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-214БВ-25

Студент: Лоскутова А. Д.

Преподаватель: Бахарев В.Д.

Оценка: _____

Дата: 12.12.25

Москва, 2025

Постановка задачи

Вариант 5.

Требуется создать две динамические библиотеки, реализующие один и тот же интерфейс (контракт), но с различными алгоритмами обработки данных.

Контракты и реализации:

1. Расчет интеграла функции sin(x) на отрезке [A, B] с шагом e:
   Сигнатура функции: float sin_integral(float a, float b, float e);
   - Реализация №1: Подсчет интеграла методом прямоугольников;
   - Реализация №2: Подсчет интеграла методом трапеций.
2. Расчет значения числа e (основание натурального логарифма):
   Сигнатура функции: float e(int x);
   - Реализация №1: (1 + 1 / x)^x
   - Реализация №2: Сумма ряда по n от 0 до x, где элементы ряда равны: (1 / (n!))

Необходимо разработать две тестовые программы: первая должна использовать неявную (статическую) линковку с одной из библиотек на этапе компиляции, а вторая — реализовывать динамическую загрузку библиотек (runtime loading) для поддержки переключения между различными реализациями по команде пользователя через консоль.

Общий метод и алгоритм решения

Использованные системные вызовы:

● void* dlopen(const char* filename, int flags); – открывает динамическую библиотеку и возвращает дескриптор (handle).
● void* dlsym(void* handle, const char* symbol); – возвращает адрес символа (функции) в памяти загруженной библиотеки.
● int dlclose(void* handle); – уменьшает счетчик ссылок на библиотеку и выгружает её, если счетчик равен 0.
● char* dlerror(void); – возвращает текстовое описание последней ошибки, возникшей в функциях dl*.

В рамках лабораторной работы были созданы два файла исходного кода библиотек (**libmath1.c**, **libmath2.c**), которые компилируются с флагами **-fPIC** (позиционно-независимый код) и **-shared** для создания динамических библиотек **libmath1.so** и **libmath2.so**.

Библиотеки:

● libmath1.so реализует функцию sin_integral методом прямоугольников и функцию e как (1 + 1/x)^x.

● libmath2.so реализует функцию sin_integral методом трапеций и функцию e как сумму ряда 1/n!.

program1:

- Реализует статическую компоновку.
- При компиляции указывается путь к библиотеке (-L.) и её имя (-lmath1).
- Ввод данных парсится с помощью strtok_r и atoi/atof.
- Вызовы функций происходят напрямую через заголовочный файл mathlib.h.

program2:

- Реализует динамическую загрузку библиотек во время выполнения.

1. При запуске или по команде «0» программа вызывает dlopen() для загрузки соответствующего .so файла (libmath1.so или libmath2.so).
2. С помощью dlsym() программа получает указатели на функции sin_integral и e.
3. Вызовы функций происходят через полученные указатели.
4. При переключении библиотек старая библиотека выгружается через dlclose(), после чего загружается новая.

Код программы

mathlib.h

```
#ifndef MATHLIB_H
#define MATHLIB_H

typedef float sin_integral_func(float a, float b, float e);
typedef float e_func(int x);

float sin_integral(float a, float b, float e);
float e(int x);

#endif
```

