работе №11

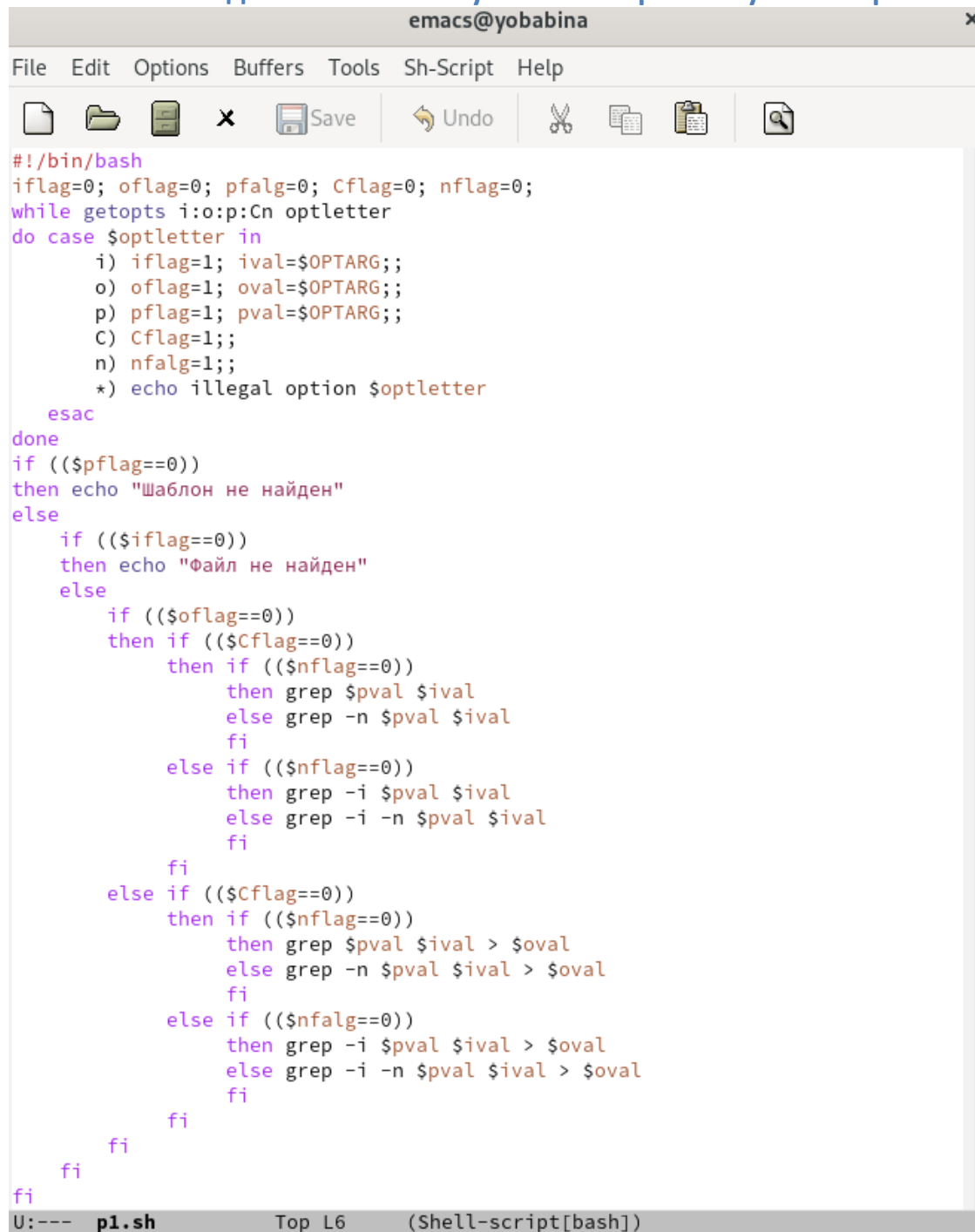
Бабина Ю.О.

Цель работы

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

Ход работы

Выполнение задания 1. Поиск в указанном файле нужной строки.



```
#!/bin/bash
iflag=0; oflag=0; pflag=0; cflag=0; nflag=0;
while getopts i:o:p:C:n 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 $optletter
    esac
done
if (($pflag==0))
then echo "Шаблон не найден"
else
    if (($iflag==0))
    then echo "Файл не найден"
    else
        if (($oflag==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 $ival
                fi
            fi
        else if (($cflag==0))
            then if (($nflag==0))
                then grep $pval $ival > $oval
                else grep -n $pval $ival > $oval
                fi
            else if (($nflag==0))
                then grep -i $pval $ival > $oval
                else grep -i -n $pval $ival > $oval
                fi
            fi
        fi
    fi
fi
fi
```

