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: Казанцев Данила Игоревич

Группа: М8О-207Б-21

Вариант:8

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2022

**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/thgdanilaya/mai\_os\_labs

**Постановка задачи**

**Цель работы**

Целью является приобретение практических навыков в:

Создание динамических библиотек

Создание программ, которые используют функции динамических библиотек

**Задание**

Требуется создать динамические библиотеки, которые реализуют определенный функционал.

Далее использовать данные библиотеки 2-мя способами:

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

Во время исполнения программы. Библиотеки загружаются в память с помощью

интерфейса ОС для работы с динамическими библиотеками

В конечном итоге, в лабораторной работе необходимо получить следующие части:

Динамические библиотеки, реализующие контракты, которые заданы вариантом;

Тестовая программа (программа №1), которая используют одну из библиотек, используя

знания полученные на этапе компиляции;

Тестовая программа (программа №2), которая загружает библиотеки, используя только их

местоположение и контракты.

Провести анализ двух типов использования библиотек.

**Общие сведения о программе**

Есть две реализации – realisation1.cpp and realisation2.cpp, в которых находятся реализации по варианту. В файле main1.cpp статическое решение, в main2.cpp - динамическое

**Исходный код**

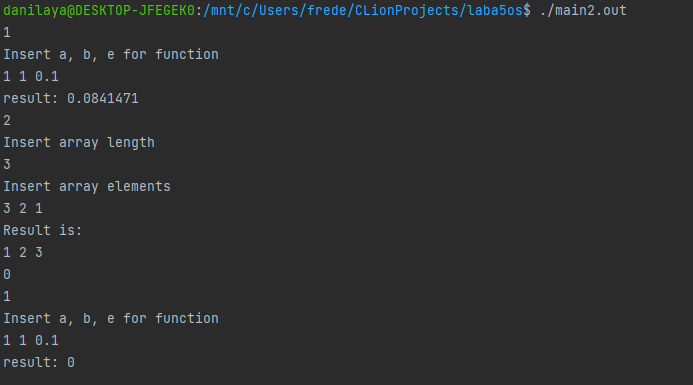
<…>main1.cpp#include <iostream>  
#include "../include/lib.h"  
#include <stdlib.h>  
#include <dlfcn.h>  
  
using namespace std;  
  
  
int main() {  
 int command;  
 cout << "Insert a command\n 1 - engage sin function\n 2 - engage Sort function\n";  
 cout << SYSTEM << '\n';  
  
 while (cin >> command) {  
 if (command == 1) { //sin function  
 float a, b, e;  
 cout << "Enter A, B, e: ";  
 cin >> a >> b >> e;  
 cout << "Integral value " << SinIntegral(a, b, e) << '\n';  
 break;  
 }  
 if (command == 2) { //sort function  
 cout << "Insert array length\n";  
 int N;  
 cin >> N;  
 int array[N];  
 cout << "Insert array elements\n";  
 for (int i = 0; i < N; ++i) {  
 cin >> array[i];  
 };  
 int size = sizeof(array) / sizeof(array[0]);  
 Sort(array, size - 1, 0);  
 cout << "Result is:\n";  
 for (int i = 0; i < N; ++i) {  
 cout << array[i] << " ";  
 }  
 cout << "\n";  
 break;  
 } else {  
 cout << "--Wrong command!--\n";  
 cout << "Insert a command\n 0 - change library\n 1 - engage Pi function\n 2 - engage Sort function\n";  
 }  
 }  
  
  
cout << "Shutting down...\n";  
  
return 0;  
}

Main2.cpp#include <iostream>  
#include "../include/lib.h"  
#include <stdlib.h>  
#include <dlfcn.h>  
  
using namespace std;  
  
  
int main() {  
  
 int command;  
 string lib1 = "./libdynamic1.so";  
 string lib2 = "./libdynamic2.so";  
 void \*curlib = dlopen(lib1.c\_str(), RTLD\_LAZY);  
  
 int \*(\*Sort)(int \*array, int size, int left);  
 float (\*SinIntegral)(float a, float b, float e);  
  
