

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Институт №8 "Компьютерные науки и прикладная математика"
Кафедра 806 "Вычислительная математика и программирование"

Лабораторная работа №4
По курсу «Операционные системы»

Студент: Снетков Н.С.
Группа: М8О-208Б-23
Вариант: 28
Преподаватель: Миронов Е. С.

Дата: _____

Оценка: _____

Подпись: _____

Москва, 2024

Содержание

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

Репозиторий

<https://github.com/mxdesta/osLabs/tree/main/lab5>

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 заданный вариантом 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

информацию полученные на этапе компиляции;

- Тестовая программа (программа №2), которая загружает библиотеки, используя только их относительные пути и контракты.

Провести анализ двух типов использования библиотек.

Контракты и реализации функций(**мой вариант**):

5	Расчет значения числа Пи при заданной длине ряда (K)	float Pi(int K)	Ряд Лейбница	Формула Валлиса
7	Подсчет площади плоской геометрической фигуры подвум сторонам	Float Square(float A, float B)	Фигура прямоугольник	Фигура прямоугольный треугольник

Общие сведения о программе

Программа представляет собой приложение для вычисления числа π и площади фигуры с использованием двух динамических библиотек (`lib1.so` и `lib2.so`). Библиотеки содержат функции `Pi` (вычисление числа π с использованием ряда Лейбница или формулы Валлиса) и `Square` (вычисление площади прямоугольника или прямоугольного треугольника). Программа позволяет динамически переключаться между библиотеками во время выполнения и взаимодействует с пользователем через консольный интерфейс.

Общий метод и алгоритм решения

Алгоритм работы программы

Загрузка библиотек: Программа загружает функции `Pi` и `Square` из динамических библиотек с помощью `dlopen` и `dlsym`. По умолчанию загружается первая библиотека (`lib1.so`).

Взаимодействие с пользователем: Программа принимает команды от пользователя:

0 — переключение между библиотеками.

1 K — вычисление числа π с длиной ряда K.

2 A B — вычисление площади фигуры с параметрами A и B.

Выполнение команд: В зависимости от выбранной команды вызывается соответствующая функция из текущей библиотеки. Результаты выводятся в консоль.

Завершение работы: Программа завершает работу при закрытии консоли или прерывании пользователем.

Исходный код

functions.h:

```
// include/functions.h

#ifndef FUNCTIONS_H
#define FUNCTIONS_H

extern "C" {
```

```
float Pi(int K);  
float Square(float A, float B);  
}
```

```
#endif // FUNCTIONS_H
```

lib1.cpp:

```
#include <cmath>  
#include "../include/functions.h"  
  
extern "C" {  
    float Pi(int K) {  
        float pi = 0.0;  
        for (int i = 0; i < K; i++) {  
            pi += pow(-1, i) / (2 * i + 1);  
        }  
        return pi * 4;  
    }  
  
    float Square(float A, float B) {  
        return A * B; // Площадь прямоугольника  
    }  
}
```

lib2.cpp:

```
#include <cmath>  
#include "../include/functions.h"  
  
extern "C" {
```

```

float Pi(int K) {
    double pi = 1.0;
    for (int i = 1; i <= K; i++) {
        pi *= (2.0 * i / (2.0 * i - 1)) * (2.0 * i / (2.0 * i + 1));
    }
    return static_cast<float>(pi * 2);
}

float Square(float A, float B) {
    return 0.5 * A * B; // Площадь прямоугольного треугольника
}
}

```

program1.cpp

```

#include <iostream>

#include "include/functions.h"

int main() {
    int command;

    std::cout << "Введите команду: ";
    std::cin >> command;

    if (command == 1) {
        int K;

        std::cout << "Введите K: ";
        std::cin >> K;

        std::cout << "Pi = " << Pi(K) << std::endl;
    } else if (command == 2) {
        float A, B;
    }
}

```

```

        std::cout << "Введите А и В: ";

        std::cin >> A >> B;

        std::cout << "Площадь = " << Square(A, B) << std::endl;
    }

    return 0;
}