U:--- p1.sh Top L6 (Shell-script[bash])

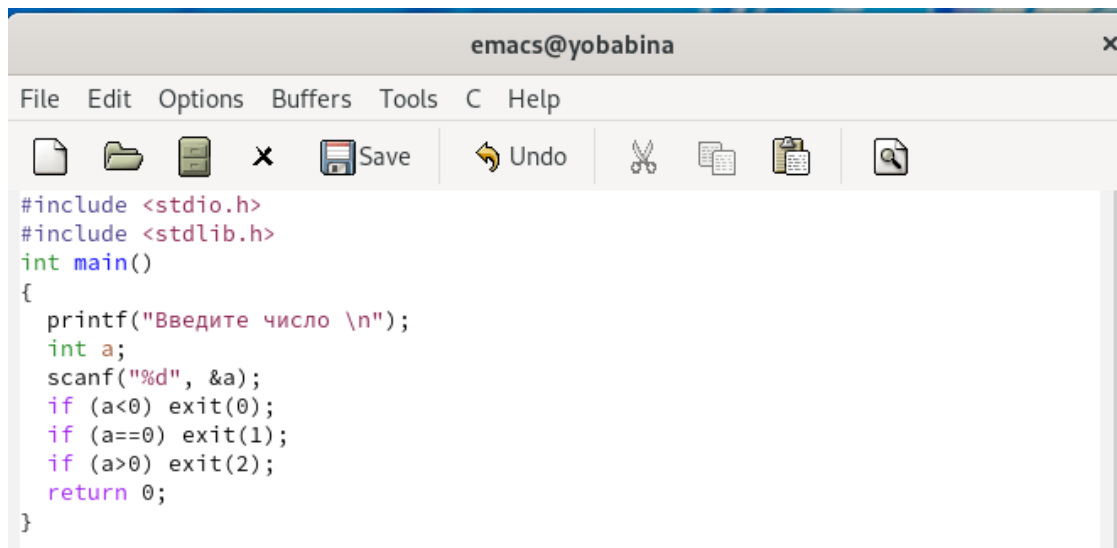
код первой программы

```
[babinayuliaolegovna@yobabina ~]$ ./p1.sh -i one.txt -o two.txt -p Изучить -C -n
[babinayuliaolegovna@yobabina ~]$ cat one.txt
Изучить основы программирования в оболочке ОС UNIX. Научится писать более
сложные командные файлы с использованием логических управляющих конструкций
и циклов.

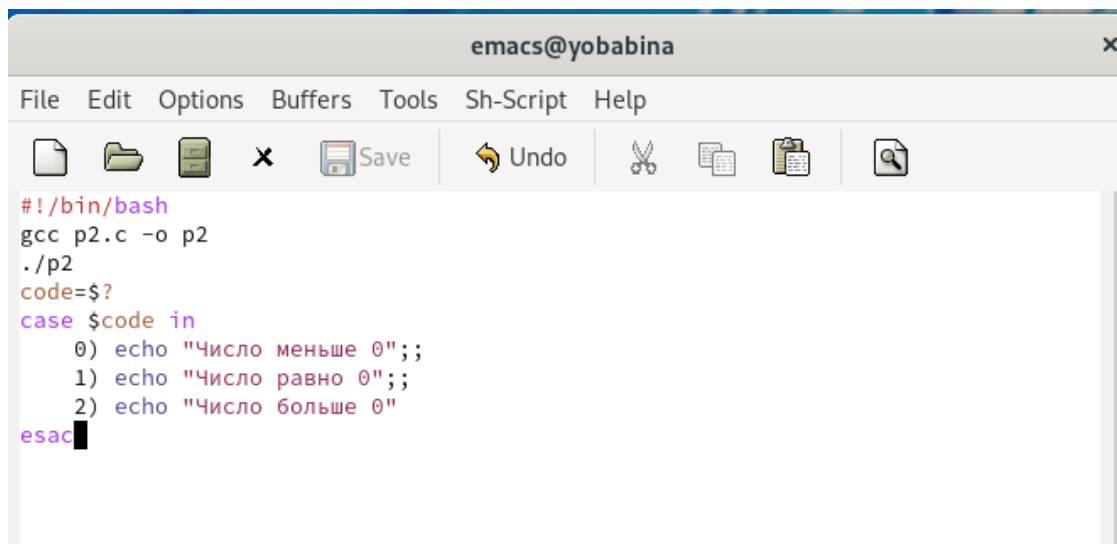
[babinayuliaolegovna@yobabina ~]$ cat two.txt
1:Изучить основы программирования в оболочке ОС UNIX. Научится писать более
[babinayuliaolegovna@yobabina ~]$
```

результат выполнения первой программы

Выполнение задания 2. Вызов программы С через командный файл.



```
#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;
}
```



