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: Немкова Анастасия Романовна

Группа: М8О-208Б-22

Преподаватель: Миронов Евгений Сергеевич

Вариант: 13

Оценка: _____

Дата: _____

Подпись: _____

Москва, 2023

Содержание

Репозиторий

`https://github.com/anastasia-nemkova/OS_labs`

Постановка задачи

Цель работы:

Создание динамических библиотек и создание программ, которые используют функции динамических библиотек

Задание:

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 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.

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.

Вариант 13:

Таблица 1: Контракты и реализации функций

№	Описание	Сигнатура	Реализация 1	Реализация 2
2	Рассчет производной функции $\cos(x)$ в точке A с приращением δx	`Float Derivative (float A, float deltaX)`	$f'(x) = \frac{f(A+deltaX)-f(A)}{deltaX}$	$f'(x) = \frac{f(A+deltaX)-f(A-deltaX)}{2 \cdot deltaX}$
7	Подсчет площади плоской геометрической фигуры по двум сторонам	`Float Square(float A, float B)`	Фигура прямоугольник	Фигура прямоугольный треугольник

Общие сведения о программе

Программа компилируется из файлов realizations.hpp, realization1.cpp, realization2.cpp, static_main.cpp, dynamic_main.cpp. Также имеются файлы с тестами lab4_realiz1_test.cpp и lab4_realiz2_test.cpp В программе работы были использованы следующие системные вызовы:

- dlsym() - получение адреса функции из динамической библиотеки
- dlopen() - открытие динамической библиотеки
- dlclose() - закрытие динамеческой библиотеки
- dlerror() - возвращение строки, описывающей последнюю ошибку, произошедшую при вызове функций из динамической библиотеки

Общий метод и алгоритм решения

У нас имеется статическая и динамическая реализации. Для первой из них мы компилируем основной файл вместе с динамической библиотекой. Для второй используем переменные окружения PATH_TO_LIB1 и PATH_TO_LIB2 для определения путей к двум различным динамическим библиотекам.

В динамической реализации основной файл программы загружает библиотеки с помощью системного вызова dlopen и получает указатели на функции из этих библиотек с помощью dlsym. При вводе команды "0"пользователем программа переключает одну библиотеку на другую.

В статической реализации функции из динамических библиотек вызываются напрямую из основного файла программы, так как эти функции уже были статически связаны с основным файлом в процессе компиляции

Исходный код

realizations.hpp

```
#pragma once

#include <iostream>

#ifdef __cplusplus
extern "C" {
#endif

float Derivative(float a, float deltaX);
float Square(float a, float b);

#ifdef __cplusplus
}
#endif
```

realization1.cpp

```
#include "realizations.hpp"
#include <cmath>

extern "C" float Derivative(float a, float deltaX) {
    return (cos(a + deltaX) - cos(a)) / deltaX;
}

extern "C" float Square(float a, float b) {
    return a * b;
}
```

realization2.cpp

```
#include "realizations.hpp"
#include <cmath>

extern "C" float Derivative(float a, float deltaX) {
    return (cos(a + deltaX) - cos(a)) / (2 * deltaX);
}

extern "C" float Square(float a, float b) {
    return (a * b) / 2;
}
```

static_main.cpp

```
#include "realizations.hpp"

#include <iostream>
#include <sstream>


void mainRoutine(const std::string& command) {
    std::istringstream iss(command);
    int operation;
    iss >> operation;

    if (operation == 1) {
        float arg1, arg2;
        iss >> arg1 >> arg2;
        float result = Derivative(arg1, arg2);
        std::cout << "Result of Derivative: " << result << std::endl;
    } else if (operation == 2) {
        float arg1, arg2;
        iss >> arg1 >> arg2;
        float result = Square(arg1, arg2);
        std::cout << "Result of Square: " << result << std::endl;
    } else {
        std::cout << "Invalid command." << std::endl;
    }
```

```
}

int main() {
    std::string command;

    while (true) {
        std::cout << "Enter command (0 to exit): ";
        std::getline(std::cin, command);

        if (command == "0") {
            break;
        }

        mainRoutine(command);
    }

    return 0;
}
```

dynamic_main.cpp

```
#include <iostream>
#include <sstream>
#include <cstdlib>
#include <dlfcn.h>

using DerivativeFunc = float (*)(float, float);
using SquareFunc = float (*)(float, float);

void* loadLibrary(const std::string& libraryName) {
    void* handle = dlopen(libraryName.c_str(), RTLD_LAZY);
    if (!handle) {
        std::cerr << "Cannot load library: " << dlerror() << std::endl;
        exit(EXIT_FAILURE);
    }
    return handle;
}

void unloadLibrary(void* handle) {
    if (dlclose(handle) != 0) {
        std::cerr << "Cannot unload library: " << dlerror() << std::endl;
        exit(EXIT_FAILURE);
    }
}

int main() {
    const char* pathToLib1 = getenv("PATH_TO_LIB1");
    const char* pathToLib2 = getenv("PATH_TO_LIB2");

    if (pathToLib1 == nullptr || pathToLib2 == nullptr) {
        std::cout << "PATH_TO_LIB1 or PATH_TO_LIB2 is not specified\n";
        exit(1);
    }


    void* libraryHandle = loadLibrary(pathToLib1);


    DerivativeFunc Derivative = (DerivativeFunc)dlsym(libraryHandle, "Derivative")
    ;
    SquareFunc Square = (SquareFunc)dlsym(libraryHandle, "Square");

    std::string command;

    while (true) {
        std::cout << "Enter command (0 to switch libraries, 1 or 2 to call
    functions, q to exit): ";
        std::getline(std::cin, command);

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

