Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №4 по курсу**

**«Операционные системы»**

Студент: Ибрагимов Далгат Магомедалиевич

Группа: М8О-208Б-22

Вариант: 8

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2023

**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/L0ckR/OS_LABS>

**Постановка задачи**

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

**Общие сведения о программе**

Программа состоит из одного заголовочного файла – lib.h и из двух его реализаций – implementation1.cpp и implementation2.cpp, которые будут линковаться разными способами в файлы dynamic\_main.cpp и static\_main.cpp соответственно. Программа написана под ОС Linux.

Были использованы следующие системные вызовы:

1. dlopen()
2. dlsym()

**Общий метод и алгоритм решения**

В файлах implementation1.cpp и implementation2.cpp воплощены разные реализации функций с одним и тем же интерфейсом, которые по разному будут линковаться в файлах dynamic\_main.cpp и static\_main.cpp

**Исходный код**

**======================== implementations.hpp ========================**

**#pragma once**

**#include <iostream>**

**#include <cmath>**

**#ifdef \_\_cplusplus**

**extern "C" {**

**#endif**

**constexpr int NUM\_POINTS = 3000; // число разбиений**

**const float PI = 3.1415926535;**

**float SinIntegral(float a, float b, float e);**

**int\* Sort(int\* array, size\_t size);**

**#ifdef \_\_cplusplus**

**}**

**#endif**

**======================== implementation1.cpp =======================**

**#include "implementations.hpp"**

**extern "C" float SinIntegral(float a, float b, float e) {**

**float integral = 0.0;**

**e = (b - a) / NUM\_POINTS;**

**for (int i = 0; i <= NUM\_POINTS; ++i) {**

**integral = integral + e \* sin(a + e \* (i - 0.5));**

**}**

**return integral;**

**}**

**extern "C" int \* Sort(int \* array, size\_t size){**

**if (size == 1 || size == 0) {**

**return array;**

**}**

**for (size\_t i = 0; i < size; i++) {**

**for (int j = size - 1; j >= 1; j--) {**

**if (array[j] < array[j - 1]) {**

**int c = array[j];**

**array[j] = array[j - 1];**

**array[j - 1] = c;**

**}**

**}**

**}**

**return array;**

**}**

**======================== implementation2.cpp =======================**

**#include "implementations.hpp"**

**#include <vector>**

**extern "C" void swap(int\* lhs, int\* rhs) {**

**int c = \*lhs;**

**\*lhs = \*rhs;**

**\*rhs = c;**

**}**

**extern "C" float SinIntegral(float a, float b, float e) {**

**float integral = 0.0;**

**e = (b - a) / NUM\_POINTS;**

**for (int i = 1; i < NUM\_POINTS; ++i) {**

**float x1 = a + i \* e;**

**float x2 = a + (i + 1) \* e;**

**integral += 0.5 \* e \* (sin(x1) + sin(x2));**

**}**

**return integral;**

**}**

**extern "C" void QuickSort(int\* array, int first, int last) {**

**int i = first, j = last, x = array[(first + last) / 2];**

**do {**

**while (array[i] < x) i++;**

**while (array[j] > x) j--;**

**if(i <= j) {**

**if (array[i] > array[j]) swap(&array[i], &array[j]);**

**i++;**

**j--;**

**}**

**} while (i <= j);**

**if (i < last) {**

**QuickSort(array, i, last);**

**}**

**if (first < j) {**

**QuickSort(array, first, j);**

**}**

**}**

**extern "C" int \* Sort(int \* array, size\_t size){**

**if (size == 1 || size == 0) {**

**return array;**

**}**

**QuickSort(array, 0, size - 1);**

**return array;**

**}**

**========================== dynamic\_main.cpp ==========================**

**#include "utils.hpp"**

**int main() {**

**const char \*pathToLib1 = getenv("PATH\_TO\_LIB1");**

**const