

Презентация по лабораторной работе 11

Елизавета Александровна Гайдамака

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

1. 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;
 - `-р`шаблон — 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`.

2. Написать 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.

3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до [?] (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).
4. Написать 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).

1. Каково предназначение команды `getopts`?

Весьма необходимой при программировании является команда `getopts`, которая осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных.

2. Какое отношение метасимволы имеют к генерации имён файлов?

Метасимволом называется особая комбинация символов, которые в реальной строке могут совпадать с разными символами. Кроме того, некоторые метасимволы совпадают с границами между символами (например с началом строки). «.» {точка} (любой символ). Точка означает любой символ. Следует учесть, что символ всё-таки должен присутствовать, например RE `/x.z/` будет найдено в строке «xyz», «x-z», «abcxRzdef», но не в строке «xz». «[abc]» (символьный класс). на этом месте может быть только «a», «b», или «c». «[0-9]» (символьный диапазон). Под диапазон попадают все символы диапазона, здесь 0,1,2,3,4,5,6,7,8,9

3. Какие операторы управления действиями вы знаете?

Ветвления, выбора, цикла.

4. Какие операторы используются для прерывания цикла?

break

5. Для чего нужны команды false и true?

false - 0 true - 1

6. Что означает строка `if test -f mans/i.$$`, встречающаяся в командном файле?

Это условие оператора ветвления, если оно выполняется, то программа выполнит определенные команды, если нет - то нет.

7. Объясните различия между конструкциями `while` и `until`.

`while` – выполняет действие до тех пор, пока условие является истинным; `until` – будет выполняться до тех пор, пока условие не станет истинным, т. е. пока оно `false`.

Выполнение лабораторной работы

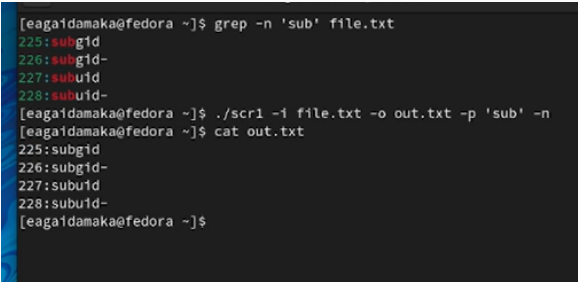
Код задания 1.

```
while getopts 1:o:p:cn optletter
do case $optletter in
    i) inp=${OPTARG};;
    o) outp=${OPTARG};;
    p) templ=${OPTARG};;
    c) reg=true;;
    n) num=true;;
esac
done

if $reg
then if $num
    then grep -i -n $templ $inp > $outp
    else grep -i $templ $inp > $outp
    fi
else if $num
    then grep -n $templ $inp > $outp
    else grep $templ $templ $inp > $outp
    fi
fi
```

Рис. 1: Рис.1

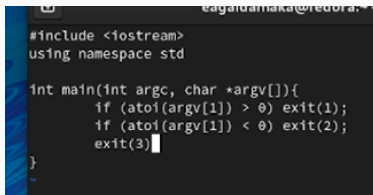
Результат работы.

A terminal window with a dark background and a blue vertical bar on the left. The text is white, with some output lines in red. The commands and their outputs are as follows:

```
[eagaidamaka@fedora ~]$ grep -n 'sub' file.txt
225:subgid
226:subgid-
227:subuid
228:subuid-
[eagaidamaka@fedora ~]$ ./scr1 -i file.txt -o out.txt -p 'sub' -n
[eagaidamaka@fedora ~]$ cat out.txt
225:subgid
226:subgid-
227:subuid
228:subuid-
[eagaidamaka@fedora ~]$
```

Рис. 2: Рис.2

Код файла сpp задания 2.

