Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №5 по курсу**

**«Операционные системы»**

**Тема работы**

**“Динамические библиотеки”**

Студент: Меджидли Махмуд Ибрагим оглы

Группа: М8О-208Б-20

Вариант: 31

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2021

**Содержание**

1. Репозиторий
2. Постановка задачи
3. Общие сведения о программе
4. Общий метод и алгоритм решения
5. Исходный код
6. Демонстрация работы программы
7. Выводы

**Репозиторий**

https://github.com/loshadkaigogo

**Постановка задачи**

Задача: реализовать 2 динамические библиотеки и 2 программы для работы с ними. Первая программа будет загружать библиотеку (одну) на этапе компиляции при помощи ключа -lmylib, а вторая программа будет подключать две динамические библиотеки при помощи dl-функций в самом коде.

**Общие сведения о программе**

Для выполнения данной лабораторной работы я предварительно создал 5 файлов: первые два - first.cpp и second.cpp являются исходным кодом для наших динамических библиотек. Файлы compilation.cpp и launch.cpp являются двумя программами, которые нужно было реализовать по заданию. compilation.cpp является программой, к которой библиотека подгружается на этапе компиляции, а launch.cpp является программой, к которой библиотека подключается непосредственно в самом коде.

Помимо этого, для удобства компиляции всех программ я создал MakeFile со следующим набором команд:  
1) g++ -fPIC -c first.cpp -o d1.o

g++ -fPIC -c second.cpp -o d2.o

При помощи этих команд наши cpp-библиотеки превращаются в объектные файлы. Это, так называемый, “промежуточный этап” создания динамических библиотек.

2) g++ -shared d1.o -o libd1.so

g++ -shared d2.o -o libd2.so

При помощи флага -shared мы создаем наши нужные по заданию динамические библиотеки.

3) g++ compilation.cpp -L. -ld1 -o main1 -Wl,-rpath -Wl,.

Этой строчкой мы делаем исполняемый файл из нашей программы compilation.cpp, при этом компилируем мы только с одной библиотекой (то есть компиляция может проходить либо с ключом -ld1, либо с ключом -ld2).

4) g++ launch.cpp -L. -ldl -o main2 -Wl,-rpath -Wl,.

Этой строчкой мы делаем исполняемый файл из нашей программы launch.cpp, только теперь с флагом -ldl. Далее в нашей программе main2 будут доступны 2 динамические библиотеки, действия над которыми будут обрабатываться при помощи следующих функций:  
void\* dlopen(...) - вгружает нашу библиотеку;

void\* dlsym(...) - присваивает указателю на функцию ее адрес в библиотеке

int dlclose(...) - освобождает указатель на библиотеку

5) rm -r \*.so \*.o main1 main2

При помощи команды make clean происходит удаление всех созданных файлов, вследствие чего в папке остаются исходные 5 объектов.

**Общий метод и алгоритм решения**

В самом начале выполнения лабораторной работы я реализовал две библиотеки: first.cpp и second.cpp. В библиотеке first.cpp реализовано вычисление экспоненты при помощи второго замечательного предела и реализован поиск площади прямоугольника. В библиотеке second.cpp реализовано вычисление экспоненты при помощи ряда Тейлора и реализован поиск прямоугольного треугольника. Далее в файле compilation.cpp я реализовал обычное считывание команды при помощи проверки равенства функции scanf на -1 (вводится EOF - Ctrl+D на Ubuntu) и конструкции switch-case. Если вводится команда, отличная от 1 или 2, вылезает сообщение о том, что ввод был осуществлен неправильно. Если вводится 1, то считается экспонента при помощи второго замечательного предела. Если вводится 2, то считается площадь прямоугольника.  
Что же касается launch.cpp, то там суть почти та же. В начале создаю необходимые указатели, позже загружаю какую-либо дин. библиотеку в зависимости от ввода пользователя. При помощи известного нам считывания до EOF я считываю команду. Если это не 0, не 1 и не 2, то прошу ввести правильную команду. Если эта команда 0, то программа меняет библиотеки (то есть, если раньше мне был доступен подсчет экспоненты вторым зам. пределом и площадь прямоугольника, то теперь мне будет доступен подсчет экспоненты посредством ряда Тейлора и площадь прямоугольника треугольника). Если команда 1, я считаю экспоненту. Если 2, то площадь. В конце освобождается указатель на библиотеку в целях избежания утечек памяти, программа завершается.

