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: Муханов Андрей Ильич

Группа: М8О-209Б-23

Вариант: 12

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2024

**Содержание**

1. Репозиторий
2. Постановка задачи
3. Шаги по изменению лабораторной номер 1
4. Демонстрация работы программы
5. Выводы

**Репозиторий**

https://github.com/andreymuch/OC\_MAI

**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 No1), которая используют одну из библиотек, используя информацию полученные на этапе компиляции;

• Тестовая программа (программа No2), которая загружает библиотеки, используя только их относительные пути и контракты.

Провести анализ двух типов использования библиотек.  
Пользовательский ввод для обоих 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

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.

1. Реализация динамических библиотек

Были созданы две динамические библиотеки, реализующие контракты для функций `Derivative` и `E`. Каждая библиотека содержит свою реализацию этих функций.

Библиотека 1:  
Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Библиотека 2:

Изображение выглядит как текст, снимок экрана, Шрифт, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

2. Тестовая программа No1 (статическое связывание)

Программа No1 использует одну из библиотек на этапе компиляции. В данном случае используется библиотека с реализацией функций из `func1.h`.

Код программы No1:

Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Пример использования:

- Ввод: `1 1.0 0.001`

Вывод: -0.84178

- Ввод: `2 10`

Вывод: 11

3. Тестовая программа No2 (динамическое связывание)

Программа No2 загружает библиотеки динамически на этапе выполнения. Она использует функции `LoadLibrary` и `GetProcAddress` для загрузки и вызова функций из библиотеки.

Код программы No2:

Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Пример использования:

- Ввод: `1 1.0 0.001`

Вывод: -0.84178.

- Ввод: `2 10`

Вывод: 1.71828

4. Анализ двух типов использования библиотек

Статическое связывание (программа No1):

- Преимущества:

- Программа полностью автономна, не зависит от внешних файлов.

- Нет накладных расходов на загрузку библиотеки во время выполнения.

- Недостатки:

- Исполняемый файл имеет больший размер.

- Для обновления библиотеки требуется перекомпиляция программы.

Динамическое связывание (программа No2):

- Преимущества:

- Исполняемый файл имеет меньший размер.

- Библиотеку можно обновлять без перекомпиляции программы.

- Возможность загружать разные реализации библиотек в зависимости от условий.

- Недостатки:

- Накладные расходы на загрузку библиотеки во время выполнения.

- Зависимость от наличия библиотеки в системе.

Вывод

В ходе выполнения лабораторной работы были успешно реализованы:

- Две динамические библиотеки с разными реализациями функций.

- Программа No1, использующая статическое связывание.

- Программа No2, использующая динамическое связывание.

Проведен анализ преимуществ и недостатков обоих подходов. Программа No2 также поддерживает переключение между реализациями библиотек, что позволяет гибко управлять функциональностью во время выполнения.