```

program2.cpp:

```

#include <iostream>

#include <dlfcn.h>

#include <string>

#include <cstdlib>

using PiFunc = float (*)(int);

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

void* loadLibrary(const std::string& path, PiFunc& piFunc, SquareFunc& squareFunc) {
    void* handle = dlopen(path.c_str(), RTLD_LAZY);

    if (!handle) {
        std::cerr << "Ошибка загрузки библиотеки: " << dlerror() << std::endl;
        exit(EXIT_FAILURE);
    }

    piFunc = reinterpret_cast<PiFunc>(dlsym(handle, "Pi"));

    if (!piFunc) {
        std::cerr << "Ошибка загрузки функции Pi: " << dlerror() << std::endl;
        dlclose(handle);
        exit(EXIT_FAILURE);
    }
}

```

```

    }

    squareFunc = reinterpret_cast<SquareFunc>(dlsym(handle, "Square"));
    if (!squareFunc) {
        std::cerr << "Ошибка загрузки функции Square: " << dlerror() << std::endl;
        dlclose(handle);
        exit(EXIT_FAILURE);
    }

    return handle;
}

int main() {
    std::string lib1Path = "./lib1.so";
    std::string lib2Path = "./lib2.so";

    void* currentLib = nullptr;
    PiFunc piFunc = nullptr;
    SquareFunc squareFunc = nullptr;

    // Загружаем первую библиотеку по умолчанию
    currentLib = loadLibrary(lib1Path, piFunc, squareFunc);
    bool usingLib1 = true;

    std::cout << "Введите команды: \n";
    std::cout << "0 - переключить библиотеку\n";
    std::cout << "1 K - вычислить Pi с длиной ряда K\n";
    std::cout << "2 A B - вычислить площадь фигуры\n";

```



```

std::string command;
while (true) {
    std::cout << "\nВведите команду: ";
    if (!std::getline(std::cin, command) || command.empty()) {
        continue;
    }

    if (command[0] == '0') {
        // Переключение библиотеки
        dlclose(currentLib);
        if (usingLib1) {
            currentLib = loadLibrary(lib2Path, piFunc, squareFunc);
            usingLib1 = false;
            std::cout << "Переключено на lib2.so\n";
        } else {
            currentLib = loadLibrary(lib1Path, piFunc, squareFunc);
            usingLib1 = true;
            std::cout << "Переключено на lib1.so\n";
        }
    } else if (command[0] == '1') {
        // Вызов функции Pi
        try {
            int K = std::stoi(command.substr(2));
            float result = piFunc(K);
            std::cout << "Pi(" << K << ") = " << result << std::endl;
        } catch (...) {
            std::cerr << "Ошибка: Неверный формат команды. Ожидается '1 K'.\n";
        }
    } else if (command[0] == '2') {

```

```

// Вызов функции Square
try {
    size_t pos = command.find(' ', 2);
    float A = std::stof(command.substr(2, pos - 2));
    float B = std::stof(command.substr(pos + 1));
    float result = squareFunc(A, B);
    std::cout << "Square(" << A << ", " << B << ") = " << result << std::endl;
} catch (...) {
    std::cerr << "Ошибка: Неверный формат команды. Ожидается '2 A B'.\n";
}
} else {
    std::cout << "Неизвестная команда. Попробуйте снова.\n";
}
}

if (currentLib) {
    dlclose(currentLib);
}

return 0;
}

```

Демонстрация работы программы

unix@DESKTOP-MPQDBS2:~/labs/osLabs/build/lab5\$./prog2

Введите команды:

0 - переключить библиотеку

1 К - вычислить P_i с длиной ряда К

2 A B - вычислить площадь фигуры

Введите команду: 1

Ошибка: Неверный формат команды. Ожидается '1 K'.

Введите команду: 1 5

$Pi(5) = 3.33968$

Введите команду: 1 10 5

$Pi(10) = 3.04184$

Введите команду: 0

Переключено на lib2.so

Введите команду: 1 5

$Pi(5) = 3.00218$

Введите команду: 0

Переключено на lib1.so

Введите команду:

Выводы

В процессе работы с программой были изучены ключевые концепции программирования на языке C++ и работы с динамическими библиотеками. Я научился использовать функции **dlopen**, **dlsym** и **dlclose** для динамической загрузки и работы с библиотеками, что позволяет гибко выбирать реализации функций во время выполнения. Этот опыт помог мне лучше понять принципы работы с динамическими библиотеками.