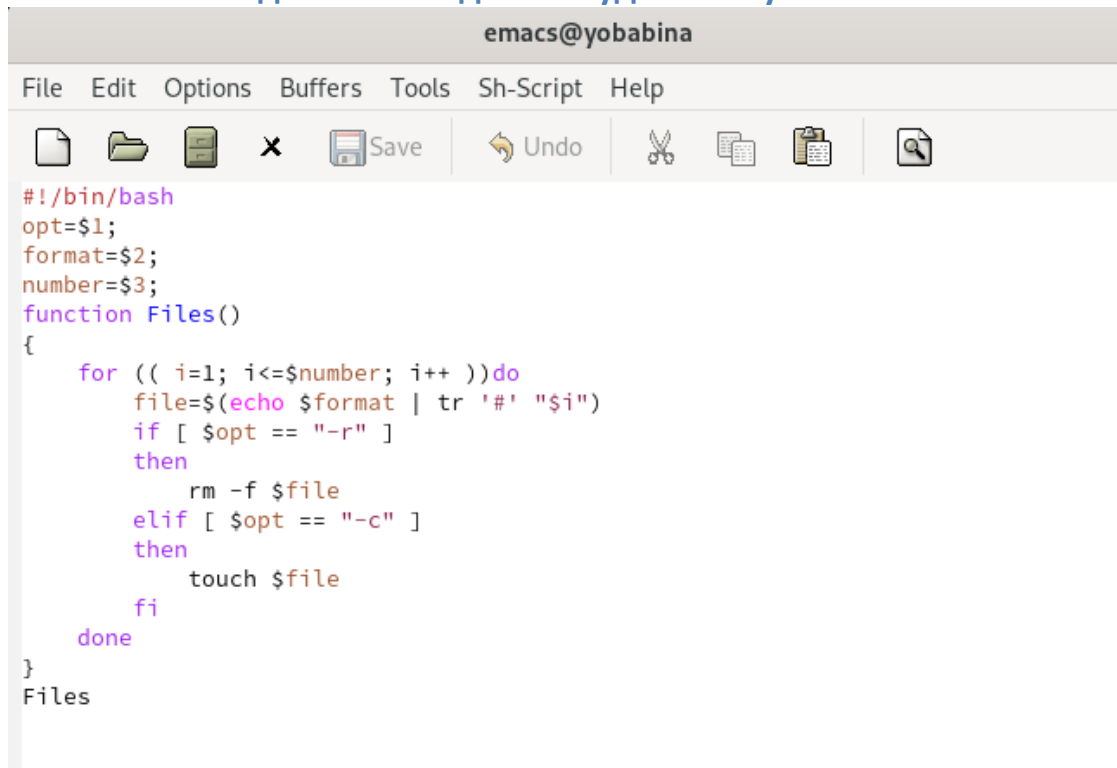
```
#!/bin/bash
gcc p2.c -o p2
./p2
code=$?
case $code in
    0) echo "Число меньше 0";;
    1) echo "Число равно 0";;
    2) echo "Число больше 0"
esac
```

коды программы

```
[babinayuliaolegovna@yobabina ~]$ chmod +x p2.sh
[babinayuliaolegovna@yobabina ~]$ ./p2.sh
Введите число
5
Число больше 0
[babinayuliaolegovna@yobabina ~]$
```

результат выполнения второй программы

Выполнение задания 3. Создание и удаление указанного количества файлов.

The image shows a screenshot of the Emacs text editor interface. The title bar at the top reads "emacs@yobabina". Below the title bar is a menu bar with the following items: "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", and "Help". Underneath the menu bar is a toolbar containing several icons: a file icon, a folder icon, a save icon, a close icon, a "Save" button, an "Undo" button, a scissors icon, a document icon, a clipboard icon, and a search icon. The main editing area contains a shell script written in a syntax-highlighted format. The script starts with a shebang line "#!/bin/bash", followed by variable assignments: "opt=\$1;", "format=\$2;", and "number=\$3;". It then defines a function "function Files()" which contains a "for" loop. The loop iterates from "i=1" to "i=\$number". Inside the loop, it calculates a file name "file=\$(echo \$format | tr '#' '\$i')". It then checks if the option "opt" is equal to "-r". If so, it runs "rm -f \$file". If "opt" is equal to "-c", it runs "touch \$file". The loop ends with "done", and the function ends with "fi" and "}".