program1.c

```
#define _DEFAULT_SOURCE
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include "mathlib.h"

extern float sin_integral(float a, float b, float e);
extern float e(int x);

#define BUFFER_SIZE 4096

static void command_1(char **saveptr)
{
```

```
char *arg1 = strtok_r(NULL, " \t\n", saveptr);
char *arg2 = strtok_r(NULL, " \t\n", saveptr);
char *arg3 = strtok_r(NULL, " \t\n", saveptr);

int len = 0;
char buffer[256];

if (arg1 && arg2 && arg3)
{
    float a = atof(arg1);
    float b = atof(arg2);
    float e_step = atof(arg3);

    if (e_step <= 0)
    {
        len = snprintf(buffer, sizeof(buffer), "ошибка значения\n");
    }
    else
    {
        float result = sin_integral(a, b, e_step);
        len = snprintf(buffer, sizeof(buffer),
                    "Интеграл sin(x) результат= %.6f\n",
                    result);
    }
}
else
{
```

```
        len = snprintf(buffer, sizeof(buffer), "недостаточно аргументов\n");
    }
    write(STDOUT_FILENO, buffer, len);
}

static void command_2(char **saveptr)
{
    char *arg1 = strtok_r(NULL, " \t\n", saveptr);

    int len = 0;
    char buffer[256];

    if (arg1)
    {
        int x = atoi(arg1);

        if (x <= 0)
        {
            len = snprintf(buffer, sizeof(buffer), "ошибка значения\n");
        }
        else
        {
            float result = e(x);
            len = snprintf(buffer, sizeof(buffer),
                    "e результат = %.10f\n", result);
        }
    }
```

```
    else
    {
        len = snprintf(buffer, sizeof(buffer), "недостаточно аргументов\n");
    }
    write(STDOUT_FILENO, buffer, len);
}

int main(void)
{
    const char *prompt =
        "Статическая линковка с libmath1.so\n"
        "1 a b e / 2 x / q\n> ";

    write(STDOUT_FILENO, prompt, strlen(prompt));

    int bytes_read = 0;
    char buffer[BUFFER_SIZE];

    while ((bytes_read = read(STDIN_FILENO, buffer, BUFFER_SIZE - 1)) > 0)
    {
        buffer[bytes_read] = '\0';
        for (int i = 0; i < bytes_read; i++)
        {
            if (buffer[i] == '\n' || buffer[i] == '\r')
            {
                buffer[i] = '\0';
                break;
```

```
    }
}

char *saveptr;
char *token = strtok_r(buffer, " \t\n", &saveptr);

if (token == NULL)
{
    write(STDOUT_FILENO, "> ", 2);
    continue;
}

if (strcmp(token, "q") == 0)
    break;

int cmd = atoi(token);

switch (cmd)
{
case 1:
{
    command_1(&saveptr);
    break;
}
case 2:
{
    command_2(&saveptr);
```

```
            break;
        }
        default:
        {
            const char msg[] = "неизвестная команда\n";
            write(STDOUT_FILENO, msg, strlen(msg));
        }
        }
        write(STDOUT_FILENO, "> ", 2);
    }

    write(STDOUT_FILENO, "Exit\n", 6);
    return 0;
}
```