 Sort = (int \*(\*)(int \*, int, int)) dlsym(curlib, "Sort");  
 SinIntegral = (float (\*)(float, float, float)) dlsym(curlib, "SinIntegral");  
  
 int lib\_id = 1;  
  
 while (cin >> command) {  
 switch (command) {  
 case 0: { //change lib  
 dlclose(curlib);  
 if (lib\_id == 1) {  
 curlib = dlopen(lib2.c\_str(), RTLD\_LAZY);  
 lib\_id = 2;  
 } else {  
 curlib = dlopen(lib1.c\_str(), RTLD\_LAZY);  
 lib\_id = 1;  
 }  
 Sort = (int \*(\*)(int \*, int, int)) dlsym(curlib, "Sort");  
 SinIntegral = (float (\*)(float, float, float)) dlsym(curlib, "SinIntegral");  
 continue;  
 }  
 case 1: { //sin function  
 cout << "Insert a, b, e for function\n";  
 float a, b, e;  
 cin >> a >> b >> e;  
 float ans = SinIntegral(a, b, e);  
 cout << "result: " << ans << '\n';  
 break;  
 }  
 case 2: { //sort function  
 cout << "Insert array length\n";  
 int N;  
 cin >> N;  
 int array[N];  
 cout << "Insert array elements\n";  
 for (int i = 0; i < N; ++i) {  
 cin >> array[i];  
 };  
 int size = sizeof(array) / sizeof(array[0]);  
 Sort(array, size - 1, 0);  
 cout << "Result is:\n";  
 for (int i = 0; i < N; ++i) {  
 cout << array[i] << " ";  
 }  
 cout << "\n";  
 break;  
 }  
 default: {  
 cout << "--Wrong command!--\n";  
 cout << "Insert a command\n 0 - change library\n 1 - engage Pi function\n 2 - engage Sort function\n";  
 }  
 }  
 }  
 cout << "Library closed.\n";  
}

Realistion1.cpp#include <iostream>  
#include <cmath>  
#include "../include/lib.h"  
  
using namespace std;  
  
  
int\* Sort(int \* array, int size, int left = 0) {  
 ++size;  
 int tmp;  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size - 1; j++) {  
 if (array[j] > array[j + 1]) {  
 tmp = array[j];  
 array[j] = array[j + 1];  
 array[j + 1] = tmp;  
 }  
 }  
 }  
}  
  
  
  
float SinIntegral(float a, float b, float e)  
{  
 float square = 0;  
 for (float i = a; i <= b; i += e) {  
 square += e \* sin(i);  
 }  
 return square;  
}

Realisation2.cpp#include <iostream>  
#include <cmath>  
#include "../include/lib.h"  
using namespace std;  
  
int \*Sort(int \*s\_arr, int last, int first) {  
 int i = first, j = last, x = s\_arr[(first + last) / 2];  
 do {  
 while (s\_arr[i] < x) i++;  
 while (s\_arr[j] > x) j--;  
  
 if (i <= j) {  
 if (s\_arr[i] > s\_arr[j]) {  
 int t;  
 t = s\_arr[i];  
 s\_arr[i] = s\_arr[j];  
 s\_arr[j] = t;  
 }  
 i++;  
 j--;  
 }  
 } while (i <= j);  
  
 if (i < last)  
 Sort(s\_arr, last, i);  
 if (first < j)  
 Sort(s\_arr, j, first);  
}  
  
  
float SinIntegral(float a, float b, float e)  
{  
 float square = 0;  
 for (float i = a; i < b; i += e) {  
 square += e \* ((sin(i) + sin(i + e)) / 2);  
 }  
 return square;  
}

**Демонстрация работы программы**



**Выводы**

Я приобрел навыки в создании динамических библеотек и программ, которые используют эти библеотеки