

Лабораторная работа №11

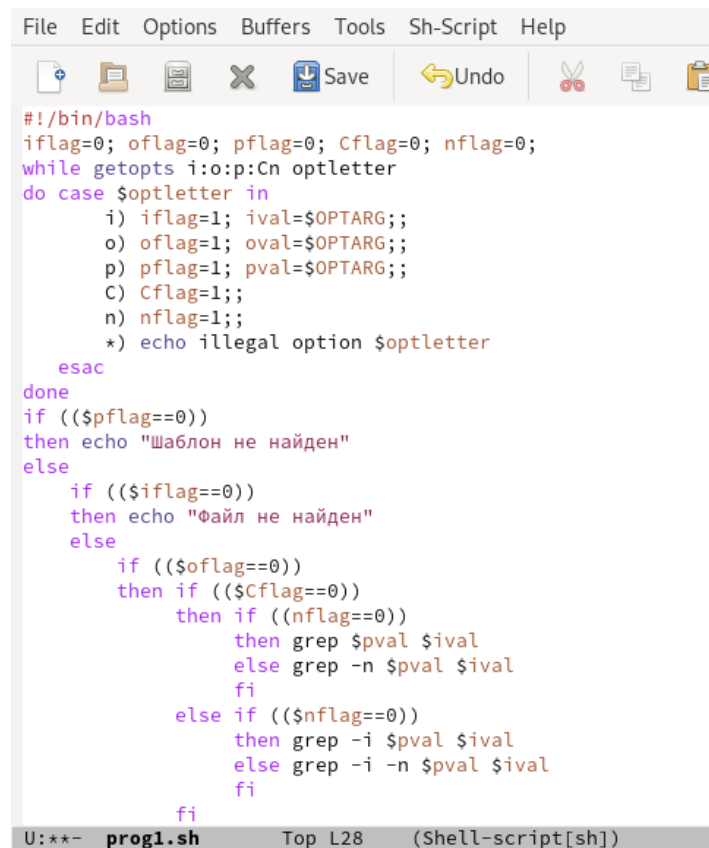
Макарова Анастасия Михайловна

Цель работы

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

Первая программа

Используя команды `getopts` `grep`, напомним командный файл, который анализирует командную строку с ключами: `* -i` inputfile; `* -o` outputfile; `* -p` шаблон; `* -C`; `* -n`. А затем ищет в указанном файле нужные строки, определяемые ключом `-p`



The image shows a screenshot of a text editor window titled "Shell-script". The window has a menu bar with "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", and "Help". Below the menu bar is a toolbar with icons for opening, saving, undo, redo, and other editing functions. The main text area contains a shell script written in a syntax-highlighted format. The script starts with a shebang line and initializes several flags. It then enters a loop that reads command-line options and processes them. After the loop, it checks if the pattern flag is set, and if so, it searches for the pattern in the file. The script ends with a status bar showing the current file name, line number, and the shell being used.

```
#!/bin/bash
iflag=0; oflag=0; pflag=0; Cflag=0; nflag=0;
while getopts i:o:p: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 $optletter
    esac
done
if (($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
        fi
    fi
fi
```

U:**- **prog1.sh** Top L28 (Shell-script[sh])

