Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №5 по курсу**

**«Операционные системы»**

Студент: Николаева Елизавета Сергеевна

Группа: М8О-201Б-20

Вариант: 1

Преподаватель: Миронов Евгений Сергеевич

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/esnikolaeva/OS\_5

**Постановка задачи**

Требуется создать динамические библиотеки, которые реализуют определенный функционал. Далее использовать данные библиотеки 2-мя способами:

1. Во время компиляции (на этапе «линковки»/linking)

2. Во время исполнения программы. Библиотеки загружаются в память с помощью интерфейса ОС для работы с динамическими библиотеками

В итоге, необходимо получить следующие части:

1. Динамические библиотеки, реализующие контракты, которые заданы вариантом

2. Тестовая программа (программа №1), которая используют одну из библиотек, используя знания полученные на этапе компиляции

3. Тестовая программа (программа №2), которая загружает библиотеки, используя только их местоположение и контракты

Провести анализ двух типов использования библиотек.

Пользовательский ввод для обоих программ должен быть организован следующим образом:

1. Если пользователь вводит команду «0», то программа переключает одну реализацию контрактов на другую (необходимо только для программы №2)

2. «1 arg1 arg2 … argN», где после «1» идут аргументы для первой функции, предусмотренной контрактами. После ввода команды происходит вызов первой функции, и на экране появляется результат её выполнения

3. «2 arg1 arg2 … argM», где после «2» идут аргументы для второй функции, предусмотренной контрактами. После ввода команды происходит вызов второй функции, и на экране появляется результат её выполнения

**Общие сведения о программе**

Программа написана на языке Си в UNIX-подобной операционной системе. Сборка проекта происходит при помощи make-файла.

**Общий метод и алгоритм решения**

В файлах libary1.c и libary2.c я реализовала функции методами, соответствующими условию задания. В тестовой программе №1, которая использует данные библиотеки во время компиляции, прописала цикл while, в условии которого идет чтение ключа. Пока ключ может считаться, в зависимости от его значения я выбираю функцию, которая будет обрабатываться и выводить результат в консоль (1 – SinIntegral, 2-Derivative). После выбора функции считываю входные данные для функции, вычисляю её и вывожу результат.

В тестовой программе №2 изначально прошу пользователя выбрать стартовую библиотеку. В зависимости от его выбора, загружаю открываю ту или иную библиотеку с помощью функции dlopen. Нахожу начало адреса функций SinIntegral и Derivative с помощью функции dlsym. В цикле while считываю ключ и в зависимости от его значения обрабатываю ту или иную функцию и вывожу результат на консоль. Если ключ равен 0, с помощью функции dclose закрываю библиотеку и, открываю и загружаю (dlopen) другую библиотеку. Также с помощью dlsym нахожу начало адреса функций SinIntegral и Derivative уже другой библиотеки. После окончания ввода, закрываю библиотеку.

**Исходный код**

**main1.cpp**

#include <iostream>

using namespace std;

//extern "C" для использования float в работе с С++

extern "C" float SinIntegral(float A, float B, float e);

extern "C" float Derivative(float A, float deltaX);

int main()

{

int flag;

cout << "Please enter the flag:\n";

while (scanf("%d",&flag) > 0 ){

if (flag == 1){

cout<<"Please enter your dates:\n";

float A, B, e;

cin>>A>>B>>e;

cout<<"SinIntegral("<< A<<","<<B<<","<< e<<")="<<SinIntegral(A, B, e)<< endl;

}

else if (flag == 2){

cout<<"Please enter your dates:\n";

float A, deltaX;

cin>>A>>deltaX;

cout<< "Derivative("<<A<<","<<deltaX<<")="<<Derivative(A,deltaX)<<endl;

}

}

return 0;

}

**main2.cpp**

#include <stdio.h>

#include <iostream>

#include <stdlib.h>

#include <dlfcn.h>

using namespace std;

int main()

{

void\* adres = NULL;//адресс для доступа к библиотеке

//указатели на функции 1 и 2

float (\*SinIntegral)(float A, float B, float e);

float (\*Derivative)(float A, float deltaX);

