

# Программирование в командном процессоре ОС UNIX. Ветвления и циклы.

Лабораторная работа №11

---

Миронова М. В.

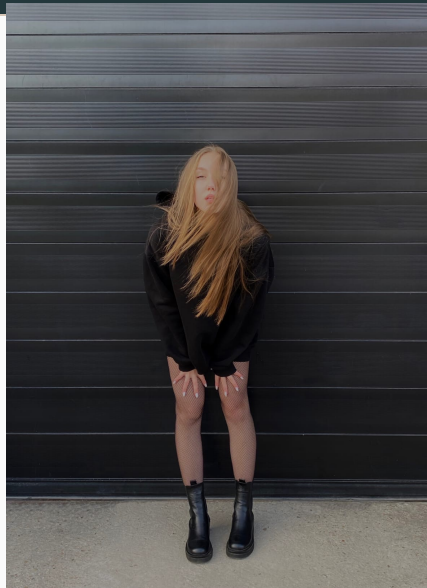
22 апреля 2023

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Миронова Мария Вадимовна
- студент 1 курса, группа НММбд-03-22
- Российский университет дружбы народов



## Вводная часть

---

- Командный процессор ОС UNIX
- Командные файлы

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

- Ознакомиться с теоретическим материалом.
- Выполнить упражнения.
- Ответить на контрольные вопросы.

# Выполнение лабораторной работы №11

---



# Первая программа


```
Открыть  lab11_1.sh
#!/bin/bash
iflag=0; oflag=0; pflag=0; cflag=0; nflag=0;
while getopts i:op: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 (($iflag==0))
    then echo "файл не найден"
    exit
    else
        if (($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
            fi
        fi
    fi
fi
```

```
[mvmironova@fedora ~]$ cat 1.txt
1
2
2
3
3
4
4
4
4


[mvmironova@fedora ~]$ ./lab11_1.sh -i 1.txt -o 2.txt -p 3 -C -n
[mvmironova@fedora ~]$ cat 2.txt
4:3
5:3
6:3
```

```
[mvmironova@fedora ~]$ ./lab11_1.sh -i 1.txt -p 2 -C -n
2:2
3:2
[mvmironova@fedora ~]$ ./lab11_1.sh -i 1.txt -C -n
шаблон не найден
[mvmironova@fedora ~]$ ./lab11_1.sh -o 2.txt -p 3 -C -n
файл не найден
```

## Вторая программа

Открыть ▾  • lab11\_2.c  
~/

```
#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;
}
```

Открыть ▾  • lab11\_2.sh  
~/

```
#!/bin/bash
gcc lab11_2.c -o lab11_2
./lab11_2
code=$?
case $code in
    0) echo "Число меньше 0";;
    1) echo "Число больше 0";;
    2) echo "Число равно 0"
esac
|
```

```
[mvmironova@fedora ~]$ ./lab11_2.sh
Введите число
1
Число больше 0
[mvmironova@fedora ~]$ ./lab11_2.sh
Введите число
-1
Число меньше 0
[mvmironova@fedora ~]$ ./lab11_2.sh
Введите число
0
Число равно 0
```

# Третья программа

Открыть ▾ 

• lab11\_3.sh

~/

```
#!/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
}
```

```
[mvmironova@fedora ~]$ chmod +x lab11_3.sh
```

```
[mvmironova@fedora ~]$ ./lab11_3.sh -c 1.txt 3
[mvmironova@fedora ~]$ ls
'1s'  's3s'  lab11_1.sh lab11_3.sh Зарплата 'Рабочий стол'
1.txt 's4s'  lab11_2 work Изображения Заблони
's2s' 'slab09.sh' lab11_2.c Индекс Музыка
2.txt lab09.sh lab11_2.sh документы Общие папки
[mvmironova@fedora ~]$ ./lab11_3.sh -r 1.txt 3
[mvmironova@fedora ~]$ ls
'1s'  's3s'  lab11_1.sh lab11_3.sh Зарплата 'Рабочий стол'
1.txt 's4s'  lab11_2 work Изображения Заблони
's2s' 'slab09.sh' lab11_2.c Индекс Музыка
2.txt lab09.sh lab11_2.sh документы Общие папки
```

# Четвёртая программа

```
Открыть ▾ + • lab11_4.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
```

```
[mvmironova@fedora ~]$ ./lab11_4.sh
.cache/
.cache/mesa_shader_cache/
.cache/mesa_shader_cache/index
.cache/mesa_shader_cache/e1/
.cache/mesa_shader_cache/e1/0424d04d5ddf09a74f6ff99c9e7517efc637d5
.cache/mesa_shader_cache/e1/21ef70ddfd0e934a62df732d3a41fb4769bd62
.cache/mesa_shader_cache/e1/2afbabca4ed92c2f850b638e9407cd5e3a83d2
.cache/mesa_shader_cache/T8/
.cache/mesa_shader_cache/T8/cf40ac01de166dd8e41468ecd082e0ba7cc690
.cache/mesa_shader_cache/T8/ecc572a1e741792e53bdc4c6717a767304dea1
.cache/mesa_shader_cache/T8/5642ecf79dc7bb519288a12f63d0a3c418c231
.cache/mesa_shader_cache/fd/
.cache/mesa_shader_cache/fd/f07c73761644fa6363746d1e5aa6256dd7e44
.cache/mesa_shader_cache/fd/129a84d9ef9974869aeac22e50253188511a92
.cache/mesa_shader_cache/fd/1436b8dbf29721277b26476c2bbb0b08cd1be0
.cache/mesa_shader_cache/36/
.cache/mesa_shader_cache/36/edbc1e8a6cbc2685c55bac94c774b8ec5c86ef
.cache/mesa_shader_cache/21/
.cache/mesa_shader_cache/21/e3410af190d065343481ad0fa65013a0142ead
.cache/mesa_shader_cache/21/eb9d549aa18352b95eeaa7d7a4b160c5fb6891
.cache/mesa_shader_cache/21/0094d88894f6304645d0343b58d78d6caa08c8
```

```
[mvmironova@fedora ~]$ ls
'21*'  '4d*'  lab11_2.c  work  Музыка
1.txt  '1ab09.shf'  lab11_3.sh  видео  Общедоступные
'28*'  lab09.sh  lab11_3.sh  документы  'Рабочий стол'
2.txt  lab11_1.sh  lab11_4.sh  загрузки  Шаблоны
'31*'  lab11_2  mvmironova.tar  Изображения
```

1. Команда `getopts` осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных. Синтаксис команды следующий: `getopts option-string variable [arg ... ]` Флаги – это опции командной строки, обычно помеченные знаком минус; Например, для команды `ls` флагом может являться `-F`. Строка опций `option-string` – это список возможных букв и чисел соответствующего флага. Если ожидается, что некоторый флаг будет сопровождаться некоторым аргументом, то за символом, обозначающим этот флаг, должно следовать двоеточие. Соответствующей переменной присваивается буква данной опции. Если команда `getopts` мо-

3. Часто бывает необходимо обеспечить проведение каких-либо действий циклически и управление дальнейшими действиями в зависимости от результатов проверки некоторого условия. Для решения подобных задач язык программирования `bash` предоставляет возможность использовать такие управляющие конструкции, как `for`, `case`, `if` и `while`. С точки зрения командного процессора эти управляющие конструкции являются обычными командами и могут использоваться как при создании командных файлов, так и при работе в интерактивном режиме. Команды, реализующие подобные конструкции, по сути, являются операторами языка программирования

## Результаты

---

В ходе выполнения лабораторной работы были изучены основы программирования в оболочке ОС UNIX. Научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.