

# Лабораторная №11

---

Газизова Регина<sup>4</sup>

26 мая 2022, Москва

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

## Первый командный файл

---

# Первый командный файл

- Поиск в указанном файле нужные строки

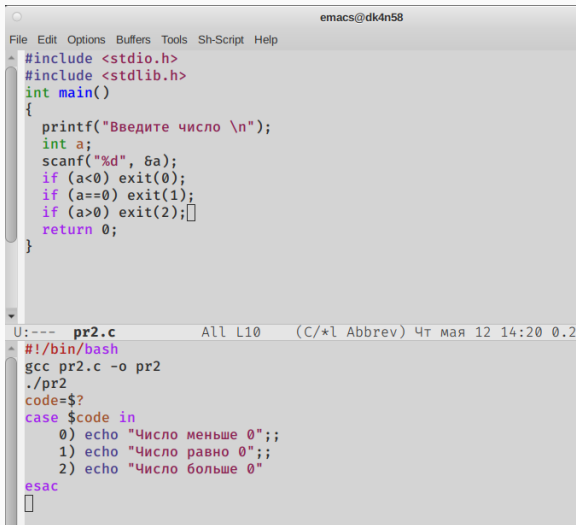
```
#!/bin/bash
iflag=0; oflag=0; pflag=0; Cflag=0; nflag=0;
while getopts i:o:p:Cn optletter
do case $optletter in
    i) iflag=1; ival=$OPTARG;;
    o) oflag=1;  oval=$OPTARG;;
    p) pflag=1;  pval=$OPTARG;;
    C) Cflag=1;;
    n) nflag=1;;
    *) echo illegal option $optletter
    esac
done
if (($pflag==0))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

## Второй командный файл

---

## Второй командный файл

- Вызов программы С через командный файл



The screenshot shows an Emacs editor window titled 'emacs@dk4n58'. The top menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The first file, 'pr2.c', is a C program that prompts the user to enter a number and then exits with a status code based on the input. The second file is a shell script that compiles 'pr2.c' and runs it, then uses a 'case' statement to print a message based on the exit code.

```
emacs@dk4n58
File Edit Options Buffers Tools Sh-Script Help
^ #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;
  }

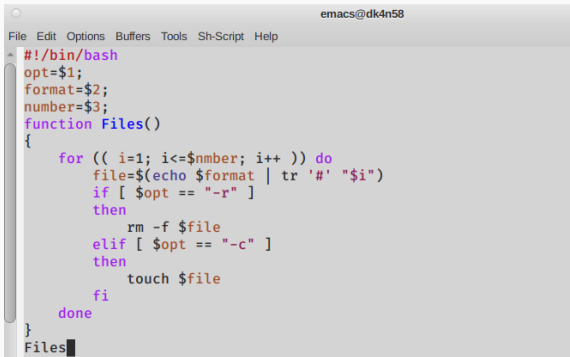
U:--- pr2.c All L10 (C/*l Abbrev) Чт мая 12 14:20 0.2
^ #!/bin/bash
gcc pr2.c -o pr2
./pr2
code=$?
case $code in
  0) echo "Число меньше 0";;
  1) echo "Число равно 0";;
  2) echo "Число больше 0"
esac
^
```

Figure 2: Программы

## Третий командный файл

---

- Создание указанного количества файлов

The image shows a screenshot of an Emacs editor window. The title bar at the top reads 'emacs@dk4n58'. Below the title bar is a menu bar with the following items: 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The main editing area contains a shell script. The script starts with a shebang line '#!/bin/bash' in red. It then sets three variables: 'opt=\$1' (blue), 'format=\$2' (blue), and 'number=\$3' (blue). A function named 'Files()' is defined in blue. Inside the function, there is a 'for' loop in blue that iterates from 'i=1' to 'i=\$number'. Inside the loop, a file path is constructed using 'file=\$(echo \$format | tr '#' "\$i")'. Then, there is an 'if' statement in blue that checks if '\$opt == "-r"'. If true, it runs 'rm -f \$file'. Otherwise, it runs 'touch \$file'. The loop ends with 'done' and the function ends with '}' and 'Files' followed by a cursor. The script is color-coded: keywords are blue, variables are blue, and file names/paths are blue. The background is a light gray with a vertical scrollbar on the left.

```
#!/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
```

Figure 3: Командный файл

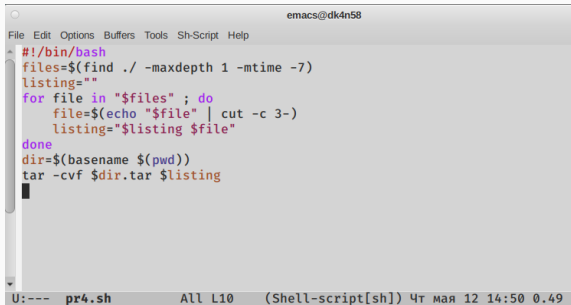
## Четвертый командный файл

---



## Четвертый командный файл

- Упаковка в архив всех файлов в текущей директории



```
emacs@dk4n58
File Edit Options Buffers Tools Sh-Script Help
^_ #!/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:--- pr4.sh All L10 (Shell-script[sh]) Чт мая 12 14:50 0.49
```

Figure 4: Командный файл

That's all