program2.c

```
#define _DEFAULT_SOURCE
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
#include <dlfcn.h>
#include <stdio.h>
#include "mathlib.h"

#define LIB1_PATH "./libmath1.so"
#define LIB2_PATH "./libmath2.so"
#define BUFFER_SIZE 4096
```

```
static float sin_integral_stub(float a, float b, float e)
{
    (void)a;
    (void)b;
    (void)e;
    const char msg[] = "библиотека не загружена или функция не найдена\n";
    write(STDERR_FILENO, msg, sizeof(msg) - 1);
    return 0.0f;
}
static float e_stub(int x)
{
    (void)x;
    const char msg[] = "библиотека не загружена или функция не найдена\n";
    write(STDERR_FILENO, msg, sizeof(msg) - 1);
    return 0.0f;
}

static int load_library(const char *library_path, void **library_handle,
                        sin_integral_func **sin_impl_ptr, e_func **e_impl_ptr)
{

    if (*library_handle != NULL)
    {
        dlclose(*library_handle);
        *library_handle = NULL;
    }
```

```
void *handle = dlopen(library_path, RTLD_LAZY);
if (handle == NULL)
{
    const char msg[] = "ошибка загрузки библиотеки: ";
    write(STDERR_FILENO, msg, sizeof(msg) - 1);
    write(STDERR_FILENO, dlerror(), strlen(dlerror()));
    write(STDERR_FILENO, "\n", 1);

    *sin_impl_ptr = sin_integral_stub;
    *e_impl_ptr = e_stub;
    return 0;
}

*sin_impl_ptr = (sin_integral_func *)dlsym(handle, "sin_integral");
*e_impl_ptr = (e_func *)dlsym(handle, "e");

if (*sin_impl_ptr == NULL || *e_impl_ptr == NULL)
{
    const char msg[] = "не удалось найти функции в библиотеке\n";
    write(STDERR_FILENO, msg, sizeof(msg) - 1);

    dlclose(handle);
    *library_handle = NULL;
    *sin_impl_ptr = sin_integral_stub;
    *e_impl_ptr = e_stub;
    return 0;
}
```

```
    *library_handle = handle;

    char msg[256];
    int length = snprintf(msg, sizeof(msg), "библиотека '%s' загружена\n", library_path);
    write(STDOUT_FILENO, msg, length);
    return 1;
}

static void command_0(const char **current_lib_ptr, void **library,
                      sin_integral_func **sin_impl_ptr, e_func **e_impl_ptr)
{

    const char *new_lib_path;
    if (strcmp(*current_lib_ptr, LIB1_PATH) == 0)
    {
        new_lib_path = LIB2_PATH;
    }
    else
    {
        new_lib_path = LIB1_PATH;
    }

    if (load_library(new_lib_path, library, sin_impl_ptr, e_impl_ptr))
    {
        *current_lib_ptr = new_lib_path;
    }
```

```
}

static void command_1(sin_integral_func sin_integral_impl, char **saveptr)
{
    char *arg1 = strtok_r(NULL, " \t\n", saveptr);
    char *arg2 = strtok_r(NULL, " \t\n", saveptr);
    char *arg3 = strtok_r(NULL, " \t\n", saveptr);

    int len = 0;
    char buffer[256];

    if (arg1 && arg2 && arg3)
    {
        float a = atof(arg1);
        float b = atof(arg2);
        float e_step = atof(arg3);

        if (e_step <= 0)
        {
            len = snprintf(buffer, sizeof(buffer), "ошибка значения\n");
        }
        else
        {
            float result = sin_integral_impl(a, b, e_step);
            len = snprintf(buffer, sizeof(buffer),
                           "интеграл sin(x) результат = %.6f\n", result);
        }
```

```
    }
    else
    {
        len = snprintf(buffer, sizeof(buffer), "недостаточно аргументов\n");
    }
    write(STDOUT_FILENO, buffer, len);
}

static void command_2(e_func e_impl, char **saveptr)
{
    char *arg1 = strtok_r(NULL, " \t\n", saveptr);

    int len = 0;
    char buffer[256];

    if (arg1)
    {
        int x = atoi(arg1);

        if (x <= 0)
        {
            len = snprintf(buffer, sizeof(buffer), "ошибка значения\n");
        }
        else
        {
            float result = e_impl(x);
            len = snprintf(buffer, sizeof(buffer),
```

```
                    "e результат = %.10f\n", result);
        }
    }
    else
    {
        len = snprintf(buffer, sizeof(buffer), "недостаточно аргументов\n");
    }
    write(STDOUT_FILENO, buffer, len);
}

int main(void)
{
    void *library_handle = NULL;
    sin_integral_func *sin_integral_impl = NULL;
    e_func *e_impl = NULL;
    const char *current_lib_path = LIB1_PATH;

    if (!load_library(current_lib_path, &library_handle, &sin_integral_impl, &e_impl))
    {
        write(STDERR_FILENO, "не удалось загрузить библиотеку\n", 61);
        return 1;
    }

    const char *prompt =
        "Динамическая загрузка\n"
        " 0 (switch) / 1 a b e / 2 x / q\n> ";
```

```
write(STDOUT_FILENO, prompt, strlen(prompt));

int bytes_read = 0;
char buffer[BUFFER_SIZE];

while ((bytes_read = read(STDIN_FILENO, buffer, BUFFER_SIZE - 1)) > 0)
{
    buffer[bytes_read] = '\0';
    for (int i = 0; i < bytes_read; i++)
    {
        if (buffer[i] == '\n' || buffer[i] == '\r')
        {
            buffer[i] = '\0';
            break;
        }
    }

    char *saveptr;
    char *token = strtok_r(buffer, " \t\n", &saveptr);

    if (token == NULL)
    {
        write(STDOUT_FILENO, "> ", 2);
        continue;
    }

    if (strcmp(token, "q") == 0)
```

```
    break;

int cmd = atoi(token);

switch (cmd)
{
case 0:
{
    command_0(&current_lib_path, &library_handle, &sin_integral_impl, &e_impl);
    break;
}
case 1:
{
    command_1(sin_integral_impl, &saveptr);
    break;
}
case 2:
{
    command_2(e_impl, &saveptr);
    break;
}
default:
{
    const char msg[] = "неизвестная команда\n";
    write(STDOUT_FILENO, msg, strlen(msg));
}
}
```

```
        write(STDOUT_FILENO, "> ", 2);
    }

    if (library_handle != NULL)
    {
        dlclose(library_handle);
    }
    write(STDOUT_FILENO, "Exit\n", 5);
    return 0;
}
```