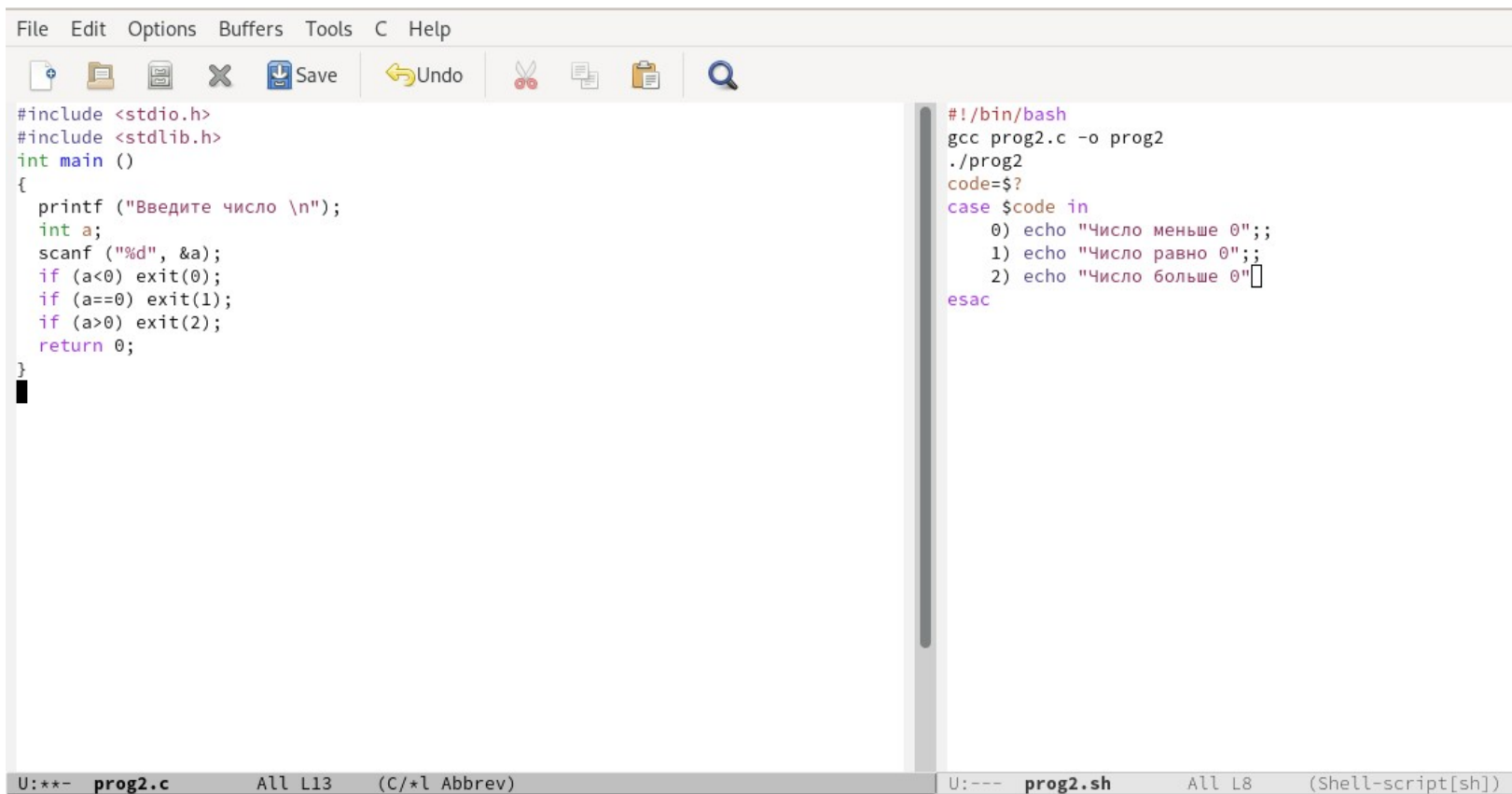
Написание программы

```
[ammakarova@10 ~]$ ./prog1.sh -i a.txt -o b.txt -p money -C -n  
[ammakarova@10 ~]$ cat a.txt  
I wanna money money money  
[ammakarova@10 ~]$ cat b.txt  
1:I wanna money money money
```

Работа программы

Вторая программа

Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю.



The image shows a code editor window with two files open. The left pane displays the C source code for `prog2.c`, and the right pane displays the shell script for `prog2.sh`. The editor has a menu bar (File, Edit, Options, Buffers, Tools, C, Help) and a toolbar with icons for file operations and search. The status bar at the bottom shows the current file, line number, and character count.

```
#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 prog2.c -o prog2
./prog2
code=$?
case $code in
    0) echo "Число меньше 0";;
    1) echo "Число равно 0";;
    2) echo "Число больше 0";;
esac
```

U:*** prog2.c All L13 (C/*l Abbrev) U:--- prog2.sh All L8 (Shell-script[sh])

Написание программы

```
[ammakarova@10 ~]$ chmod +x prog2.sh
[ammakarova@10 ~]$ ./prog2.sh
Введите число
7
Число больше 0
[ammakarova@10 ~]$ ./prog2.sh
Введите число
0
Число равно 0
[ammakarova@10 ~]$ ./prog2.sh
Введите число
-3
Число меньше 0
```

Работа программы

Третья программа

Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N .

