

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Институт №8 "Компьютерные науки и прикладная математика"
Кафедра 806 "Вычислительная математика и программирование"

Лабораторная работа №4
По курсу «Операционные системы»

Студент: Власко М. М.

Группа: М8О-208Б-23

Преподаватель: Живалев Е. А.

Дата: _____

Оценка: _____

Подпись: _____

Москва, 2024

Тема: Динамические библиотеки

Цель работы: Приобретение практических навыков в:

- Создании динамических библиотек.
- Создании программ, использующих функции динамических библиотек.

Вариант: 27.

Задачи:

1. Создать динамические библиотеки, реализующие два контракта:
 - Расчет числа π (Пи) двумя способами: по ряду Лейбница и по формуле Валлиса.
 - Расчет числа e (основание натурального логарифма) двумя способами: $(1 + 1/x)^x$ и суммированием ряда с элементами вида $1/(n!)$.
2. Реализовать две программы для работы с библиотеками:
 - Первая программа использует функции библиотек на этапе компиляции.
 - Вторая программа загружает библиотеки динамически во время исполнения и позволяет переключаться между их реализациями.
3. Реализовать возможность переключения реализаций библиотек во время выполнения программы.

Ход работы:

1. Создание динамических библиотек Были реализованы две динамические библиотеки с различными реализациями функций для вычисления числа π и числа e .

Функции библиотеки №1:

- `float Pi(int K):` Вычисление числа π по ряду Лейбница.
- `float E(int x):` Вычисление числа e по формуле $(1 + 1/x)^x$.

Функции библиотеки №2:

- `float Pi(int K):` Вычисление числа π по формуле Валлиса.
- `float E(int x):` Вычисление числа e через суммирование ряда с элементами вида $1/(n!)$.

Репозиторий: <https://github.com/MMVlasko/mai-os-lab-work/tree/main/LW4>

Код библиотеки №1:

```
#include <math.h>

float Pi(const int K) {
    float pi = 0.0;
    int sign = 1;
    for (int i = 0; i < K; i++) {
        pi += sign * 4.0 / (2 * i + 1);
        sign = -sign;
    }
    return pi;
}

float E(const int x) {
    if (x == 0) {
        return 1;
    }
    return pow(1.0 + 1.0 / x, x);
}
```

Код библиотеки №2:

```
float Pi(const int K) {
    float pi = 1.0;
    for (int i = 1; i <= K; i++) {
        pi *= (4.0 * i * i) / (4.0 * i * i - 1);
    }
    return pi * 2;
}

float E(const int x) {
    float e = 1.0;
    float fact = 1.0;

    for (int n = 1; n <= x; n++) {
        fact *= n;
        e += 1.0 / fact;
    }
    return e;
}
```

2. Первая программа использует функции динамических библиотек на этапе компиляции. Пользователь может вызывать функции для вычисления числа π и числа e , вводя соответствующие команды.

3. Вторая программа загружает динамические библиотеки во время выполнения. Пользователь может переключаться между реализациями библиотек и вызывать функции для вычисления числа π и числа e .

Пример работы:

```
./lab4l
```

```
Usage:
```

```
    -1 - exit
```

```
    1 - find Pi
```

```
    2 - find E
```

```
command> 1 10000
```

```
3.141498
```

```
command> 2 10000
```

```
2.718146
```

```
command> -1
```

```
./lab4d
```

```
Usage:
```

```
    -1 - exit
```

```
    0 - change implementation
```

```
    1 - find Pi
```

```
    2 - find E
```

```
command> 1 10000
```

```
3.141498
```

```
command> 0
```

```
Implementation changed to second
```

```
command> 1 10000
```

```
3.141330
```

```
command> -1
```

Выводы: В ходе выполнения лабораторной работы были созданы две динамические библиотеки с различными реализациями функций для вычисления числа π и числа e . Реализация программ продемонстрировала преимущества и особенности работы с динамическими библиотеками:

1. Статическая линковка обеспечивает более высокую производительность программы, так как подключение библиотек происходит на этапе компиляции.
2. Динамическая загрузка библиотек позволяет изменять функциональность программы без её повторной компиляции, что особенно удобно для приложений, требующих гибкости и расширяемости.
3. Возможность переключения между реализациями функций в динамически загружаемых библиотеках даёт возможность проводить сравнение производительности или функциональности различных подходов.

Полученные результаты подтвердили преимущества использования динамических библиотек и соответствуют теоретическим ожиданиям.