

Московский Авиационный Институт
(Национальный Исследовательский Университет)
Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

Лабораторная работа №4 по курсу
«Операционные системы»

Тема работы
“Динамические библиотеки”

Студент: Слободин Никита Алексеевич
Группа: М8О-203Б-23
Вариант: 34

Преподаватель: Миронов Евгений Сергеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2024

Постановка задачи

Задача: 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 использует одну из библиотек, используя знания, полученные на этапе компиляции;
- Тестовая программа (программа №2), которая загружает библиотеки, используя только их местоположение и контракты.

Задание 7. Реализовать функцию `Float Square(float A, float B)` для вычисления площади плоской геометрической фигуры. В рамках задания требуется вычислить площадь прямоугольника и прямоугольного треугольника по заданным сторонам AAA и BBB.

Задание 8. Реализовать функцию `Char* translation(long x)` для перевода числа xxx из десятичной системы счисления в другую. В частности, необходимо обеспечить перевод в двоичную и троичную системы счисления.

Общие сведения о программе

Программа состоит из двух основных частей: консольное приложение и динамически подключаемые библиотеки. Консольное приложение предоставляет пользователю меню для выполнения следующих операций: вычисление площади прямоугольника, вычисление площади прямоугольного треугольника, перевод числа в двоичную или троичную систему счисления. В

зависимости от команды, приложение вызывает соответствующие функции, реализованные в библиотеках `geometry` и `convert`. При этом библиотеки загружаются динамически с использованием `dlopen`, что позволяет переключаться между ними в процессе работы программы.

Функции в библиотеке `geometry` отвечают за расчет площади геометрических фигур, а в библиотеке `convert` — за перевод чисел в другие системы счисления. Вывод результата обработки данных осуществляется непосредственно в консоль.

Исходный код программы приведен в приложении.

Выводы

В ходе выполнения лабораторной работы была создана программа, демонстрирующая работу с динамически подключаемыми библиотеками, обработку пользовательского ввода и выполнение вычислительных операций. Динамическая загрузка библиотек позволила оптимизировать управление функциональностью, предоставляя возможность переключения между библиотеками без остановки программы. Программа обеспечивает корректную обработку ошибок ввода и исключительных ситуаций, что улучшает ее надежность и стабильность. Таким образом, задачи по реализации вычислений площади фигур и переводу чисел в другие системы счисления выполнены с использованием современных механизмов динамической работы с библиотеками.

Приложение

src/lib1.cpp

```
#include <cmath>
#include <algorithm>
#include "functions.h"

// Реализация 1: Площадь прямоугольника
extern "C" {
    float Square(float A, float B) {
        return A * B; // Площадь прямоугольника
    }

    // Реализация 2: Перевод в двоичную систему
    char* translation(long x) {
        static char buffer[65]; // Достаточно для представления числа в 64 бита
        int i = 0;
        if (x == 0) {
            buffer[i++] = '0';
        } else {
            while (x > 0) {
                buffer[i++] = (x % 2) + '0';
                x /= 2;
            }
        }
        buffer[i] = '\0';

        // Переворачиваем строку
        for (int j = 0, k = i - 1; j < k; ++j, --k) {
            std::swap(buffer[j], buffer[k]);
        }

        return buffer;
    }
}
```

src/lib2.cpp

```
#include <cmath>
#include <algorithm>
#include "functions.h"

// Реализация 1: Площадь прямоугольного треугольника
extern "C" {
    float Square(float A, float B) {
        return 0.5f * A * B; // Площадь прямоугольного треугольника
    }

    // Реализация 2: Перевод в троичную систему
    char* translation(long x) {
        static char buffer[65]; // Достаточно для представления числа в 64 бита
```

```

    int i = 0;
    if (x == 0) {
        buffer[i++] = '0';
    } else {
        while (x > 0) {
            buffer[i++] = (x % 3) + '0';
            x /= 3;
        }
    }
    buffer[i] = '\\0';

    // Переворачиваем строку
    for (int j = 0, k = i - 1; j < k; ++j, --k) {
        std::swap(buffer[j], buffer[k]);
    }

    return buffer;
}
}

```

src/program2.cpp

```

#include <iostream>
#include <dlfcn.h>
#include <cstring>

using SquareFunc = float(*)(float, float);
using TranslationFunc = char*(*)(long);

int main() {
    std::cout << "Программа №2 (динамическая загрузка библиотек)\n";

    const char* pathToLib1 = std::getenv("PATH_TO_LIB1");
    if (!pathToLib1) {
        std::cerr << "Переменная окружения PATH_TO_LIB1 не установлена" << std::endl;
        return 1;
    }

    const char* pathToLib2 = std::getenv("PATH_TO_LIB2");
    if (!pathToLib2) {
        std::cerr << "Переменная окружения PATH_TO_LIB2 не установлена" << std::endl;
        return 1;
    }

    const char* lib_paths[] = {pathToLib1, pathToLib2};
    int current_lib = 0;
    void* handle = dlopen(lib_paths[current_lib], RTLD_LAZY);

    if (!handle) {
        std::cerr << "Ошибка загрузки библиотеки: " << dlerror() << "\n";
        return 1;
    }
}