**Исходный код**

**first.cpp**

extern "C" float E (int x);

extern "C" float Square (float A, float B);

float E (int x) {

if (x < 0) {

return -1;

}

float e = 1.0;

for (int i = 0; i < x; ++i) {

e \*= 1 + 1/(float)x;

}

return e;

}

float Square (float A, float B) {

if (A <= 0 || B <= 0) {

return -1;

}

float square = A \* B;

return square;

}

**second.cpp**

extern "C" float E (int x);

extern "C" float Square (float A, float B);

float E (int x) {

if (x < 0) {

return -1;

}

float e = 1.0;

float term = 1.0;

for (int i = 1; i <= x; ++i) {

term /= i;

e += term;

}

return e;

}

float Square (float A, float B) {

if (A <= 0 || B <= 0) {

return -1;

}

float square = 0.5 \* A \* B;

return square;

}

**compilation.cpp**#include <iostream>

extern "C" float E (int x);

extern "C" float Square (float A, float B);

int main () {

int command;

while (printf("Please enter your command: ") && (scanf("%d", &command)) != EOF) {

switch (command) {

case 1: {

int x;

std:: cin >> x;

float exp = E(x);

if (exp == -1) {

std:: cout << "Please enter a positive accuracy!" << std:: endl;

}

else {

std:: cout << "An approximate exponent value is: " << exp << std:: endl;

}

break;

}

case 2: {

float A, B, square;

std:: cin >> A >> B;

square = Square(A, B);

if (square == -1) {

std:: cout << "Please enter positive variables!" << std:: endl;

}

else {

std:: cout << "The square of your figure is: " << square << std:: endl;

}

break;

}

default:

std:: cout << "You must enter 1 or 2!" << std:: endl;

break;

}

}

return 0;

}

**launch.cpp**#include <stdio.h>

#include <iostream>

#include <stdlib.h>

#include <dlfcn.h>

int main () {

void\* handle = NULL;

float (\*E)(int x);

float (\*Square)(float A, float B);

const char\* lib\_array[] = {"libd1.so", "libd2.so"};

int curlib;

int start\_library;

std:: cout << "Enter start library: " << std:: endl;

std:: cout << '\t' << "1 for using first library" << std:: endl;

std:: cout << '\t' << "2 for using second library" << std:: endl;

std:: cin >> start\_library;

bool flag = true;

while (flag) {

if (start\_library == 1) {

curlib = 0;

flag = false;

}

else if (start\_library == 2) {

curlib = 1;

flag = false;

}

else {

std:: cout << "You should enter only 1 or 2!" << std:: endl;

std:: cin >> start\_library;

}

}

handle = dlopen(lib\_array[curlib], RTLD\_LAZY);

if (!handle) {

std:: cout << "An error while opening library has been detected" << std:: endl;

exit(EXIT\_FAILURE);

}

E = (float(\*)(int))dlsym(handle, "E");

Square = (float(\*)(float, float))dlsym(handle, "Square");

int command;

std:: cout << "Hello there! Please enter your command according to next rules: " << std:: endl;

std:: cout << '\t' << "0 for changing the contract;" << std:: endl;

std:: cout << '\t' << "1 for calculating the exponent; " << std:: endl;

std:: cout << '\t' << "2 for calculating the square; " << std:: endl;

while (printf("Please enter your command: ") && (scanf("%d", &command)) != EOF) {

if (command == 0) {

dlclose(handle);

if (curlib == 0) {

curlib = 1 - curlib;

handle = dlopen(lib\_array[curlib], RTLD\_LAZY);

if (!handle) {

std:: cout << "An error while opening library has been detected" << std:: endl;

exit(EXIT\_FAILURE);

}

E = (float(\*)(int))dlsym(handle, "E");

Square = (float(\*)(float, float))dlsym(handle, "Square");