libmath1.c

```
#include "mathlib.h"
#include <math.h>

#ifdef _WIN32
#define EXPORT __declspec(dllexport)
#else
#define EXPORT
#endif

EXPORT float sin_integral(float a, float b, float e_step)
{
    if (e_step <= 0.0f || b <= a)
        return 0.0f;

    float integral = 0.0f;
    float x = a;
```

```
    while (x < b)
    {
        float step = (x + e_step < b) ? e_step : (b - x);
        float mid_point = x + step / 2.0f;
        integral += step * sinf(mid_point);
        x += step;
    }

    return integral;
}

EXPORT float e(int x)
{
    if (x <= 0)
        return 0.0f;
    return powf(1.0f + 1.0f / (float)x, (float)x);
}
```

libmath2.c

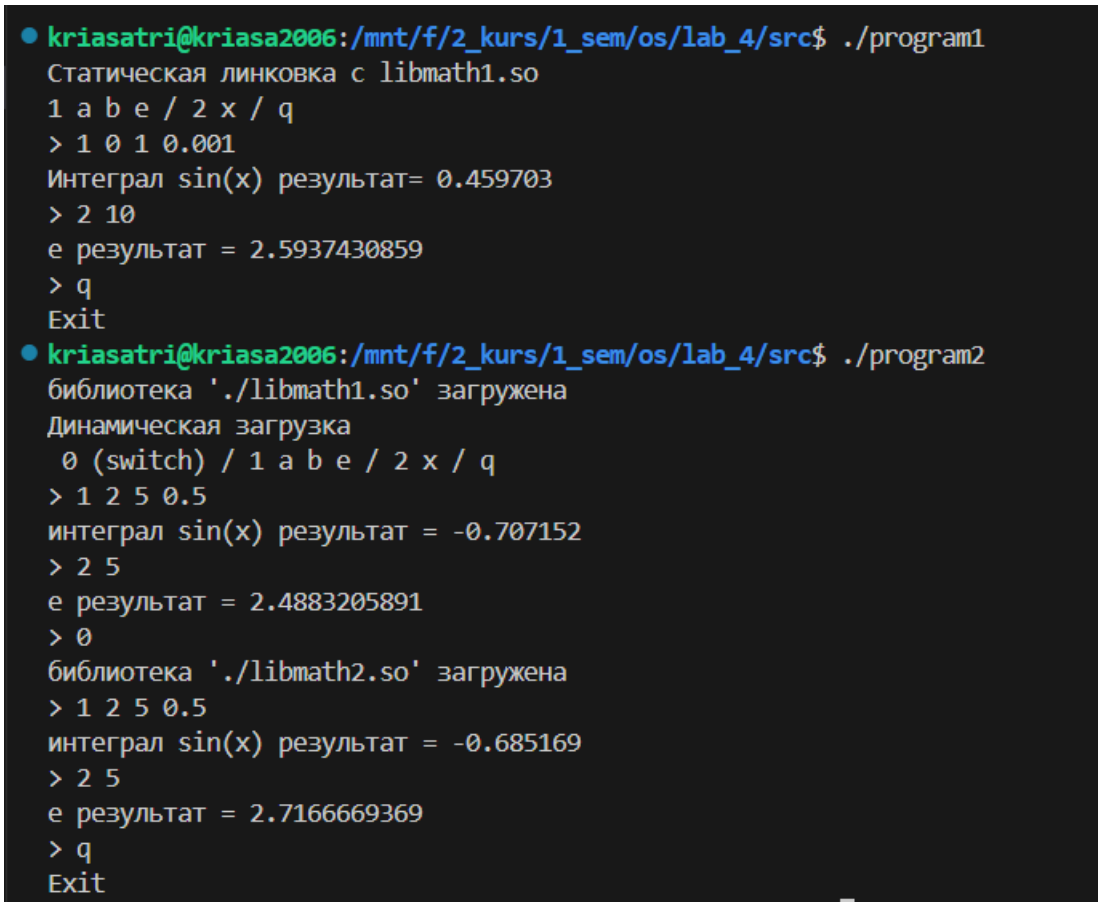
```
#include "mathlib.h"
#include <math.h>

#ifdef _WIN32
#define EXPORT __declspec(dllexport)
#else
#define EXPORT
#endif
```

```
EXPORT float sin_integral(float a, float b, float e_step)
{
    if (e_step <= 0.0f || b <= a)
        return 0.0f;

    float integral = 0.0f;
    float x = a;
    float prev_sin = sinf(x);

    while (x < b)
    {
        float step = (x + e_step < b) ? e_step : (b - x);
        float next_x = x + step;
        float next_sin = sinf(next_x);
        integral += (prev_sin + next_sin) * step / 2.0f;

        x = next_x;
        prev_sin = next_sin;
    }

    return integral;
}

EXPORT float e(int x)
{
    if (x <= 0)
```

```
        return 0.0f;

    float result = 1.0f;
    float factorial = 1.0f;

    for (int n = 1; n <= x; n++)
    {
        factorial *= n;
        result += 1.0f / factorial;
    }

    return result;
}
```

