

# Программирование в командном процессоре ОС UNIX.

## Ветвления и циклы

---

ОЗЪЯС Стив Икнэль Дани

27 May, 2022 Moscow, Russia

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

Цель работы

---

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

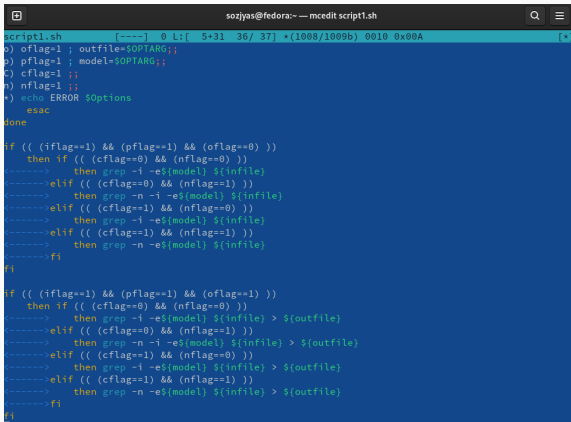
## Ход работы

---

1. Используя команды `grep`, написал командный файл, который анализирует командную строку с ключами:
  - `-i`inputfile прочитать данные из указанного файла;
  - `-o`outputfile вывести данные в указанный файл;
  - `-r`шаблон указать шаблон для поиска;
  - `-C` различать большие и малые буквы;
  - `-n` выдавать номера строк.

# Задание 1

- а затем ищет в указанном файле нужные строки, определяемые ключом -r. (рис. 1)



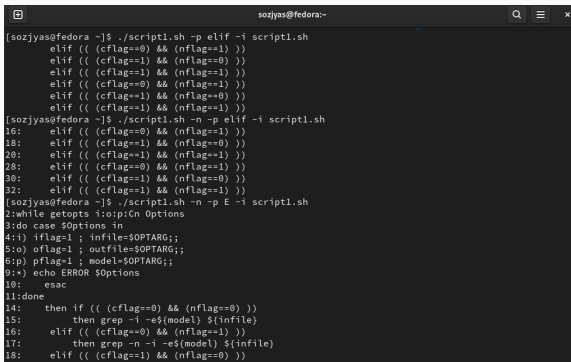
```
sozlyas@fedora:~ -- mcedit script1.sh
script1.sh [-----] 0 L: [ 5+31 36/ 37] *(1008/1009b) 0010 0x00A (*)
o) oflag=1 ; outfile=$OPTARG;;
p) pflag=1 ; model=$OPTARG;;
c) cflag=1 ;;
n) nflag=1 ;;
*) echo ERROR $Options
   esac
done

if (( (iflag==1) && (pflag==1) && (oflag==0) ))
then if (( (cflag==0) && (nflag==0) ))
-----> then grep -i -e${model} ${infile}
-----> elif (( (cflag==0) && (nflag==1) ))
-----> then grep -n -i -e${model} ${infile}
-----> elif (( (cflag==1) && (nflag==0) ))
-----> then grep -i -e${model} ${infile}
-----> elif (( (cflag==1) && (nflag==1) ))
-----> then grep -n -e${model} ${infile}
-----> fi
fi

if (( (iflag==1) && (pflag==1) && (oflag==1) ))
then if (( (cflag==0) && (nflag==0) ))
-----> then grep -i -e${model} ${infile} > ${outfile}
-----> elif (( (cflag==0) && (nflag==1) ))
-----> then grep -n -i -e${model} ${infile} > ${outfile}
-----> elif (( (cflag==1) && (nflag==0) ))
-----> then grep -i -e${model} ${infile} > ${outfile}
-----> elif (( (cflag==1) && (nflag==1) ))
-----> then grep -n -e${model} ${infile} > ${outfile}
-----> fi
fi
```

Figure 1: Командный файл №1

- Выполнил его. (рис. 2)



```
sozjyas@fedora:~  
[sozjyas@fedora ~]$ ./script1.sh -p elif -i script1.sh  
    elif (( (cflag==0) && (nflag==1) ))  
    elif (( (cflag==1) && (nflag==0) ))  
    elif (( (cflag==1) && (nflag==1) ))  
    elif (( (cflag==0) && (nflag==1) ))  
    elif (( (cflag==1) && (nflag==0) ))  
    elif (( (cflag==1) && (nflag==1) ))  
[sozjyas@fedora ~]$ ./script1.sh -n -p elif -i script1.sh  
16:    elif (( (cflag==0) && (nflag==1) ))  
18:    elif (( (cflag==1) && (nflag==0) ))  
20:    elif (( (cflag==1) && (nflag==1) ))  
28:    elif (( (cflag==0) && (nflag==1) ))  
30:    elif (( (cflag==1) && (nflag==0) ))  
32:    elif (( (cflag==1) && (nflag==1) ))  
[sozjyas@fedora ~]$ ./script1.sh -n -p E -i script1.sh  
2:while getopts i:o:p:Cn Options  
3:do case $Options in  
4:i) iflag=1 ; infile=$OPTARG;;  
5:o) oflag=1 ; outfile=$OPTARG;;  
6:p) pflag=1 ; model=$OPTARG;;  
9:*) echo ERROR $Options  
10:    esac  
11:done  
14:    then if (( (cflag==0) && (nflag==0) ))  
15:        then grep -i -e${model} ${infile}  
16:    elif (( (cflag==0) && (nflag==1) ))  
17:        then grep -n -i -e${model} ${infile}  
18:    elif (( (cflag==1) && (nflag==0) ))
```

