Федеральное агентство связи

Сибирский Государственный Университет Телекоммуникаций и Информатики

## Кафедра ТС и ВС

**Лабораторная работа № 7**

**По дисциплине: Операционные системы**

**Выполнил**: Зарубин Максим Евгеньевич

**Группа**: ИА-831

## Вариант: 5

**Проверила**: Моренкова Ольга Ильинична

Новосибирск, 2020 г

**Взаимодействие программ с командной оболочкой. Обработка исключительных ситуаций.**

**Задание на лабораторную работу.**

Написать программу на языке Си, состоящую из двух файлов: заголовочного файла и файла программы.

В заголовочном файле должны быть подключены необходимые системные заголовочные файлы, и описаны все глобальные переменные и функции, используемые в файле программы. Программа должна выполнять следующие действия:

1) вывести первые 10 переменных окружения на экран двумя различными способами (способ задаётся пользователем в командной строке опциями -1 и -2);

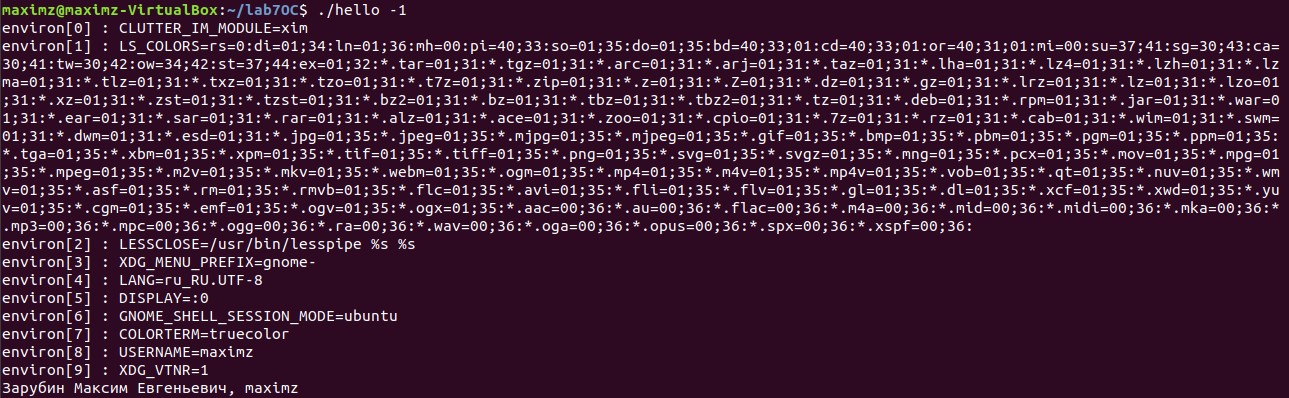
2) обработать все указанные опции командной строки и в случае ошибочных опций выдать сообщение в стандартный поток ошибок о неправильном использовании программы;

3) в командной строке опцией -f файл указывается имя файла, который необходимо открыть и вывести его содержимое на экран. Также необходимо обработать ошибки, которые могут возникнуть при открытии, закрытии и выводе файла на экран;

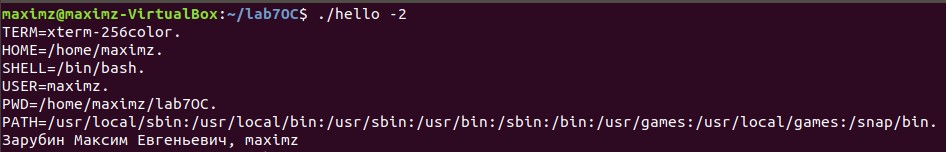
4) при завершении программы (даже в случае возникновения ошибок) на экран должно быть выдано сообщение об авторе программы: ФИО, идентификатор пользователя.

**Результаты работы программы.**

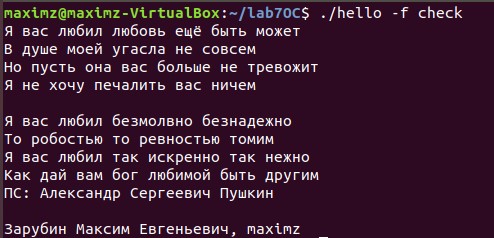
**Опция -1:**



**Опция -2:**



**Опция –f:**



**Листинг программы.**

**Main.cpp:**

#include "main.h"

void p1\_1()

{

for (int i = 0; i < 10; i ++)

if (environ[i] != NULL)

printf("environ[%d] : %s\n",i, environ[i]);

}

void p1\_2()

{

char \*term, \*home, \*shell, \*user, \*pwd, \*path;

term = getenv("TERM");

home = getenv("HOME");

shell = getenv("SHELL");

user = getenv("USER");

pwd = getenv("PWD");

path = getenv("PATH");

printf("TERM=%s.\nHOME=%s.\nSHELL=%s.\nUSER=%s.\nPWD=%s.\nPATH=%s.\n", term, home, shell, user, pwd, path);

}

int p3(const char\* filename){

int ch;

FILE \*file = fopen(filename, "r");

while (!feof(file) && !ferror(file) && file != NULL)

{

ch = getc(file);

if (ch != EOF)

{

putchar(ch);

}

}

putchar('\n');

return 0;

}

int main(int argc, char \*argv[])

{

int c;

while (1){

c = getopt(argc, argv, "f:12");

switch (c){

case 'f':{

p3(optarg);

break;

}

case '?':{

printf("Неверная опция");

break;

}

case '1':{

p1\_1();

break;

}

case '2':{

p1\_2();

break;

}

default:{

printf ("Произошла проблема\n");

break;

}

}

break;

}

printf ("Зарубин Максим Евгеньевич, maximz\n");

exit (0);

}

**Main.h:**

#ifndef MAIN\_H

#define MAIN\_H

#include <stddef.h>

#include <stdio.h>

#include <stdlib.h>

#include <getopt.h>

extern char \*\*environ;

void p1\_1();

void p1\_2();

int p3(const char\* filename);

#endif

**Контрольные вопросы.**

8) Передача окружения программы производить через глобальную переменную environ и по их именам: getenv и putenv.

9)Стандарт ANSI С определяет только два первых параметра функции main–argc и argv.

10)Специальный аргумент «--» (два тире) служит для обозначения конца опций независимо от режима работы функции getopt. Если getopt не распознал символ опции, то он выводит в стандартный поток ошибок соответствующее сообщение, заносит символ в переменную optopt и возвращает «?». Вызывающая программа может предотвратить вывод сообщения об ошибке, установив нулевое значение opterr. Если найден параметр, начинающийся с двух символов «-» (тире), то считается, что найдена длинная опция. В этом случае производится сравнение найденного аргумента (исключая двойное тире) со значениями name массива longopts. Если совпадение найдено, то значение val помещается либо по адресу, указанному в flag, и getopt\_long завершается с результатом 0, либо getopt\_long завершается с результатом val. В любом из двух случаев, если параметр long index не равен NULL, то по указанному в нём адресу помещается номер элемента массива, с которым совпала найденная длинная опция.  
Длинная опция может иметь параметр, указываемый через знак равенства (--опция=параметр) или через пробел (--опция параметр). Наличие параметра указывается в поле has\_arg. Функция getopt\_long\_only работает так же, как getopt\_long, но обрабатывает только длинные опции. В качестве указателя опции могут служить не только символы «--», но и «-». Если опция, начинающаяся с «-» (не с «--»), не совпадает с длинной опцией, но совпадает с короткой, то она обрабатывается как короткая опция.