

# Laboratory №12

---

Krupennikova V.

MAY-2021

RUDN University, Moscow, Russian Federation

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

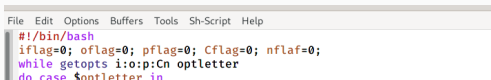
1. Ознакомиться с теоретическим материалом.
2. Изучить основы программирования в оболочке ОС UNIX/Linux.
3. Выполнить упражнения.
4. Ответить на контрольные вопросы.

## Выполнение:

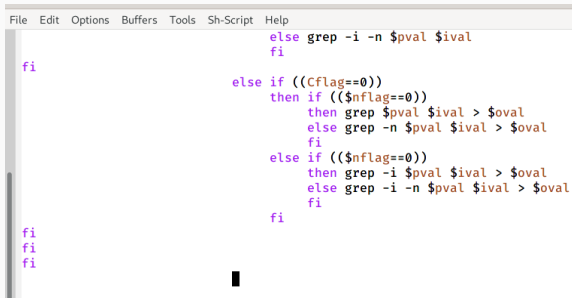
1)Используя команды getopts и grep, написала командный файл, который анализирует командную строку с ключами:

- i inputfile — прочитать данные из указанного файла;
- o outputfile — вывести данные в указанный файл;
- р шаблон — указать шаблон для поиска;
- С — различать большие и малые буквы;
- п — выдавать номера строк, а затем ищет в указанном файле нужные строки, определяемые ключом -р.

Для данной задачи я создала файл prog1.sh и написала соответствующие скрипты.



```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
iflag=0; oflag=0; pflag=0; Cflag=0; nflaf=0;
while getopts i:o:p:Cn optletter
do case $optletter in
```



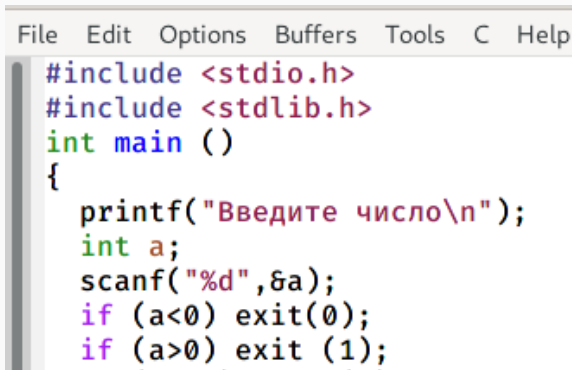
```
File Edit Options Buffers Tools Sh-Script Help
else grep -i -n $pval $ival
fi
fi
else if (($cflag==0))
then if (($nflag==0))
then grep $pval $ival > $oval
else grep -n $pval $ival > $oval
fi
else if (($nflag==0))
then grep -i $pval $ival > $oval
else grep -i -n $pval $ival > $oval
fi
fi
fi
fi
fi
```

Figure 2: Продолжение скрипта

Далее я проверила работу написанного скрипта, используя различные опции (например, команда «./prog.sh-la1.txt-oa2.txt-pcapital-C-n»), предварительно добавив право на исполнение файла (команда «chmod+хprog1.sh») и создав 2 файла, которые необходимы для выполнения программы: a1.txt и a2.txt

```
Файл  Правка  Вид  Закладки  Настройка  Справка
vakrupennikova@dk8n75 ~$ touch prog1.sh
vakrupennikova@dk8n75 ~$ emacs &
[1] 3656
vakrupennikova@dk8n75 ~$ touch a1.txt a2.txt
vakrupennikova@dk8n75 ~$ chmod +x prog1.sh
vakrupennikova@dk8n75 ~$ cat a1.txt
vakrupennikova@dk8n75 ~$ cat a1.txt
Hello , WORLD!
vakrupennikova@dk8n75 ~$ ./prog1.sh -i a1.txt -o a2.txt -p hello -C -n
./prog1.sh: строка 20: синтаксическая ошибка рядом с неожиданным маркером «)»
./prog1.sh: строка 20: `          then if ($Cflag==0))'
vakrupennikova@dk8n75 ~$ ./prog1.sh -i a1.txt -o a2.txt -p hello -C -n
vakrupennikova@dk8n75 ~$ cat a2.txt
vakrupennikova@dk8n75 ~$ chmod +x prog1.sh
vakrupennikova@dk8n75 ~$ ./prog1.sh -i a1.txt -o a2.txt -p hello -C -n
vakrupennikova@dk8n75 ~$ cat a2.txt
vakrupennikova@dk8n75 ~$ cat a1.txt
Hello , WORLD!
no
cp
vakrupennikova@dk8n75 ~$ cat a2.txt
vakrupennikova@dk8n75 ~$ ./prog1.sh -i a1.txt -o a2.txt -p hello -C -n
vakrupennikova@dk8n75 ~$ cat a2.txt
vakrupennikova@dk8n75 ~$ cat a2.txt
vakrupennikova@dk8n75 ~$ ./prog1.sh -i a1.txt -o a2.txt -p hello -C -n
vakrupennikova@dk8n75 ~$ ./prog1.sh -i a1.txt -o a2.txt -p Hello -C -n
vakrupennikova@dk8n75 ~$ cat a2.txt
1:Hello , WORLD!
vakrupennikova@dk8n75 ~$ touch chislo.c
vakrupennikova@dk8n75 ~$ touch chislo.sh
vakrupennikova@dk8n75 ~$ emacs &
[2] 9748
vakrupennikova@dk8n75 ~$ emacs &
[3] 10328
[1] Завершён      emacs
vakrupennikova@dk8n75 ~$ ./prog1.sh -i a1.txt -o a2.txt -p Hello -n
```