```
48              unloadLibrary(libraryHandle);
49
50              std::cout << "Enter library number (1 or 2): ";
51              std::getline(std::cin, command);
52
53              if (command == "1") {
54                  libraryHandle = loadLibrary(pathToLib1);
55              } else if (command == "2") {
56                  libraryHandle = loadLibrary(pathToLib2);
57              } else {
58                  std::cout << "Invalid library number.\n";
59                  exit(1);
60              }
61
62              Derivative = reinterpret_cast<DerivativeFunc>(dlsym(libraryHandle, "
    Derivative"));
63              Square = reinterpret_cast<SquareFunc>(dlsym(libraryHandle, "Square"));
64
65          } else if (command == "q") {
66              break;
67
68          } else {
69              std::istringstream iss(command);
70              int operation;
71              iss >> operation;
72
73              if (operation == 1) {
74                  float arg1, arg2;
75                  iss >> arg1 >> arg2;
76                  float result = Derivative(arg1, arg2);
77                  std::cout << "Result of Derivative: " << result << std::endl;
78
79              } else if (operation == 2) {
80                  float arg1, arg2;
81                  iss >> arg1 >> arg2;
82                  float result = Square(arg1, arg2);
83                  std::cout << "Result of Square: " << result << std::endl;
84
85              } else {
86                  std::cout << "Invalid command." << std::endl;
87              }
88          }
89      }
90
91      unloadLibrary(libraryHandle);
92
93      return 0;
94  }
95
96  // export PATH_TO_LIB1="/home/arnemkova/OS_labs/lab4/build/librealization1.so"
97  // export PATH_TO_LIB2="/home/arnemkova/OS_labs/lab4/build/librealization2.so"
```

lab4_realiz1_test.cpp

```
1  #include "gtest/gtest.h"
2  #include "realizations.hpp"
3
4  TEST(FourthLabTest, DerivativeStaticTest) {
5      float result = Derivative(0.0, 0.1);
6      EXPECT_FLOAT_EQ(result, -0.049958348);
7  }
8
9  TEST(FourthLabTest, SquareStaticTest) {
10     float result = Square(2.0, 3.0);
11     EXPECT_FLOAT_EQ(result, 6.0);
12 }
13
14 int main(int argc, char **argv) {
15     testing::InitGoogleTest(&argc, argv);
16     return RUN_ALL_TESTS();
```

```
17 }
```

lab4_realiz2_test.cpp

```
1  #include "gtest/gtest.h"
2  #include "realizations.hpp"
3
4  TEST(FourthLabTest, DerivativeStaticTest) {
5      float result = Derivative(0.0, 0.1);
6      EXPECT_FLOAT_EQ(result, -0.024979174);
7  }
8
9  TEST(FourthLabTest, SquareStaticTest) {
10     float result = Square(2.0, 3.0);
11     EXPECT_FLOAT_EQ(result, 3.0);
12 }
13
14 int main(int argc, char **argv) {
15     testing::InitGoogleTest(&argc, argv);
16     return RUN_ALL_TESTS();
17 }
```

Демонстрация работы программы

Тесты для статической реализации

```
arnemkova@LAPTOP-TA2RV74U:~/OS_labs/build$ ./lab4/static_main
Enter command (0 to exit): 1 3 5
Result of Derivative: 0.168898
Enter command (0 to exit): 2 4 5
Result of Square: 20
Enter command (0 to exit): 0
```

Тесты для динамической реализации

```
arnemkova@LAPTOP-TA2RV74U:~/OS_labs/build$ ./lab4/dynamic_main
Enter command (0 to switch libraries, 1 or 2 to call functions, q to exit): 1
 3 5
Result of Derivative: 0.168898
Enter command (0 to switch libraries, 1 or 2 to call functions, q to exit): 2 4 5
Result of Square: 20
Enter command (0 to switch libraries, 1 or 2 to call functions, q to exit): 0
Enter library number (1 or 2): 2
Enter command (0 to switch libraries, 1 or 2 to call functions, q to exit): 1 3 5
Result of Derivative: 0.0844492
Enter command (0 to switch libraries, 1 or 2 to call functions, q to exit): 2 4 5
Result of Square: 10
```

Тесты

```
arnemkova@LAPTOP-TA2RV74U:~/OS_labs/build$ ./tests/lab4_realiz1_test
[==========] Running 2 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 2 tests from FourthLabTest
[ RUN      ] FourthLabTest.DerivativeStaticTest
[       OK ] FourthLabTest.DerivativeStaticTest (0 ms)
[ RUN      ] FourthLabTest.SquareStaticTest
[       OK ] FourthLabTest.SquareStaticTest (0 ms)
[----------] 2 tests from FourthLabTest (0 ms total)
```

```
[----------] Global test environment tear-down
[==========] 2 tests from 1 test suite ran. (0 ms total)
[  PASSED  ] 2 tests.
arnemkova@LAPTOP-TA2RV74U:~/OS_labs/build$ ./tests/lab4_realiz2_test
[==========] Running 2 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 2 tests from FourthLabTest
[ RUN      ] FourthLabTest.DerivativeStaticTest
[       OK ] FourthLabTest.DerivativeStaticTest (0 ms)
[ RUN      ] FourthLabTest.SquareStaticTest
[       OK ] FourthLabTest.SquareStaticTest (0 ms)
[----------] 2 tests from FourthLabTest (0 ms total)

[----------] Global test environment tear-down
[==========] 2 tests from 1 test suite ran. (0 ms total)
[  PASSED  ] 2 tests.
```

Вывод

В ходе данной лабораторной работы я познакомилась с использованием динамических библиотек в ОС Linux, которые позволяют программе загружать и использовать функции из библиотек во время выполнения, что обеспечивает гибкость и возможность изменения программы без перекомпиляции. Также я узнала про этапы сборки программы и особенности использования extern "C"при линковке файлов с общим include.