Figure 2: Выполнение командного файла №1

- Часть 2 (рис. 3)

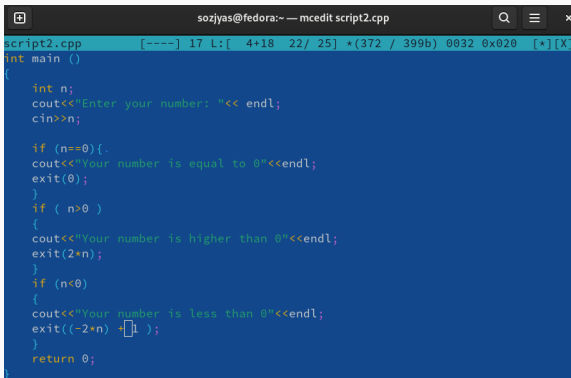
```
sozjyas@fedora:~  
[sozjyas@fedora ~]$ ./script1.sh -Cn -p E -i script1.sh  
9:*) echo ERROR $Options  
[sozjyas@fedora ~]$ ./script1.sh -p elif -i script1.sh -o out1.txt  
[sozjyas@fedora ~]$ ls  
avatar.jpg  Documents      'install-tl-unx (1)'  
backup     Images        install-tl-unx.tar.gz  
bin        _index.md     logfile  
Bureau    install-tl-unx  Modèles  
morefun   out1.txt      Téléchargements  
Musique  Pictures      Vidéos  
newdir   Public        work  
newdir1  script1.sh   z  
[sozjyas@fedora ~]$ cat out1.txt  
elif (( (cflag==0) && (nflag==1) ))  
elif (( (cflag==1) && (nflag==0) ))  
elif (( (cflag==1) && (nflag==1) ))  
elif (( (cflag==0) && (nflag==1) ))  
elif (( (cflag==1) && (nflag==0) ))  
elif (( (cflag==1) && (nflag==1) ))  
[sozjyas@fedora ~]$ ./script1.sh -n -p elif -i script1.sh -o out2.txt  
[sozjyas@fedora ~]$ cat out2.txt  
16: elif (( (cflag==0) && (nflag==1) ))  
18: elif (( (cflag==1) && (nflag==0) ))  
20: elif (( (cflag==1) && (nflag==1) ))  
28: elif (( (cflag==0) && (nflag==1) ))  
30: elif (( (cflag==1) && (nflag==0) ))  
32: elif (( (cflag==1) && (nflag==1) ))  
[sozjyas@fedora ~]$ ./script1.sh -n -p E -i script1.sh -o out2.txt  
[sozjyas@fedora ~]$ ./script1.sh -n -p E -i script1.sh -o out3.txt  
[sozjyas@fedora ~]$ cat out3.txt  
2:while getopts i:o:p:Cn Options  
3:do case $Options in  
4:i) iflag=1 ; infile=$OPTARG;;  
5:o) oflag=1 ; outfile=$OPTARG;;  
6:p) pflag=1 ; model=$OPTARG;;  
9:*) echo ERROR $Options
```

Figure 3: Выполнение командного файла №1



## Задание 2

- 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 о коде завершения в оболочку. (рис. 4)



```
sozjyas@fedora:~ — mcedit script2.cpp
script2.cpp [----] 17 L:[ 4+18 22/ 25] *(372 / 399b) 0032 0x020 [*][X]
int main ()
{
    int n;
    cout<<"Enter your number: "<< endl;
    cin>>n;

    if (n==0){
        cout<<"Your number is equal to 0"<<endl;
        exit(0);
    }
    if ( n>0 )
    {
        cout<<"Your number is higher than 0"<<endl;
        exit(2*n);
    }
    if (n<0)
    {
        cout<<"Your number is less than 0"<<endl;
        exit((-2*n) + 1);
    }
    return 0;
}
```

Figure 4: Программа на языке Си

- Написал командный файл, который вызывает эту программу и, проанализировав с помощью команды \$?, выдает сообщение о том, какое число было введено. (рис. 5)

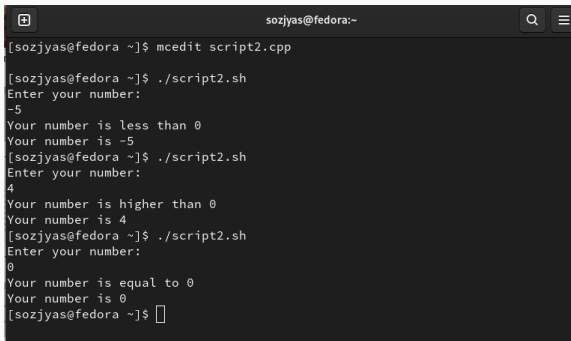


```
sozjyas@fedora:~ — mcedit script2.sh
script2.sh [----] 40 L:[ 1+14 15/ 16] *(235 / 238b) 0010 0x00A [*][X]
~/bin/bash
g++ -o cpp script2.cpp
./cpp
a=?
b=$a%2

if (( $a == 0 ))
then echo Your number is 0
elif (( b==0 ))
then echo Your number is "$(( $a / 2 ))"
else
let "a=$a-1"
b=-2
echo Your number is "$(( $a / $b ))"
fi
```

