

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: Тарасов Егор Дмитриевич
Группа: М8О-209Б-23
Вариант: 19
Преподаватель: Миронов Евгений Сергеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2024

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

Постановка задачи

Цель работы

Приобретение практических навыков в:

⑩ 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:

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:

⑩ Динамические библиотеки, реализующие контракты, которые заданы вариантом;

⑩ Тестовая программа (программа No1), которая использует одну из библиотек, используя знания полученные на этапе компиляции;

⑩ Тестовая программа (программа No2), которая загружает библиотеки, используя только их местоположение и контракты.

Провести анализ двух типов использования библиотек.

Пользовательский ввод для обеих программ должен быть организован следующим образом:

1. Если пользователь вводит команду «0», то программа переключает одну реализацию контрактов на другую (необходимо только для программы No2). Можно реализовать лабораторную работу без данной функции, но максимальная оценка в этом случае будет «хорошо»;
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

N	Описание	Сигнатура	Реализация 1	Реализация 2
3	Подсчёт количества простых	Int PrimeCount(int	Наивный	Решето

	чисел на отрезке [A, B] (A, B - натуральные)	A, int B)	алгоритм. Проверить делимость текущего числа на все предыдущие числа.	Эратосфена
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 состоит из файлов, реализующих работу с библиотеками (library1.h, library1.cpp, library2.cpp), а также из файлов программ (program1.cpp, program2.cpp).

Используемые библиотечные вызовы:

HMODULE LoadLibrary(LPCTSTR lpLibFileName);	Загружает динамическую библиотеку, указанную в строке lpLibFileName, и возвращает её дескриптор.
FARPROC GetProcAddress (HMODULE hModule, LPCSTR lpProcName);	Возвращает указатель на функцию или переменную, экспортируемую библиотекой, загруженной с помощью LoadLibrary.
BOOL FreeLibrary (HMODULE hModule);	Освобождает загруженную библиотеку, выгружая её из памяти.
DWORD GetLastError();	Возвращает код последней ошибки, вызванной функцией Windows API, например, если не удалось загрузить библиотеку или найти функцию.

Общий метод и алгоритм решения

Для реализации поставленной задачи необходимо:

1. Изучить работу с библиотеками.
2. Реализовать две библиотеки согласно заданию.
3. Реализовать две программы (для работы с динамическими и статическими библиотеками).

Исходный код

Library1.h

```
#ifndef LIBRARY1_H
#define LIBRARY1_H

#ifdef _WIN32
    #ifdef BUILDING_DLL
        #define DLL_EXPORT __declspec(dllexport)
    #else
        #define DLL_EXPORT __declspec(dllimport)
    #endif
#else
    #define DLL_EXPORT
#endif

extern "C" DLL_EXPORT int PrimeCount(int A, int B);
extern "C" DLL_EXPORT float Square(float A, float B);

#endif
```

Library1.cpp

```
#define BUILDING_DLL
#include "library1.h"

// Реализация PrimeCount: наивный алгоритм
int PrimeCount(int A, int B) {
    int count = 0;
    for (int i = A; i <= B; i++) {
        bool isPrime = true;
        if (i < 2) continue;
        for (int j = 2; j * j <= i; j++) {
            if (i % j == 0) {
                isPrime = false;
                break;
            }
        }
        if (isPrime) count++;
    }
    return count;
}

// Реализация Square: площадь прямоугольника
float Square(float A, float B) {
    return A * B;
}
```

Library2.cpp

```
#include <windows.h>
#include <vector>

// Реализация PrimeCount: Решето Эратосфена
extern "C" __declspec(dllexport) int PrimeCount(int A, int B) {
    if (B < 2) return 0;

    std::vector<bool> isPrime(B + 1, true);
    isPrime[0] = isPrime[1] = false;

    for (int i = 2; i * i <= B; i++) {
        if (isPrime[i]) {
            for (int j = i * i; j <= B; j += i) {
                isPrime[j] = false;
            }
        }
    }

    int count = 0;
    for (int i = A; i <= B; i++) {
        if (isPrime[i]) count++;
    }
    return count;
}

// Реализация Square: площадь прямоугольного треугольника
extern "C" __declspec(dllexport) float Square(float A, float B) {
    return 0.5f * A * B;
}
```