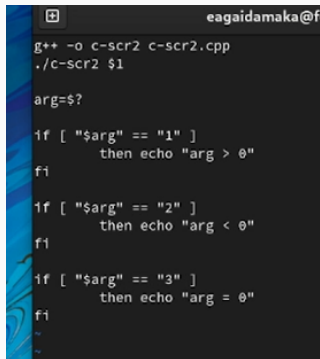
A screenshot of a terminal window with a dark background and light blue text. The terminal title bar at the top shows a user icon and the text 'eagandamaka@redora:'. The code displayed is a C++ program that includes the <iostream> header, uses the std namespace, and defines a main function. The main function takes argc and argv as arguments and uses atoi to convert argv[1] to an integer. It then checks if the integer is greater than 0 (calling exit(1)), less than 0 (calling exit(2)), or zero (calling exit(3)).

```
eagandamaka@redora:~$ cat task2.cpp
#include <iostream>
using namespace std

int main(int argc, char *argv[]){
    if (atoi(argv[1]) > 0) exit(1);
    if (atoi(argv[1]) < 0) exit(2);
    exit(3);
}
```

Рис. 3: Рис.3

Код файла bash задания 2.

A screenshot of a terminal window with a dark background and a blue and white patterned sidebar on the left. The terminal title bar shows a window icon and the username 'eagaidamaka@f'. The terminal content shows the compilation of a C++ file 'c-scr2.cpp' into an executable 'c-scr2', followed by the execution of the script with argument '1'. The script sets 'arg' to the first argument and contains three conditional checks for '1', '2', and '3', each printing a message if the condition is met.

```
eagaidamaka@f
g++ -o c-scr2 c-scr2.cpp
./c-scr2 1

arg=?

if [ "$arg" == "1" ]
then echo "arg > 0"
fi

if [ "$arg" == "2" ]
then echo "arg < 0"
fi

if [ "$arg" == "3" ]
then echo "arg = 0"
fi

~
~
```

Рис. 4: Рис.4

Результат работы.

```
[eagaidamaka@fedora ~]$ ./scr2 -3  
arg < 0  
[eagaidamaka@fedora ~]$ ./scr2 0  
arg = 0  
[eagaidamaka@fedora ~]$ ./scr2 3  
arg > 0
```

Рис. 5: Рис.5

Код задания 3.

```
v=false
r=false
while getopts c:r optletter
do case $optletter in
    c) c="$OPTARG"; v=true;;
    r) r=true;;
esac
done

if $v
then for i in $(seq 1 $c)
    do touch "$i.tmp"
    done
fi

if $r
then for i in $(ls | find -name "*.tmp")
    do rm $i
    done
fi
```

Рис. 6: Рис.6

Результат работы.

```
[eagaidamaka@fedora ~]$ ./scr3 -c 4
[eagaidamaka@fedora ~]$ ls
1.tmp                c-scr2.cpp          scr3                Изображения
2.tmp                file.txt             test                Музыка
3.tmp                outp.txt             work                Общедоступные
4.tmp                out.txt             Видео              'Рабочий стол'
course-directory-student-template  scr1                Документы          Шаблоны
c-scr2              scr2                Загрузки
[eagaidamaka@fedora ~]$ ./scr3 -r
[eagaidamaka@fedora ~]$ ls
course-directory-student-template  out.txt            work                Музыка
c-scr2                             scr1               Видео              Общедоступные
c-scr2.cpp                         scr2               Документы          'Рабочий стол'
file.txt                           scr3               Загрузки           Шаблоны
outp.txt                           test               Изображения
[eagaidamaka@fedora ~]$
```

Рис. 7: Рис.7

Код задания 4

```
find $1 -type f -mtime -7 > ttar.txt  
tar -cvf arch.tar -T ttar.txt
```

Рис. 8: Рис.8

Результат работы.

```
[eagaidamaka@fedora ~]$ mv arch.tar sdf
[eagaidamaka@fedora ~]$ cd sdf
[eagaidamaka@fedora sdf]$ tar -xf arch.tar
[eagaidamaka@fedora sdf]$ cd home/
[eagaidamaka@fedora home]$ cd eagaidamaka/
[eagaidamaka@fedora eagaidamaka]$ ls
c-scr2      file.txt  out.txt  scr2  scr4      ttar.txt
c-scr2.cpp outp.txt  scr1    scr3  test.tar
[eagaidamaka@fedora eagaidamaka]$
```

Рис. 9: Рис.9

Благодаря данной работе я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.