

# Отчет по лабораторной работе №12

---

Перельгин Сергей Викторович

## Цель работы

---

## Цель лабораторной работы

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

# **Выполнение лабораторной работы**

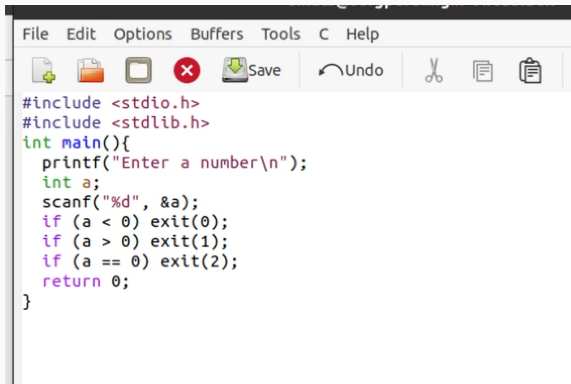
---

Используя команды `getopts` `grep`, написал командный файл, который анализирует командную строку с ключами, а затем ищет в указанном файле нужные строки, определяемые ключом `-p`.

```
#!/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 "НЕТ шаблона"
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
```

Рис. 1: Скрипт 1

Написал 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 код завершения оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.

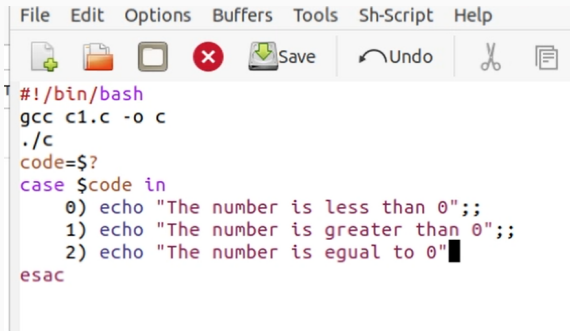


The image shows a screenshot of a C code editor window. The window has a menu bar with 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'C', and 'Help'. Below the menu bar is a toolbar with icons for file operations (new, open, save, close, save all), editing (undo, redo, cut, copy, paste), and a search icon. The main area of the window contains the following C code:

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    printf("Enter a number\n");
    int a;
    scanf("%d", &a);
    if (a < 0) exit(0);
    if (a > 0) exit(1);
    if (a == 0) exit(2);
    return 0;
}
```

**Рис. 2:** Скрипт 2

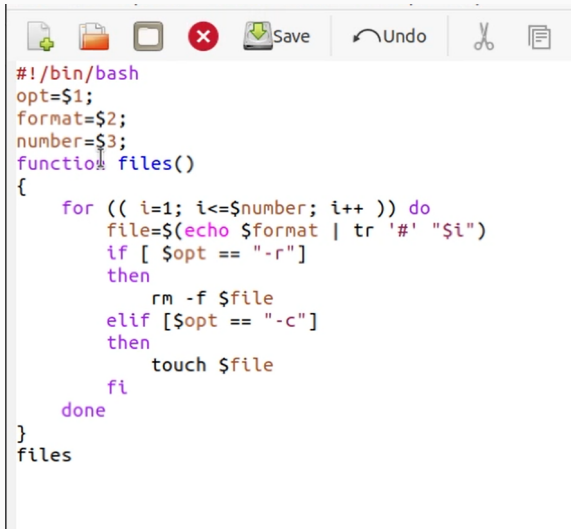




```
#!/bin/bash
gcc c1.c -o c
./c
code=$?
case $code in
    0) echo "The number is less than 0";;
    1) echo "The number is greater than 0";;
    2) echo "The number is equal to 0";;
esac
```

Рис. 3: Скрипт 2

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 должен уметь удалять все созданные им файлы (если они существуют).

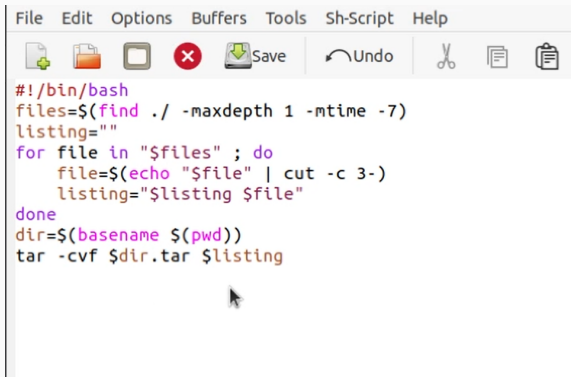


```
#!/bin/bash
opt=$1;
format=$2;
number=$3;
function files()
{
    for (( i=1; i<=$number; i++ )) do
        file=$(echo $format | tr '#' "$i")
        if [ $opt == "-r" ]
        then
            rm -f $file
        elif [ $opt == "-c" ]
        then
            touch $file
        fi
    done
}
files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

Рис. 4: Скрипт 3

Написал командный файл, который 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(использовать команду `find`).

A screenshot of a terminal window with a menu bar (File, Edit, Options, Buffers, Tools, Sh-Script, Help) and a toolbar with icons for file operations. The terminal displays a shell script that finds files in the current directory, lists them, and archives them into a tar file.

```
#!/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
```

Рис. 5: Скрипт 4

## Выводы

---

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

Спасибо за внимание!