Министерство науки и высшего образования Российской Федерации  
Пензенский государственный университет  
Кафедра вычислительная техника

**ОТЧЕТ**

по лабораторной работе №5

по дисциплине «Основы операционных систем»

на тему «Динамически загружаемые библиотеки»

Выполнили студенты группы 22ВВП1

Демин М. С.

Беляев Д. И.

Приняли:

Егоров В. Ю.

Федюнин Р. Н.

Пенза 2024

**Название**

Динамически загружаемые библиотеки

**Цель работы**

Изучить связывание процесса с динамически загружаемыми библиотеками на этапе загрузки и на этапе выполнения.

**Лабораторное задание**

Необходимо разработать программу, состоящую из головной программы и двух динамически загружаемых библиотек. Одна библиотека должна загружаться с использованием динамического связывания, а другая – с использованием функции LoadLibrary. Библиотеки должны выполнять действия по вариантам заданий из лабораторной работы №4. (Разработать программу периодической проверки свободного места на жестком диске (функция GetDiskFreeSpace) и сбора данной информации на едином сервере с использованием почтового ящика.)

**Описание данных**

1. Типы указателей на функции:

• typedef void (\*StartClientFunc)();

• Это определение типа для указателя на функцию, которая не принимает аргументов и не возвращает значения. Функция StartClient предназначена для запуска клиентской части приложения.

• typedef void (\*StartServerFunc)();

• Аналогично, это определение для указателя на функцию, которая запускает серверную часть приложения.

2. Переменные для хранения библиотек:

• HMODULE hClientLib: Переменная для хранения дескриптора загруженной клиентской библиотеки.

• HMODULE hServerLib: Переменная для хранения дескриптора загруженной серверной библиотеки.

**Описание структуры программы**

1. Загрузка клиентской библиотеки:

• Программа использует LoadLibrary для загрузки динамической библиотеки clientLib.dll. Если загрузка не удалась, выводится сообщение об ошибке, и программа завершает выполнение.

2. Получение адреса функции StartClient:

• С помощью GetProcAddress программа получает адрес функции StartClient из загруженной библиотеки. Если получение адреса не удалось, выводится сообщение об ошибке, освобождается загруженная библиотека и программа завершается.

3. Загрузка серверной библиотеки:

• Аналогично, программа загружает динамическую библиотеку serverLib.dll. Если загрузка не удалась, выводится сообщение об ошибке, и программа завершает выполнение.

4. Получение адреса функции StartServer:

• Программа получает адрес функции StartServer из загруженной серверной библиотеки. Если получение адреса не удалось, выводится сообщение об ошибке, освобождается библиотека, и программа завершает выполнение.

5. Запуск серверного кода:

• После успешного получения адреса функции StartServer, программа вызывает эту функцию, что инициирует выполнение серверной логики.

6. Запуск клиентского кода:

• Затем вызывается функция StartClient, что инициирует выполнение клиентской логики.

7. Освобождение ресурсов:

• В конце программы освобождается память, занятой клиентской библиотекой с помощью FreeLibrary.

**Текст программы**

#include <stdio.h>

#include <windows.h>

// #include "serverLib.h"

// Прототип функции из клиентской библиотеки

typedef void (\*StartClientFunc)();

typedef void (\*StartServerFunc)();

int main() {

// Загрузка библиотеки клиента с использованием LoadLibrary

HMODULE hClientLib = LoadLibrary(TEXT("clientLib.dll"));

if (!hClientLib) {

printf("Failed to load client library. Error: %d\n", GetLastError());

return 1;

}

// Получение адреса функции StartClient из библиотеки клиента

StartClientFunc StartClient = (StartClientFunc)GetProcAddress(hClientLib, "StartClient");

if (!StartClient) {

printf("Failed to get StartClient function address. Error: %d\n", GetLastError());

FreeLibrary(hClientLib);

return 1;

}

// Загрузка библиотеки клиента с использованием LoadLibrary

HMODULE hServerLib = LoadLibrary(TEXT("serverLib.dll"));

if (!hServerLib) {

printf("Failed to load server library. Error: %d\n", GetLastError());

return 1;

}

// Получение адреса функции StartServer из библиотеки клиента

StartServerFunc StartServer = (StartServerFunc)GetProcAddress(hServerLib, "StartServer");

if (!StartServer) {

printf("Failed to get StartServer function address. Error: %d\n", GetLastError());

FreeLibrary(hServerLib);

return 1;

}

// Запуск серверного кода

StartServer();

// Запуск клиентского кода

StartClient();

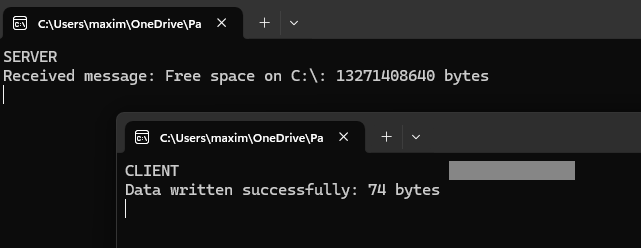
// Освобождение библиотеки клиента

FreeLibrary(hClientLib);

return 0;

}

**Результат работы программы**



**Вывод**

Изучили как делать связывание процесса с динамически загружаемыми библиотеками на этапе загрузки и на этапе выполнения.