- 2) Написала 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. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено. Для данной задачи я создала 2 файла: `chislo.c` и `chislo.sh` и написала соответствующие скрипты.

A screenshot of a text editor window showing a C program. The editor has a menu bar with 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'C', and 'Help'. The code is as follows:

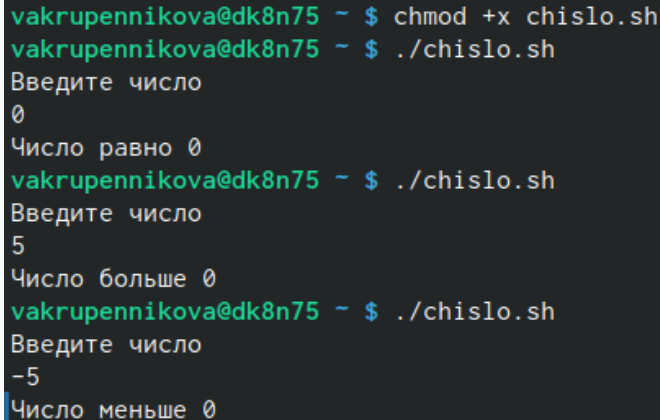
```
#include <stdio.h>
#include <stdlib.h>
int main ()
{
    printf("Введите число\n");
    int a;
    scanf("%d",&a);
    if (a<0) exit(0);
    if (a>0) exit (1);
```

```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
gcc chislo.c -o chislo
./chislo
code=$?
case $code in
    0) echo "Число меньше 0" ;;
    1) echo "Число больше 0" ;;
    2) echo "Число равно 0" ;;
esac
```

Figure 5: Третий скрипт



Далее я проверила работу написанных скриптов (команда «./chislo.sh»), предварительно добавив право на исполнение файла (команда «chmod+x chislo.sh»)



```
vakrupennikova@dk8n75 ~ $ chmod +x chislo.sh
vakrupennikova@dk8n75 ~ $ ./chislo.sh
Введите число
0
Число равно 0
vakrupennikova@dk8n75 ~ $ ./chislo.sh
Введите число
5
Число больше 0
vakrupennikova@dk8n75 ~ $ ./chislo.sh
Введите число
-5
Число меньше 0
```

Figure 6: Проверка скрипта

- 3) Написала 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). Для данной задачи я создала файл: files.sh и написала соответствующий скрипт.

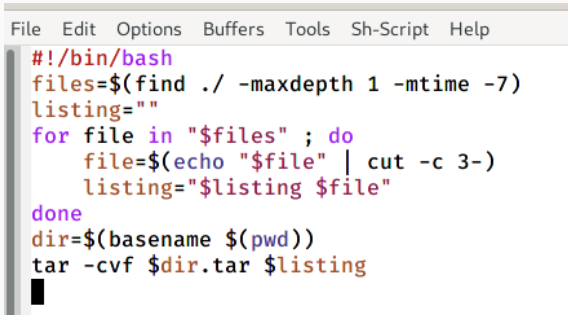
```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
opt=$1;
format=$2;
number=$3;
function Files()
{
    for (( i=1; i<=$number; i++ )) do
        file=$(echo $format | tr '#' "$i")
        if [ $opt == "-r" ]
        then
            rm -f $file
        elif [ $opt=="-C" ]
        then
            touch $file
        fi
    done
}
```

