

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»**

Группа: М8О-210БВ-24

Студент: Лабутин М.А.

Преподаватель: Бахарев В.Д.

Оценка: \_\_\_\_\_

Дата: 30.11.25

Москва, 2025

## Постановка задачи

**Вариант 31.**

**Функция 1:**

-Реализация №1: вычисление по формуле  $(1 + 1/x)^x$

-Реализация №2: вычисление как суммы ряда  $1/n!$  от  $n=0$  до  $x$

**Функция 2:**

-Реализация №1: Фигура прямоугольник

-Реализация №2: Фигура прямоугольный треугольник

## Общий метод и алгоритм решения

- `void *dlopen(const char *filename, int flags);` – загружает в память и открывает динамическую библиотеку.
- `void *dlsym(void *handle, const char *symbol);` – возвращает адрес функции или переменной из загруженной библиотеки.
- `int dlclose(void *handle);` – выгружает из памяти ранее загруженную динамическую библиотеку.

Я создал две библиотеки, которые реализуют два контракта двух функций. Также я сделал два вида подключения этой библиотеки в программу: статическая библиотека, динамическая библиотека. Первый способ заключается в том, что подключаю библиотеку во время линковки программы. Второй способ заключается в том, что я подключаю библиотеку в runtime через интерфейс ОС для работы с динамическими библиотеками.

## Код программы

lib.h:

```
#pragma once

#include <stdlib.h>

float e(int x);

float area(float a, float b);
```

lib1.c:

```
#include "lib.h"
```

```

#include <math.h>

// Реализация №1: (1 + 1/x)^x

float e(int x) {
    if (x <= 0) return 0;
    return powf(1.0f + 1.0f / x, x);
}

// Реализация №1: Площадь прямоугольника

float area(float a, float b) {
    if (a <= 0 || b <= 0) return 0;
    return a * b;
}

```

lib2.c:

```

#include "lib.h"

// Функция для вычисления факториала

static int factorial(int n) {
    int result = 1;
    for (int i = 2; i <= n; i++) {
        result *= i;
    }
    return result;
}

// Реализация №2: Сумма ряда 1/n!

float e(int x) {
    if (x < 0) return 0;

    float sum = 0;
    for (int n = 0; n <= x; n++) {
        sum += 1.0f / factorial(n);
    }
    return sum;
}

// Реализация №2: Площадь прямоугольного треугольника

float area(float a, float b) {

```

```
    if (a <= 0 || b <= 0) return 0;

    return a * b / 2.0f;

}
```

### static\_part.c:

```
#include <stdio.h>

#include "lib.h"

int main() {

    int choice;

    printf("1 x - вычисление числа e при заданном x\n");

    printf("2 a b - вычисление площади фигуры\n");

    printf("0 - выход\n\n");

    scanf("%d", &choice);

    while (choice) {

        if (choice == 1) {

            int x;

            scanf("%d", &x);

            printf("Результат: %.6f\n", e(x));

        }

        if (choice == 2) {

            float a, b;

            scanf("%f %f", &a, &b);

            printf("Результат: %.6f\n", area(a, b));

        }

        scanf("%d", &choice);

    }

    return 0;

}
```

### dynamic\_part.c:

```
#include <stdio.h>

#include <dlfcn.h>
```

```
typedef float (*e_t)(int);

typedef float (*area_t)(float, float);

char *paths[] = {"./libmy1.so", "./libmy2.so"};

int main() {

    int ind = 0;

    void *lib = dlopen(paths[ind], RTLD_LAZY);

    if (!lib) {

        printf("Ошибка загрузки библиотеки\n");

        return 1;

    }

    e_t e_func = dlsym(lib, "e");

    area_t area_func = dlsym(lib, "area");



    if (!e_func || !area_func) {

        printf("Ошибка загрузки функций\n");

        return 1;

    }

    printf("0 - переключить библиотеку\n");

    printf("1 x - вычисление числа e\n");

    printf("2 a b - вычисление площади фигуры\n");

    printf("-1 - выход\n\n");



    int choice;

    scanf("%d", &choice);

    while (choice != -1) {

        if (choice == 0) {

            dlclose(lib);

            ind = !ind; // переключатель

            lib = dlopen(paths[ind], RTLD_LAZY);

            if (!lib) {

                printf("Ошибка переключения библиотеки\n");

                return 1;

            }

            e_func = dlsym(lib, "e");

            area_func = dlsym(lib, "area");

        }

    }

}
```

```

    if (!e_func || !area_func) {

        printf("Ошибка загрузки функций\n");

        return 1;

    }

    printf("Библиотека переключена (Реализация №%d)\n", ind + 1);

}

if (choice == 1) {

    int x;

    scanf("%d", &x);

    printf("Результат: %.6f\n", e_func(x));

}

if (choice == 2) {

    float a, b;

    scanf("%f %f", &a, &b);

    printf("Результат: %.6f\n", area_func(a, b));

}

scanf("%d", &choice);

}

dlclose(lib);

return 0;
}

```

makefile:

```

all: libmy1.so libmy2.so static_part dynamic_part

libmy1.so: lib1.c lib.h

gcc -fPIC -shared -o libmy1.so lib1.c -lm

libmy2.so: lib2.c lib.h

gcc -fPIC -shared -o libmy2.so lib2.c

static_part: static_part.c lib1.c lib.h

gcc static_part.c lib1.c -o static_part -lm

dynamic_part: dynamic_part.c

gcc dynamic_part.c -o dynamic_part -ldl

clean:

```

```
rm -f *.so static_part dynamic_part
```

## Протокол работы программы

```
mxtvix@WIN-EMVSNHTNF8D:/mnt/c/Users/User/MAI-LABS/OS-25/Tab4$ ./dynamic_part
0 - переключить библиотеку
1 x - вычисление числа е
2 a b - вычисление площади фигуры
-1 - выход

2 3 5
Результат: 15.000000
1 2
Результат: 2.250000
0
Библиотека переключена (Реализация №2)
2 3 5
Результат: 7.500000
1 2
Результат: 2.500000
-1
```

## Вывод

В ходе лабораторной работы я научился создавать динамические библиотеки, создавать программы, использующие динамические библиотеки. Также я научился делать статические библиотеки и использовать их при компиляции моей программы.