Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №5 по курсу**

**«Операционные системы»**

Студент: Дегтярев Денис Андреевич

Группа: М8О-207Б-21

Вариант: 20

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2022

**Содержание**

1. Репозиторий
2. Постановка задачи
3. Общие сведения о программе
4. Общий метод и алгоритм решения
5. Исходный код
6. Выводы

**Репозиторий**

https://github.com/CHISH08/OCI/tree/main/5lab

**Постановка задачи**

**Тема**

Динамические библиотеки

## Цель работы

Целью является приобретение практических навыков в:

* Создание динамических библиотек
* Создание программ, которые используют функции динамических библиотек
* Работа со сборочной системой

## 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:

* Динамические библиотеки, реализующие контракты, которые заданы вариантом;
* Тестовая программа (*программа №1*), которая используют одну из библиотек, используя знания полученные на этапе компиляции;
* Тестовая программа (*программа №2*), которая загружает библиотеки, используя только их местоположение 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 №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.

**Варианты**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | Рассчет производной функции cos(x) в точке A с приращением deltaX | Float Derivative(float A, float deltaX) | f'(x) = (f(A + deltaX) – f(A))/deltaX | f'(x) = (f(A + deltaX) – f(A-deltaX))/(2\*deltaX) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | Подсчёт наибольшего общего делителя для двух натуральных чисел | Int GCF(int A, int B) | Алгоритм Евклида | Наивный алгоритм. Пытаться разделить числа на все числа, что меньше A и B. |

**Общие сведения о программе**

* **dlopen** динамически открывает нужную нам библиотеку.
* **dlsym** ищет нам нужную функцию

**Общий метод и алгоритм решения**

**В библиотеках записываем код функций.**

**С помощью Cmake превращаем обычные библиотеки в динамические.**

**С помощью dlopen и dlsym используем функции библиотек.**

**Исходный код(Расположен в репозитории)**

**Выводы**

Динамические библиотеки дают нам возможность ускорять работу программы и позволяют нам экономить память засчет того, что библиотеки не надо объявлять.