



# Лабораторная работа

## 11

Подготовил Манатов Рамазан



## Цель работы

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

Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами:

- `-iinputfile` — прочитать данные из указанного файла;
- `-ooutputfile` — вывести данные в указанный файл;
- `-ршаблон` — указать шаблон для поиска;
- `-C` — различать большие и малые буквы;
- `-n` — выдавать номера строк

```
File Edit Options Buffers Tools Help
[Icons] Save Undo [Icons] [Icon]
#!/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 (($iflag==0))
    then echo "don't found file"
    else
        if (($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
```

# Программа которая вводит число и проверяет $> \leq 0$

Код программы


```
#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;
}
```

Проверка корректности скрипта

```
[rrManatov@fedora ~]$ ./chislo.sh
Введите число/n
1
Число больше 0
[rrManatov@fedora ~]$ ./chislo.sh
Введите число/n
-1
Число меньше 0
[rrManatov@fedora ~]$ ./chislo.sh
Введите число/n
0
Число равно 0
```

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
```



командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицируем его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find)

```
#!/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
```



## ВЫВОД

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