код третьей программы

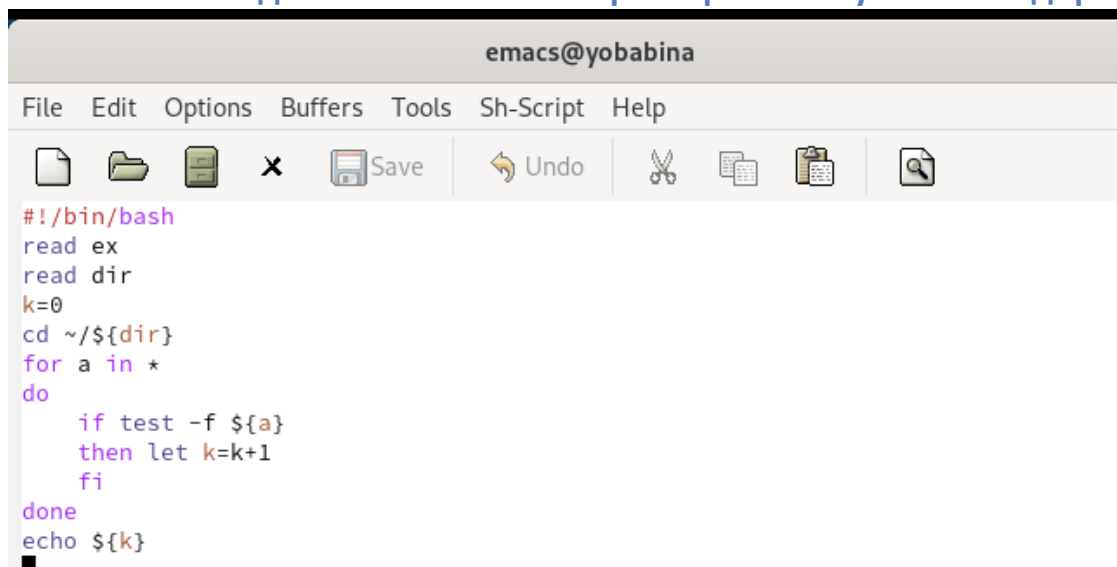
```

[babinayuliaolegovna@yobabina ~]$ chmod +x p3.sh
[babinayuliaolegovna@yobabina ~]$ ./p3.sh -c aaa#.txt 5
[babinayuliaolegovna@yobabina ~]$ ls
aaa1.txt  etc      monthly  p2.sh    s3.sh    Документы
aaa2.txt  feathers my_os    p3.sh    s3.sh~   Загрузки
aaa3.txt  file.txt newdir   p3.sh~   s4.sh    Изображения
aaa4.txt  games   one.txt  reports  s4.sh~   Музыка
aaa5.txt  june    p1.sh    s1.sh    script.cpp Общедоступные
australia l9       p1.sh~   s1.sh~   two.txt  'Рабочий стол'
backup    lab09.sh p2       s2.sh    work     Шаблоны
conf.txt  lab09.sh~ p2.c     s2.sh~   yulia2
directory may      p2.c~    '#s3.sh#' Видео
[babinayuliaolegovna@yobabina ~]$ ./p3.sh -r aaa#.txt 5
[babinayuliaolegovna@yobabina ~]$ ls
australia games  my_os  p2.c~  s2.sh  script.cpp  Изображения
backup    june   newdir p2.sh  s2.sh~  two.txt     Музыка
conf.txt  l9     one.txt p3.sh  '#s3.sh#' work        Общедоступные
directory lab09.sh p1.sh  p3.sh~ s3.sh   yulia2     'Рабочий стол'
etc       lab09.sh~ p1.sh~ reports s3.sh~  Видео      Шаблоны
feathers  may     p2      s1.sh  s4.sh   Документы
file.txt  monthly p2.c    s1.sh~ s4.sh~  Загрузки
[babinayuliaolegovna@yobabina ~]$

```

результат выполнения третьей программы

Выполнение задания 4. Упаковка в архив файлов в указанной директории



The screenshot shows the Emacs editor window titled 'emacs@yobabina'. The menu bar includes File, Edit, Options, Buffers, Tools, Sh-Script, and Help. The toolbar contains icons for file operations like Save, Undo, Cut, Copy, and Paste. The main text area displays a shell script:

```

#!/bin/bash
read ex
read dir
k=0
cd ~/${dir}
for a in *
do
    if test -f ${a}
    then let k=k+1
    fi
done
echo ${k}

```

код четвертой программы

```

[babinayuliaolegovna@yobabina ~]$ emacs
[babinayuliaolegovna@yobabina ~]$ chmod +x p4.sh
[babinayuliaolegovna@yobabina ~]$ mkdir catt
[babinayuliaolegovna@yobabina ~]$ cd catt
[babinayuliaolegovna@yobabina catt]$ ls
file.txt june one.txt
[babinayuliaolegovna@yobabina catt]$ ~/p4.sh
one.txt
[babinayuliaolegovna@yobabina catt]$

```

результат выполнения четвертой программы

Вывод

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