

Laboratory №12

Bezruk M.A.

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.

1)Используя команды `getopts``grep`, написалакомандный 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,а затем ищет в указанном файле нужные строки, определяемые ключом -р.

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

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.txtи a2.txt

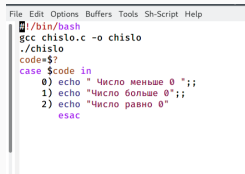


```
mbuzrukh@ubuntu:~$ touch prog1.sh
mbuzrukh@ubuntu:~$ emacs &
[1] 3128
mbuzrukh@ubuntu:~$ touch a1.txt a2.txt
mbuzrukh@ubuntu:~$ emacs &
[2] 4546
mbuzrukh@ubuntu:~$ cat a1.txt
Moscow is the capital of Russia
printf %s\n
sleep 5
cat a1.txt
```

Figure 1: Проверка скрипта

Скрипт работает корректно.

- 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.



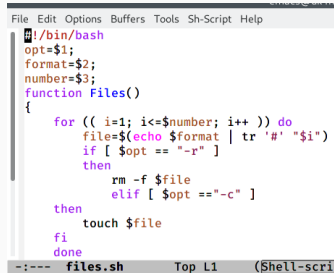
```
File Edit Options Buffers Tools Sh-Script Help
./bin/bash
gcc chislo.c -o chislo
./chislo
code=?
case $code in
0) echo " Число меньше 0 ";;
1) echo "Число больше 0" ;;
2) echo "Число равно 0"
esac
```

Figure 2: Третий скрипт

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

Скрипты работают корректно.

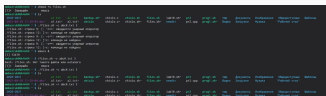
- 3) Написала командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N (например 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют). Для данной задачи я создала файл: files.sh и написала соответствующий скрипт.



```
#!/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
}
```

Figure 3: Четвертый скрипт

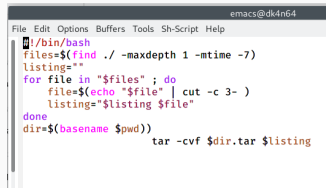
Далее я проверила работу написанного скрипта (команда «./files.sh»), предварительно добавив право на исполнение файла (команда «chmod+x files.sh»). Сначала я создала три файла (команда «./files.sh-cabc#.txt3»), удовлетворяющие условию задачи, а потом удалила их (команда «./files.sh-rabc#.txt3»)



```
root@kali:~/Desktop# ./files.sh
./files.sh: line 1: ./files.sh: Permission denied
root@kali:~/Desktop# chmod+x ./files.sh
root@kali:~/Desktop# ./files.sh-cabc#.txt3
./files.sh: line 1: ./files.sh: Permission denied
root@kali:~/Desktop# ./files.sh-rabc#.txt3
./files.sh: line 1: ./files.sh: Permission denied
root@kali:~/Desktop# ls -la
total 12
drwxr-xr-x 2 root root 4096 Nov 10 10:10 .
drwxr-xr-x 3 root root 4096 Nov 10 10:10 ..
-rwxr-xr-x 1 root root  120 Nov 10 10:10 files.sh
-rw-r--r-- 1 root root    0 Nov 10 10:10 abc.txt
-rw-r--r-- 1 root root    0 Nov 10 10:10 abc.txt
-rw-r--r-- 1 root root    0 Nov 10 10:10 abc.txt
```

Figure 4: 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.



```
emacs@dk4n64
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 5: Пятый скрипт

Далее я проверила работу написанного скрипта (команды «sudo~/prog4.sh» и «tar -tfCatalog1.tar»), предварительно добавив право на исполнение файла (команда «chmod+xprog4.sh»)и создав отдельный catalog1 с несколькими файлами. Файлы ,измененные более недели назад, заархивированы не были.

```
mbdzrsk@bde4: ~/Catalog1 $ ls -l
total 108844
-rwxr-xr-x 1 mbdzrsk studsci 111149056 mar 29 12:58 '2021-03-29 11:29-04.mw'
-rwxr-xr-x 1 mbdzrsk studsci 260 mar 29 12:31 chialo.sh
-rwxr-xr-x 1 mbdzrsk studsci 256 mar 29 12:42 files.sh
-rwxr-xr-x 1 mbdzrsk studsci 224 mar 29 12:51 prog4.sh
-rwxr-xr-x 1 mbdzrsk studsci 768 mar 29 12:09 prog1.sh
mbdzrsk@bde4: ~/Catalog1 $ tar -tf Catalog1.tar
```

Figure 6: 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.