char \*pathToLib2 = getenv("PATH\_TO\_LIB2");**

**// bash: export PATH\_TO\_LIB1="/home/lockr/projects/OS\_labs/build/lab4/libimplementation1.so"**

**// bash: export PATH\_TO\_LIB2="/home/lockr/projects/OS\_labs/build/lab4/libimplementation2.so"**

**void\* libraryHandle = LoadLibrary(pathToLib1);**

**SinIntegralFunc SinIntegral = (SinIntegralFunc)dlsym(libraryHandle, "SinIntegral");**

**SortFunc Sort = (SortFunc)dlsym(libraryHandle, "Sort");**

**std::string command;**

**while(true) {**

**std::cout << "Enter the command (0 - switch implementation, e - exit): ";**

**std::cin >> command;**

**if (command == "e") {**

**break;**

**} else if (command == "0") {**

**std::cout << "Enter the library (1 or 2): ";**

**std::cin >> command;**

**if (command == "1") {**

**libraryHandle = LoadLibrary(pathToLib1);**

**} else if (command == "2") {**

**libraryHandle = LoadLibrary(pathToLib2);**

**} else {**

**std::cout << "Invalid library" << std::endl;**

**}**

**SinIntegral = (SinIntegralFunc)dlsym(libraryHandle, "SinIntegral");**

**Sort = (SortFunc)dlsym(libraryHandle, "Sort");**

**} else {**

**if (command == "1") {**

**std::cout << "SinIntegral function:" << std::endl;**

**float arg1, arg2, arg3;**

**std::cin >> arg1 >> arg2 >> arg3;**

**float result = SinIntegral(arg1, arg2, arg3);**

**std::cout << "Result of integral = " << result << std::endl;**

**} else if (command == "2") {**

**std::cout << "Sort function:" << std::endl;**

**std::cout << "Enter size of array:" << std::endl;**

**size\_t size;**

**std::cin >> size;**

**std::cout << "Enter members of array:" << std::endl;**

**int \*arr = new int[size];**

**for (size\_t i = 0; i < size; i++) {**

**std::cin >> arr[i];**

**}**

**int \*result = Sort(arr, size);**

**for (size\_t i = 0; i < size; i++) {**

**std::cout << result[i] << " ";**

**}**

**std::cout << std::endl;**

**} else {**

**std::cout << "Invalid command" << std::endl;**

**}**

**}**

**}**

**UnloadLibrary(libraryHandle);**

**return 0;**

**}**

**========================== static\_main.cpp ==========================**

**#include "implementations.hpp"**

**#include <iostream>**

**void Task(const std::string& command) {**

**if (command == "1") {**

**float arg1, arg2, arg3;**

**std::cin >> arg1 >> arg2 >> arg3;**

**float result = SinIntegral(arg1, arg2, arg3);**

**std::cout << "Result of integral = " << result << std::endl;**

**} else if (command == "2") {**

**std::cout << "Sort function" << std::endl;**

**std::cout << "Enter size of array:" << std::endl;**

**size\_t size;**

**std::cin >> size;**

**std::cout << "Enter members of array:" << std::endl;**

**int \*arr = new int[size];**

**for (size\_t i = 0; i < size; i++) {**

**std::cin >> arr[i];**

**}**

**int \*result = Sort(arr, size);**

**for (size\_t i = 0; i < size; i++) {**

**std::cout << result[i] << " ";**

**}**

**std::cout << std::endl;**

**} else {**

**std::cout << "Invalid command" << std::endl;**

**}**

**}**

**int main() {**

**std::string command;**

**while(true) {**

**std::cout << "Enter the command (0 - exit): ";**

**std::cin >> command;**

**if (command == "0") {**

**break;**

**}**

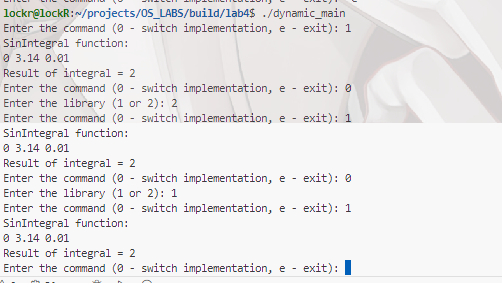
**Task(command);**

**}**

**return 0;**

**}**

**Демонстрация работы программы**

****

**Выводы**

В ходе выполнения лабораторной работе были получены глубокие знания о процессе сборки и компиляции программ на языке Си и C++. Была изучена разница между динамической линковкой на этапе компиляции и на этапе выполнения программы.