

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: Кривошапкин Егор Борисович  
Группа: М8О-209Б-23  
Вариант: 8  
Преподаватель: Миронов Евгений Сергеевич  
Оценка: \_\_\_\_\_  
Дата: \_\_\_\_\_  
Подпись: \_\_\_\_\_

Москва, 2024

## **Содержание**

1. Репозиторий
2. Постановка задачи
3. Демонстрация работы программы
4. Выводы

## Репозиторий

[https://github.com/EgorX2000/os\\_labs/tree/main/3](https://github.com/EgorX2000/os_labs/tree/main/3)

### Постановка задачи

#### Цель работы

Целью является приобретения 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 использует одну из библиотек, используя информацию, полученную на этапе компиляции;
- Тестовая программа (*программа №2*), которая загружает библиотеки, используя только их относительные пути и контракты.

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 №2*). 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.

Контракты и функции для реализации:

№	Описание	Сигнатура	Реализация 1	Реализация 2
1	Расчёт интеграла функции $\sin(x)$ на отрезке $[A, B]$ с шагом $e$	float SinIntegral(float A, float B, float e)	Подсчёт интеграла методом прямоугольников.	Подсчёт интеграла методом трапеций.
2	Отсортировать целочисленный массив	int * Sort(int * array)	Пузырьковая сортировка	Сортировка Хоара

### Демонстрация работы программы

```
PS D:\Documents\C++\os_labs\4\src> cd .\build\  
PS D:\Documents\C++\os_labs\4\src\build> ./comp  
Result: 1.99999  
PS D:\Documents\C++\os_labs\4\src\build> ./comp  
Enter command: 2 5 3 4 2 5 1  
Sorted array: 1 2 3 4 5  
PS D:\Documents\C++\os_labs\4\src\build> ./exec  
Switching library. Enter 1 for Lib1 or 2 for Lib2: 1  
Library switched successfully.  
Enter command: 1 0 3.14 0.01  
Result: 1.99999  
PS D:\Documents\C++\os_labs\4\src\build> ./exec  
Enter command: 0  
Switching library. Enter 1 for Lib1 or 2 for Lib2: 1  
Library switched successfully.  
Enter command: 2 5 3 4 2 5 1  
Sorted array: 1 2 3 4 5  
PS D:\Documents\C++\os_labs\4\src\build> ./exec  
Enter command: 0  
Switching library. Enter 1 for Lib1 or 2 for Lib2: 2
```

Library switched successfully.

Enter command: 1 0 3.14 0.01

Result: 1.99995

PS D:\Documents\C++\os\_labs\4\src\build> ./exec

Enter command: 0

Switching library. Enter 1 for Lib1 or 2 for Lib2: 2

Library switched successfully.

Enter command: 2 5 3 4 2 5 1

Sorted array: 1 2 3 4 5

PS D:\Documents\C++\os\_labs\4\src\build>

## 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.