Figure 5: Командный файл №2

- Выполнил командный файл (рис. 6)

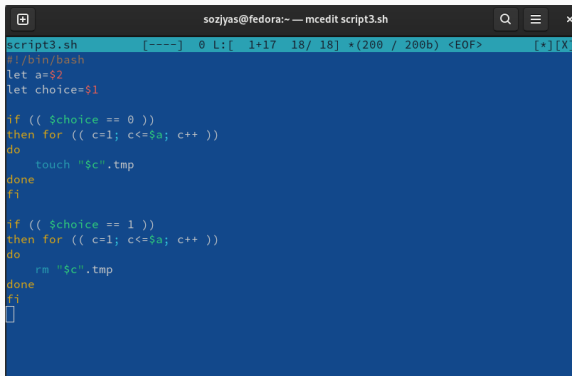


```
sozjyas@fedora:~  
[sozjyas@fedora ~]$ mcedit script2.cpp  
[sozjyas@fedora ~]$ ./script2.sh  
Enter your number:  
-5  
Your number is less than 0  
Your number is -5  
[sozjyas@fedora ~]$ ./script2.sh  
Enter your number:  
4  
Your number is higher than 0  
Your number is 4  
[sozjyas@fedora ~]$ ./script2.sh  
Enter your number:  
0  
Your number is equal to 0  
Your number is 0  
[sozjyas@fedora ~]$
```

Figure 6: Выполнение командного файла №2

- Написал командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до `%d` (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 умеет удалять все созданные им файлы (если они существуют). (рис. 7)



```
sozjyas@fedora:~ — mcedit script3.sh
script3.sh  [----]  0 L: [ 1+17 18/ 18] *(200 / 200b) <EOF>  [*] [X]
#!/bin/bash
let a=$2
let choice=$1

if (( $choice == 0 ))
then for (( c=1; c<=$a; c++ ))
do
    touch "$c".tmp
done
fi

if (( $choice == 1 ))
then for (( c=1; c<=$a; c++ ))
do
    rm "$c".tmp
done
fi
```

Figure 7: Командный файл №3

- Выполнил командный файл (рис. 8)

```
[sozjyas@fedora ~]$ ./script3.sh 0 5
[sozjyas@fedora ~]$ ls
1.tmp      Bureau      logfile     out2.txt    script3.sh
2.tmp      cpp         Modèles    out3.txt    scripto.cpp
3.tmp      Documents  morefun    out4.txt    script..sh
4.tmp      Images     Musique    Pictures    Téléchargements
5.tmp      _index.md  newdir     Public      Vidéos
avatar.jpg install-tl-unx newdir1     script1.sh  work
backup     'install-tl-unx (1)' ocpp        script2.cpp z
bin        install-tl-unx.tar.gz out1.txt    script2.sh

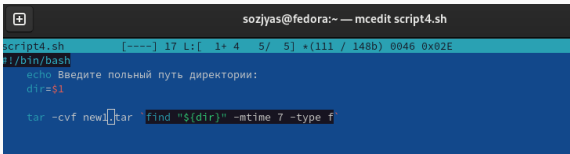
[sozjyas@fedora ~]$ ./script3.sh 1 5
[sozjyas@fedora ~]$ ls
avatar.jpg _index.md  Musique    out4.txt    scripto.cpp
backup     install-tl-unx newdir     Pictures    script..sh
bin        'install-tl-unx (1)' newdir1    Public      Téléchargements
Bureau    install-tl-unx.tar.gz ocpp        script1.sh  Vidéos
cpp        logfile     out1.txt   script2.cpp  work
Documents Modèles    out2.txt   script2.sh   z
Images    morefun    out3.txt   script3.sh

[sozjyas@fedora ~]$
```

Figure 8: Выполнение командного файла №2

## Задание 4

- Написал командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find). (рис. 9)



```
sozjyas@fedora:~ — mcedit script4.sh
script4.sh  [----] 17 L:[ 1+ 4 5/ 5] *(111 / 148b) 0046 0x02E
#!/bin/bash
echo Введите полный путь директории:
dir=$1
tar -cvf new1 tar `find "$dir" -mtime 7 -type f`
```

Figure 9: Командный файл №4

- Выполнил командный файл (рис. 10)

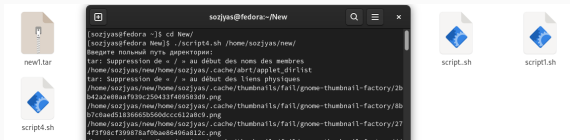


Figure 10: Выполнение командного файла №2



## Выводы

---

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

Wer's nicht glaubt, bezahlt einen Taler