Program1.cpp

```
#include <iostream>
#include "library1.h"

int main() {
    int choice;
    while (true) {
        std::cout << "Enter a command (1 or 2): ";
        std::cin >> choice;

        if (choice == 1) {
            int A, B;
            std::cout << "Enter A and B to count prime numbers: ";
            std::cin >> A >> B;
            std::cout << "Number of primes: " << PrimeCount(A, B) << std::endl;
        } else if (choice == 2) {
```

```

        float A, B;
        std::cout << "Enter sides A and B: ";
        std::cin >> A >> B;
        std::cout << "Area: " << Square(A, B) << std::endl;
    } else {
        break;
    }
}
return 0;
}

```

Program2.cpp

```

#include <iostream>
#include <windows.h>

typedef int (*PrimeCountFunc)(int, int);
typedef float (*SquareFunc)(float, float);

int main() {
    HMODULE hLib = nullptr;
    PrimeCountFunc PrimeCount = nullptr;
    SquareFunc Square = nullptr;
    int currentLibrary = 1;

    hLib = LoadLibrary("liblibrary1.dll");
    if (!hLib) {
        std::cerr << "Failed to load library1.dll" << std::endl;
        return 1;
    }
    PrimeCount = (PrimeCountFunc)GetProcAddress(hLib, "PrimeCount");
    Square = (SquareFunc)GetProcAddress(hLib, "Square");

    int choice;
    while (true) {
        std::cout << "Enter a command (0, 1, or 2): ";
        std::cin >> choice;

        if (choice == 0) {
            FreeLibrary(hLib);
            currentLibrary = (currentLibrary == 1) ? 2 : 1;
            hLib = LoadLibrary(currentLibrary == 1 ? "liblibrary1.dll" : "libli-
brary2.dll");
            if (!hLib) {
                std::cerr << "Failed to load the library" << std::endl;
                return 1;
            }
            PrimeCount = (PrimeCountFunc)GetProcAddress(hLib, "PrimeCount");
            Square = (SquareFunc)GetProcAddress(hLib, "Square");
        }
    }
}

```

```

        std::cout << "Switched to " << (currentLibrary == 1 ? "library1.dll"
: "library2.dll") << std::endl;
    } else if (choice == 1) {
        int A, B;
        std::cout << "Enter A and B to count prime numbers: ";
        std::cin >> A >> B;
        std::cout << "Number of primes: " << PrimeCount(A, B) << std::endl;
    } else if (choice == 2) {
        float A, B;
        std::cout << "Enter sides A and B: ";
        std::cin >> A >> B;
        std::cout << "Area: " << Square(A, B) << std::endl;
    } else {
        break;
    }
}

FreeLibrary(hLib);
return 0;
}

```

Демонстрация работы программы

PS C:\Users\Xiaomi\Desktop\l4\build\bin> ./program1

Enter a command (1 or 2): 1

Enter A and B to count prime numbers: 10 20

Number of primes: 4

Enter a command (1 or 2): 2

Enter sides A and B: 5 10

Area: 50

PS C:\Users\Xiaomi\Desktop\l4\build\bin> ./program2

Enter a command (0, 1, or 2): 1

Enter A and B to count prime numbers: 10 20

Number of primes: 4

Enter a command (0, 1, or 2): 2

Enter sides A and B: 5 10

Area: 50

Enter a command (0, 1, or 2): 0

Switched to library2.dll

Enter a command (0, 1, or 2): 1

Enter A and B to count prime numbers: 10 20

Number of primes: 4

Enter a command (0, 1, or 2): 2

Enter sides A and B: 5 10

Area: 25

Выводы

В ходе лабораторной работы я познакомился с созданием динамических библиотек в операционной системе Windows, а также с возможностью загружать эти библиотеки в ходе выполнения программы. Динамические библиотеки (DLL) помогают уменьшить размер исполняемых файлов, так как функции и ресурсы библиотеки хранятся отдельно и могут быть разделяемыми между несколькими программами. Загрузка динамических библиотек во время выполнения программы (с использованием функций Windows API, таких как LoadLibrary И GetProcAddress) позволяет сделать программу более гибкой, так как можно подключать разные реализации функций без необходимости перекомпиляции. Это особенно полезно в модульных системах или при реализации плагинов. Благодаря динамическим библиотекам, мы можем разрабатывать сложные программы, которые используют повторно написанные ранее функции, структуры и другие элементы, сохранённые в библиотеках. Это позволяет значительно сократить время разработки, улучшить читаемость кода и повысить его модульность.