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

U:*** prog3.sh All L18 (Shell-script[sh])

Написание программы

```

[ammakarova@10 ~]$ chmod +x prog3.sh
[ammakarova@10 ~]$ ./prog3.sh -c abc#.txt 4
[ammakarova@10 ~]$ ls
abc1      b.txt      file2.sh   play       text.txt
abc1.txt  catalog    file3.sh   prog1.sh   work
abc2.txt  '#command_file_3#' file3.sh~   prog1.sh~  Видео
abc3.txt  command_file_3.sh file4.sh    prog2       Документы
abc4.txt  command_file_4.sh lab07.sh    prog2.c     Загрузки
a.txt     command_file_4.sh~ lab07.sh~   prog2.cpp   Изображения
australia command_file.sh   may         prog2.sh    Музыка
backup    conf.txt         monthly     prog3.sh    Общедоступные
backup.sh feathers        my_os       reports     'Рабочий стол'
backup.sh~ file1.sh         OS_2022     sci.plases  Шаблоны

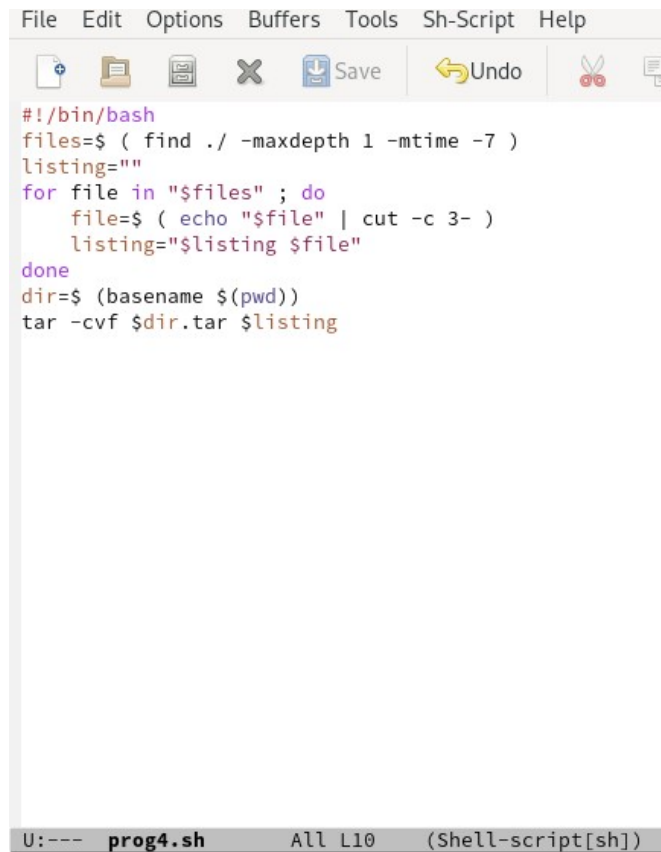
[ammakarova@10 ~]$ ./prog3.sh -r abc#.txt 4
[ammakarova@10 ~]$ ls
abc1      command_file.sh  my_os      text.txt
a.txt     conf.txt         OS_2022    work
australia feathers        play       Видео
backup    file1.sh         prog1.sh   Документы
backup.sh file2.sh         prog1.sh~  Загрузки
backup.sh~ file3.sh         prog2       Изображения
b.txt     file3.sh~        prog2.c     Музыка
catalog   file4.sh         prog2.cpp   Общедоступные
'#command_file_3#' lab07.sh         prog2.sh    'Рабочий стол'
command_file_3.sh lab07.sh~        prog3.sh    Шаблоны
command_file_4.sh may              reports
command_file_4.sh~ monthly         sci.plases

```

Работа программы

Четвертая программа

Написать командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории.



The image shows a screenshot of a shell script editor window. The window has a menu bar with 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. Below the menu bar is a toolbar with icons for opening a file, saving, closing, and undo, along with text labels 'Save' and 'Undo'. The main area of the window contains a shell script written in bash. The script finds files in the current directory and its subdirectories (up to a depth of 1) that were modified within the last 7 days. It then iterates over these files, extracts their names, and appends them to a list. Finally, it creates a tar archive named 'dir.tar' containing the list of files. The status bar at the bottom of the window shows 'U:--- prog4.sh All L10 (Shell-script[sh])'.

```
#!/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
```

U:--- **prog4.sh** All L10 (Shell-script[sh])

Написание программы

```
[ammakarova@10 catalog1]$ ~/prog4.sh  
prog1.sh  
[ammakarova@10 catalog1]$ ls  
catalog1.tar  lab07.sh  prog1.sh
```

Работа программы

Выводы

- 1) Изучила основы программирования в оболочке ОС UNIX/Linux;
- 2) Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

СПАСИБО ЗА ВНИМАНИЕ :)