Далее я проверила работу написанного скрипта (команда «./files.sh»), предварительно добавив право на исполнение файла (команда «chmod+x files.sh»). Сначала я создала три файла (команда «./files.sh-cabc#.txt3»), удовлетворяющие условию задачи, а потом удалила их (команда «./files.sh-rabc#.txt3»)

```
root@perpetrator:~/h1# chmod +x files.sh
[22] Завершено
[23] Завершено
root@perpetrator:~/h1# ls
1.png 3.png 6.png 62.txt chislo.c chislo.sh files.sh lab05 public Ведро Изображения проекты
2021-05-25 12:31:36.msv 4.png 7.png backup chislo.c docs GRNetep progr1.sh public_html Загрузка Музыка Общедоступные 'Рабочий стол'
2.png 5.png 61.txt chislo.sh files.sh image progr1.sh tmp
root@perpetrator:~/h1# ./files.sh -C abc#.txt 3
./files.sh: строка 12: [: отсуствет common x]:
./files.sh: строка 12: [: отсуствет common x]:
root@perpetrator:~/h1# ./files.sh -C abc#.txt 3
bash: ./files.sh: Нет такого файла или каталога
root@perpetrator:~/h1# ./files.sh -C abc#.txt 3
root@perpetrator:~/h1# ls
1.png 3.png 6.png 62.txt 601.txt chislo.c chislo.sh files.sh lab05 public Ведро Изображения проекты
2021-05-25 12:31:36.msv 4.png 7.png backup chislo.c docs GRNetep progr1.sh public_html Загрузка Музыка Общедоступные 'Рабочий стол'
2.png 5.png 61.txt 600.txt chislo.sh files.sh image progr1.sh tmp
root@perpetrator:~/h1# ./files.sh -r abc#.txt 3
root@perpetrator:~/h1# ls
1.png 3.png 6.png 62.txt chislo.c chislo.sh files.sh lab05 public Ведро Изображения проекты
2021-05-25 12:31:36.msv 4.png 7.png backup chislo.c docs GRNetep progr1.sh public_html Загрузка Музыка Общедоступные 'Рабочий стол'
2.png 5.png 61.txt chislo.sh files.sh image progr1.sh tmp
```

Figure 8: Проверка скрипта

- 4) Написала командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировала его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find). Для данной задачи я создала файл: prog4.sh и написала соответствующий скрипт.



```
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
```

Figure 9: Пятый скрипт

Далее я проверила работу написанного скрипта (команды «sudo~/prog4.sh» и «tar-tfCatalog1.tar»), предварительно добавив право на исполнение файла (команда «chmod+хprog4.sh»)и создав отдельный catalog1 с несколькими файлами. Файлы ,измененные более недели назад, заархивированы не были.

```
vakrupennikova@dk8n75 ~ $ chmod +x prog4.sh
[4]- Завершён      emacs
vakrupennikova@dk8n75 ~ $ cd catalog
bash: cd: catalog: Нет такого файла или каталога
vakrupennikova@dk8n75 ~ $ cd catalog1
vakrupennikova@dk8n75 ~/catalog1 $ ls -l
итого 15
-rw-r--r-- 1 vakrupennikova studsci   21 мая 29 12:57 a1.txt
-rw-r--r-- 1 vakrupennikova studsci   17 мая 29 13:12 a2.txt
-rwxr-xr-x 1 vakrupennikova studsci 8072 мая 29 13:15 chislo
-rw-r--r-- 1 vakrupennikova studsci  198 мая 29 13:05 chislo.c
-rw-r--r-- 1 vakrupennikova studsci    0 мая 29 13:00 chislo.c~
-rwxr-xr-x 1 vakrupennikova studsci  193 мая 29 13:10 chislo.sh
-rw-r--r-- 1 vakrupennikova studsci    0 мая 29 13:00 chislo.sh~
-rwxr-xr-x 1 vakrupennikova studsci  244 мая 29 13:23 files.sh
-rw-r--r-- 1 vakrupennikova studsci    0 мая 29 13:16 files.sh~
-rwxr-xr-x 1 vakrupennikova studsci  990 мая 29 12:54 prog1.sh
-rw-r--r-- 1 vakrupennikova studsci    0 мая 29 12:31 prog1.sh~
-rwxr-xr-x 1 vakrupennikova studsci  209 мая 29 13:30 prog4.sh
-rw-r--r-- 1 vakrupennikova studsci    0 мая 29 13:25 prog4.sh~
```

Figure 10: Проверка скрипта

Скрипт работает корректно.

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

The end.