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: Белков А.Д.

Преподаватель: Бахарев В.Д.

Оценка: _____

Дата: 23.11.25

Москва, 2025

Постановка задачи

Вариант 17.

Функция 1:

Подсчёт количества простых чисел на отрезке [a, b] (a, b – натуральные):

Сигнатура функции: int prime_count(int a, int b);

- Реализация №1: Наивный алгоритм. Проверить делимость текущего числа на все предыдущие числа.
- Реализация №2: Решето Эратосфена

Функция 2:

Расчет значения числа π при заданной длине ряда (k):

Сигнатура функции: float pi(int k);

- Реализация №1: Ряд Лейбница
- Реализация №2: Формула Валлиса

Общий метод и алгоритм решения

Использованные системные вызовы:

- 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 динамические библиотеки (libmy1.so и libmy2.so), которые реализуют контракты двух функций: prime_count(int a, int b) для подсчёта простых чисел на отрезке [a, b] и pi(int k) для вычисления числа π при длине ряда k. Первая библиотека использует наивный алгоритм и ряд Лейбница, вторая - решето Эратосфена и формулу Валлиса.

Я реализовал два способа использования этих библиотек:
1. static - линковка на этапе компиляции. Библиотека libmy1.so подключается через флаг -lmy1 при компиляции. Информация о зависимости записывается в исполняемый файл, и динамический загрузчик операционной системы автоматически загружает библиотеку при запуске. Поскольку библиотека находится не в стандартных системных путях поиска библиотек (/lib, /usr/lib), нам необходимо указать текущую директорию, сделать это мы можем через переменную окружения LD_LIBRARY_PATH=. при запуске исполняемого файла

2. dynamic - загрузка во время выполнения. Библиотеки загружаются программно через интерфейс операционной системы для работы с динамическими библиотеками: функции dlopen(), dlsym(), dlclose(). Это позволяет переключаться между реализациями (libmy1.so и libmy2.so) во время работы программы по команде пользователя, что невозможно при первом способе, так как

библиотека загружается автоматически при запуске программы и не может быть заменена во время её выполнения.

Код программы

lib.h:

```
#pragma once

#include <stdlib.h>


int prime_count(int a, int b);

float pi(int k);
```

lib1.c:

```
#include "lib.h"


// Наивный алгоритм подсчёта простых чисел

// Проверить делимость текущего числа на все предыдущие числа

int prime_count(int a, int b) {

    int count = 0;

    for (int num = a; num <= b; num++) {

        if (num < 2) {

            continue;

        }

        int is_prime = 1;

        for (int i = 2; i < num; i++) {

            if (num % i == 0) {

                is_prime = 0;

                break;

            }

        }
```

```
        if (is_prime) {
            count++;
        }
    }
    return count;
}

// Ряд Лейбница
float pi(int k) {
    float res = 0;
    for (int i = 0; i < k; i++) {
        res += (i & 1 ? -1. : 1.) / (2 * i + 1);
    }
    return res * 4;
}
```

lib2.c:

```
#include "lib.h"

// Решето Эратосфена
int prime_count(int a, int b) {
    if (b < 2) {
        return 0;
    }

    // Массив: true = простое, false = составное
    int *is_prime = (int *)calloc(b + 1, sizeof(int));

    // Инициализация: все числа >= 2 считаем простыми
```

```
    for (int i = 2; i <= b; i++) {
        is_prime[i] = 1;
    }

    // Отсеивание составных чисел
    for (int i = 2; i * i <= b; i++) {
        if (is_prime[i]) {
            // Вычёркиваем кратные
            for (int composite = i * i; composite <= b; composite += i) {
                is_prime[composite] = 0;
            }
        }
    }

    // Подсчёт простых в [a, b]
    int count = 0;
    int start = (a < 2) ? 2 : a;
    for (int num = start; num <= b; num++) {
        if (is_prime[num]) {
            count++;
        }
    }

    free(is_prime);
    return count;
}


// Формула Валлиса
```

```
float pi(int k) {
    float res = 1;
    for (int i = 1; i <= k; i++) {
        res *= 4. * i * i / (4 * i * i - 1);
    }
    return res * 2;
}
```

static_part.c:

```
#include <stdio.h>
#include "lib.h"

int main() {
    int choice;
    printf("1 a b - подсчёт простых чисел на [a,b]\n");
    printf("2 k - вычисление π при длине ряда k\n");
    printf("0 - выход\n\n");

    scanf("%d", &choice);
    while (choice) {
        if (choice == 1) {
            int a, b;
            scanf("%d %d", &a, &b);
            printf("Результат: %d\n", prime_count(a, b));
        }
        if (choice == 2) {
            int k;
            scanf("%d", &k);
            printf("Результат: %.10f\n", pi(k));
```

```
        }
        scanf("%d", &choice);
    }
    return 0;
}
```

dynamic_part.c:

```
#include <stdio.h>
#include <dlfcn.h>

typedef int (*prime_count_t)(int, int);
typedef float (*pi_t)(int);

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;
    }

    prime_count_t prime_func = dlsym(lib, "prime_count");
    pi_t pi_func = dlsym(lib, "pi");

    if (!prime_func || !pi_func) {
        printf("Ошибка загрузки функций\n");
```

```
    return 1;
}

printf("0 - переключить библиотеку\n");
printf("1 a b - подсчёт простых\n");
printf("2 k - вычисление π\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;
        }

        prime_func = dlsym(lib, "prime_count");
        pi_func = dlsym(lib, "pi");

        if (!prime_func || !pi_func) {
            printf("Ошибка загрузки функций\n");
            return 1;
        }
```

```
            printf("Библиотека переключена\n");
        }


        if (choice == 1) {
            int a, b;
            scanf("%d %d", &a, &b);
            printf("Результат: %d\n", prime_func(a, b));
        }
        if (choice == 2) {
            int k;
            scanf("%d", &k);
            printf("Результат: %.10f\n", pi_func(k));
        }
        scanf("%d", &choice);
    }
    dlclose(lib);
    return 0;
}
```