}

else if (curlib == 1) {

curlib = 1 - curlib;

handle = dlopen(lib\_array[curlib], RTLD\_LAZY);

if (!handle) {

std:: cout << "An error while opening library has been detected" << std:: endl;

exit(EXIT\_FAILURE);

}

E = (float(\*)(int))dlsym(handle, "E");

Square = (float(\*)(float, float))dlsym(handle, "Square");

}

std:: cout << "You have changed contracts!" << std:: endl;

}

else if (command == 1) {

int x;

std:: cout << "Please enter accuracy x: ";

std:: cin >> x;

float exp = E(x);

if (exp == -1) {

std:: cout << "Please enter a positive accuracy!" << std:: endl;

}

else {

std:: cout << "An approximate exponent value is: " << exp << std:: endl;

}

}

else if (command == 2) {

float A, B, square;

std:: cout << "Please enter sides of your figure: ";

std:: cin >> A >> B;

square = Square(A, B);

if (square == -1) {

std:: cout << "Please enter positive variables!" << std:: endl;

}

else {

std:: cout << "The square of your figure is: " << square << std:: endl;

}

}

else {

std:: cout << "You had to enter only 0, 1 or 2!" << std:: endl;

}

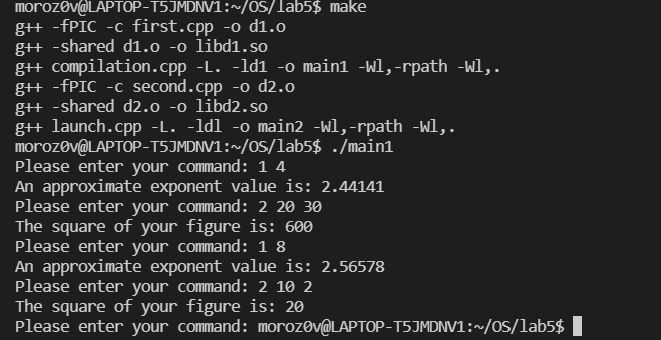
}

dlclose(handle);

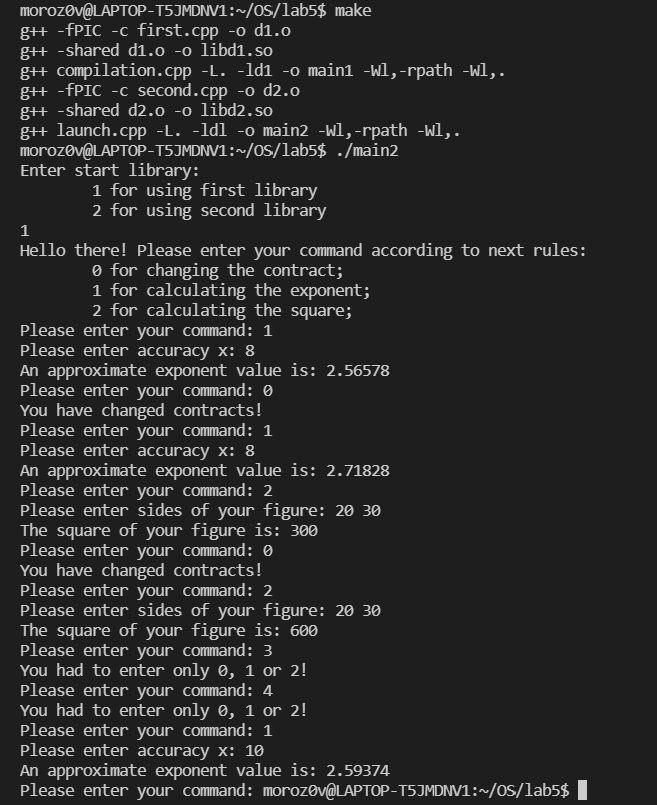
return 0;

}

**Демонстрация работы программы**

make ****./main1

**Text

Description automatically generated**make ****

./main2

**Text

Description automatically generated**

**Выводы**Данная лабораторная работа научила меня пользоваться dl-функциями, благодаря реализации исполняемых файлов по заданию, я закрепил навык работы с динамическими библиотеками и полностью осознал их отличие от статических библиотек.