```

```

    SquareFunc Square = reinterpret_cast<SquareFunc>(dlsym(handle, "Square"));
    TranslationFunc translation = reinterpret_cast<TranslationFunc>(dlsym(handle,
"translation"));

    char* error;
    if ((error = dlerror()) != nullptr) {
        std::cerr << "Ошибка получения символа: " << error << "\n";
        dlclose(handle);
        return 1;
    }

    while (true) {
        std::cout << "\nТекущая библиотека: " << lib_paths[current_lib] << "\n";
        std::cout << "Введите команду:\n";
        std::cout << "0 – переключить реализацию\n";
        std::cout << "1 A B – вычислить площадь (прямоугольник или треугольник)\n";
        std::cout << "2 x – перевести число в другую систему счисления\n";
        std::cout << "3 – выйти\n";

        int command;
        std::cin >> command;

        if (command == 0) {
            dlclose(handle);
            current_lib = 1 - current_lib;
            handle = dlopen(lib_paths[current_lib], RTLD_LAZY);
            if (!handle) {
                std::cerr << "Ошибка загрузки библиотеки: " << dlerror() << "\n";
                return 1;
            }
            Square = reinterpret_cast<SquareFunc>(dlsym(handle, "Square"));
            translation = reinterpret_cast<TranslationFunc>(dlsym(handle,
"translation"));
        } else if (command == 1) {
            float A, B;
            std::cin >> A >> B;
            float result = Square(A, B);
            std::cout << "Площадь: " << result << "\n";
        } else if (command == 2) {
            long x;
            std::cin >> x;
            char* result = translation(x);
            std::cout << "Число в другой системе счисления: " << result << "\n";
        } else if (command == 3) {
            break;
        } else {
            std::cout << "Неверная команда\n";
        }
    }

    dlclose(handle);
    return 0;

```

```
}  
  
// export PATH_TO_LIB1=$(pwd)/lab4/lib1.dylib  
// export PATH_TO_LIB2=$(pwd)/lab4/lib2.dylib  
// echo $PATH_TO_LIB1  
// echo $PATH_TO_LIB2
```

Пример вывода:

```
root@c34508d80232:/workspaces/OS_MAI_Slobodin/build#  
./lab4/program2
```

Программа №2 (динамическая загрузка библиотек)

Текущая библиотека:
/workspaces/OS_MAI_Slobodin/build/lab4/lib1.so

Введите команду:

0 - переключить реализацию
1 A B - вычислить площадь (прямоугольник или треугольник)
2 x - перевести число в другую систему счисления
3 - выйти
1 10 10

Площадь: 100

Текущая библиотека:
/workspaces/OS_MAI_Slobodin/build/lab4/lib1.so

Введите команду:

0 - переключить реализацию
1 A B - вычислить площадь (прямоугольник или треугольник)
2 x - перевести число в другую систему счисления
3 - выйти
0

Текущая библиотека:
/workspaces/OS_MAI_Slobodin/build/lab4/lib2.so

Введите команду:

0 - переключить реализацию
1 A B - вычислить площадь (прямоугольник или треугольник)

2 x - перевести число в другую систему счисления

3 - выйти

1 10 10

Площадь: 50

Текущая
/workspaces/OS_MAI_Slobodin/build/lab4/lib2.so

библиотека:

Введите команду:

0 - переключить реализацию

1 A B - вычислить площадь (прямоугольник или треугольник)

2 x - перевести число в другую систему счисления

3 - выйти

2

7

Число в другой системе счисления: 21

Текущая
/workspaces/OS_MAI_Slobodin/build/lab4/lib2.so

библиотека:

Введите команду:

0 - переключить реализацию

1 A B - вычислить площадь (прямоугольник или треугольник)

2 x - перевести число в другую систему счисления

3 - выйти

0

Текущая
/workspaces/OS_MAI_Slobodin/build/lab4/lib1.so

библиотека:

Введите команду:

0 - переключить реализацию

1 A B - вычислить площадь (прямоугольник или треугольник)

2 x - перевести число в другую систему счисления

3 - выйти

2 1000

Число в другой системе счисления: 1111101000

Текущая
/workspaces/OS_MAI_Slobodin/build/lab4/lib1.so

библиотека:

Введите команду:

0 - переключить реализацию

1 A B - вычислить площадь (прямоугольник или треугольник)

2 x - перевести число в другую систему счисления

3 - выйти

3

root@c34508d80232:/workspaces/OS_MAI_Slobodin/build#