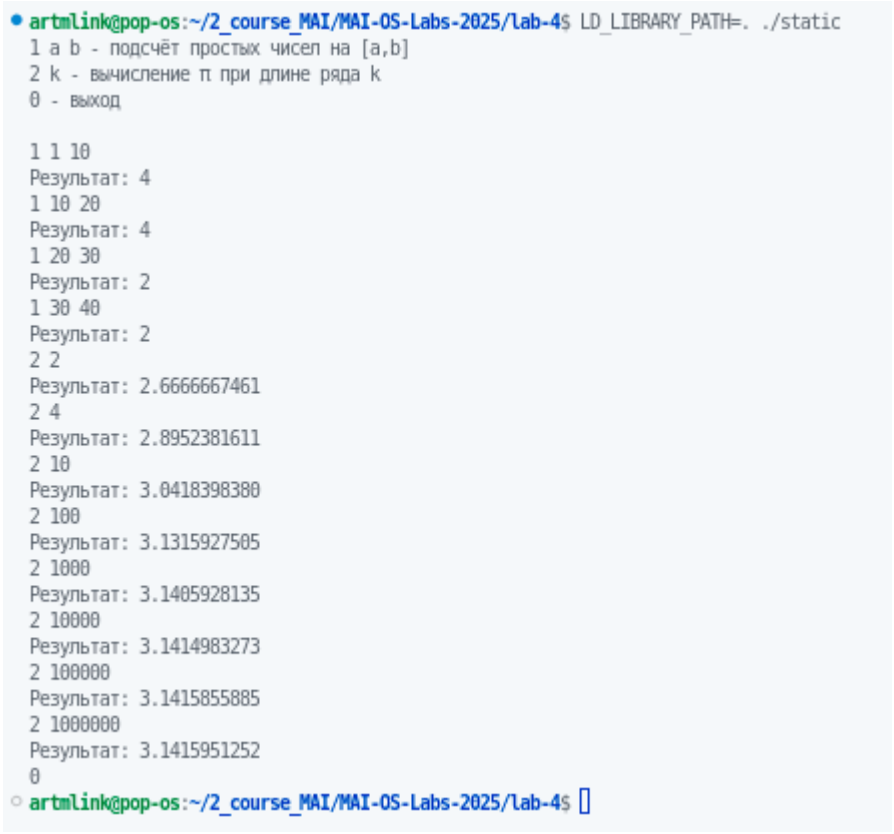
Протокол работы программы

Тестирование:

```
artmlink@pop-os:~/2_course_MAI/MAI-OS-Labs-2025/lab-4$ LD_LIBRARY_PATH=. ./static
1 a b - подсчёт простых чисел на [a,b]
2 k - вычисление π при длине ряда k
0 - выход

1 1 10
Результат: 4
1 10 20
Результат: 4
1 20 30
Результат: 2
```

```
1 30 40
Результат: 2
2 2
Результат: 2.6666667461
2 4
Результат: 2.8952381611
2 10
Результат: 3.0418398380
2 100
Результат: 3.1315927505
2 1000
Результат: 3.1405928135
2 10000
Результат: 3.1414983273
2 100000
Результат: 3.1415855885
2 1000000
Результат: 3.1415951252
0
artmlink@pop-os:~/2_course_MAI/MAI-OS-Labs-2025/lab-4$
```




```
artmlink@pop-os:~/2_course_MAI/MAI-OS-Labs-2025/lab-4$ ./dynamic
0 - переключить библиотеку
1 a b - подсчёт простых
2 k - вычисление π
-1 - выход
```

```
1 1 10
Результат: 4
1 10 20
Результат: 4
1 20 30
Результат: 2
1 30 40
Результат: 2
2 2
Результат: 2.6666667461
2 4
Результат: 2.8952381611
2 10
Результат: 3.0418398380
2 100
Результат: 3.1315927505
2 1000000
Результат: 3.1415951252
0
Библиотека переключена
1 1 100
Результат: 25
1 1 10000000
Результат: 664579
1 1 1000000000
Результат: 50847534
2 100
Результат: 3.1337866783
2 10
Результат: 3.0677034855
-1
artmlink@pop-os:~/2_course_MAI/MAI-OS-Labs-2025/lab-4$
```

```
artmlink@pop-os:~/2_course_MAI/MAI-OS-Labs-2025/lab-4$ ./dynamic
0 - переключить библиотеку
1 a b - подсчёт простых
2 k - вычисление π
-1 - выход

1 1 10
Результат: 4
1 10 20
Результат: 4
1 20 30
Результат: 2
1 30 40
Результат: 2
2 2
Результат: 2.6666667461
2 4
Результат: 2.8952381611
2 10
Результат: 3.0418398380
2 100
Результат: 3.1315927505
2 1000000
Результат: 3.1415951252
0
Библиотека переключена
1 1 100
Результат: 25
1 1 10000000
Результат: 664579
1 1 1000000000
Результат: 50847534
2 100
Результат: 3.1337866783
2 10
Результат: 3.0677034855
-1
artmlink@pop-os:~/2_course_MAI/MAI-OS-Labs-2025/lab-4$
```

Вывод

В ходе работы я освоил создание динамических библиотек и их использование двумя способами: через линковку при компиляции и через динамическую загрузку во время выполнения.