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

int curlib;

int statl;

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

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

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

cin>>statl;

bool flag = 1;

while(flag){

if (statl==1){

curlib = 0;

flag = 0;

}

else if(statl==2){

curlib=1;

flag=0;

}

else{

cout<<"You enter the wrong number. Enter again!"<<endl;

cin>>statl;

}

}

adres = dlopen(libeary\_mas[curlib],RTLD\_LAZY);//RTLD\_LAZY выполняется поиск только тех символов, на которые есть ссылки из кода

if (!adres){

cout<<"Error";

exit(EXIT\_FAILURE);

}

SinIntegral = (float(\*)(float, float, float))dlsym(adres,"SinIntegral");//возвращаем адрес функции из памяти библиотеки /dlsym присваивает указателю на функцию, объявленному в начале, ее адрес в библиотеке

Derivative = (float(\*)(float, float))dlsym(adres, "Derivative");

int command;

cout<<"Please read the nexst ruls? before you enter the command";

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

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

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

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

if (command==0){

dlclose(adres);//освобождает указатель на библиотеку и программа перестает ей пользоваться

if (curlib==0){

curlib=1;

adres=dlopen(libeary\_mas[curlib],RTLD\_LAZY);

if(!adres){

cout<<"Error";

exit(EXIT\_FAILURE);

}

SinIntegral = (float(\*)(float, float, float))dlsym(adres,"SinIntegral");//возвращаем адрес функции из памяти библиотеки /dlsym присваивает указателю на функцию, объявленному в начале, ее адрес в библиотеке

Derivative = (float(\*)(float, float))dlsym(adres, "Derivative");

}

else if(curlib==1){

curlib=0;

adres = dlopen(libeary\_mas[curlib],RTLD\_LAZY);

if(!adres){

cout<<"Error";

exit(EXIT\_FAILURE);

}

SinIntegral = (float(\*)(float, float, float))dlsym(adres,"SinIntegral");//возвращаем адрес функции из памяти библиотеки /dlsym присваивает указателю на функцию, объявленному в начале, ее адрес в библиотеке

Derivative = (float(\*)(float, float))dlsym(adres, "Derivative");

}

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

}

else if (command==1){

float A, B, e;

cin>>A>>B>>e;

float sinintegral =SinIntegral(A,B,e);

if (sinintegral==-1){

cout<<"please enter again\n";

}

else{

cout<<"SinIntegral("<< A<<","<<B<<","<< e<<")="<<SinIntegral(A, B, e)<< endl;

}

}

else if (command==2){

float A, deltaX;

cin>>A>>deltaX;

float derivative =Derivative(A, deltaX);

if (derivative==-1){

cout<<"please enter again\n";

}

else{

cout<< "Derivative("<<A<<","<<deltaX<<")="<<Derivative(A,deltaX)<<endl;

}

}

else{

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

}

}

dlclose(adres);

return 0;

}

**library1.cpp**

#include <cmath>

extern "C" float SinIntegral(float A, float B, float e);

extern "C" float Derivative(float A, float deltaX);

float SinIntegral(float A, float B, float e){

float Integral = 0;

for(float i = A; i<B; i+=e){

Integral += sinf(i) \*e;

}

return Integral;

}

float Derivative(float A, float deltaX){

return (cosf (A + deltaX) - cosf(A))/deltaX;

}

**library2.cpp**

#include <cmath>

extern "C" float SinIntegral(float A, float B, float e);

extern "C" float Derivative(float A, float deltaX);

float SinIntegral(float A, float B, float e){

float Integral = 0;

for (float i = A + e; i<B; i+=e){

Integral += (sinf(i)+sinf(i-e))/ 2\*e;

}

return Integral;

}

float Derivative(float A, float deltaX) {

return (cosf(A + deltaX) - cosf(A - deltaX)) / (2 \* deltaX);

}

**Выводы**

Эта лабораторная работа ознакомила и научила меня работать с динамическими библиотеками. Я создала динамические библиотеки функций подсчета интеграла функции sin(x) и вычисления производной от cos(x). Также я поработала с теорией, поняла основной принцип динамических библиотек, их различие от статических.