Протокол работы программы

Тесты



Strace

kriasatri@kriasa2006:/mnt/f/2_kurs/1_sem/os/lab_4/src$ strace -f ./program1

execve("./program1", ["./program1"], 0x7ffd5d88fcb8 /* 36 vars */) = 0

brk(NULL) = 0x5d29f4961000

mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x78dae8591000

access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v3/libmath1.so",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v2/libmath1.so",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "./libmath1.so", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st_mode=S_IFREG|0777, st_size=15616, ...}) = 0

getcwd("/mnt/f/2_kurs/1_sem/os/lab_4/src", 128) = 33

mmap(NULL, 16416, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x78dae858c000

mmap(0x78dae858d000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x78dae858d000

mmap(0x78dae858e000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x78dae858e000

mmap(0x78dae858f000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x78dae858f000

close(3) = 0

openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v3/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v2/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "./libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

fstat(3, {st_mode=S_IFREG|0644, st_size=41691, ...}) = 0

mmap(NULL, 41691, PROT_READ, MAP_PRIVATE, 3, 0) = 0x78dae8581000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0\0"..., 832) = 832

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x78dae8200000

mmap(0x78dae8228000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x78dae8228000

mmap(0x78dae83b0000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000) = 0x78dae83b0000

mmap(0x78dae83ff000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x78dae83ff000

mmap(0x78dae8405000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x78dae8405000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.so.6", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st_mode=S_IFREG|0644, st_size=952616, ...}) = 0

mmap(NULL, 950296, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x78dae8498000

mmap(0x78dae84a8000, 520192, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x10000) = 0x78dae84a8000

mmap(0x78dae8527000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8f000) = 0x78dae8527000

mmap(0x78dae857f000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe7000) = 0x78dae857f000

close(3) = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x78dae8495000

arch_prctl(ARCH_SET_FS, 0x78dae8495740) = 0

set_tid_address(0x78dae8495a10) = 158413

set_robust_list(0x78dae8495a20, 24) = 0

rseq(0x78dae8496060, 0x20, 0, 0x53053053) = 0

mprotect(0x78dae83ff000, 16384, PROT_READ) = 0

mprotect(0x78dae857f000, 4096, PROT_READ) = 0

mprotect(0x78dae858f000, 4096, PROT_READ) = 0

mprotect(0x5d29d4be8000, 4096, PROT_READ) = 0

mprotect(0x78dae85c9000, 8192, PROT_READ) = 0

prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0

munmap(0x78dae8581000, 41691) = 0

write(1,
"\320\241\321\202\320\260\321\202\320\270\321\207\320\265\321\201\320\272\320\260
\321\217 \320\273\320\270\320\275\320\272\320"..., 75Статическая линковка с
libmath1.so

1 a b e / 2 x / q

>) = 75

read(0, 1 0 1 0.0001

"1 0 1 0.0001\n", 4095) = 13

write(1, "\320\230\320\275\321\202\320\265\320\263\321\200\320\260\320\273 sin(x)
\321\200\320\265\320\267\321\203"..., 53Интеграл sin(x) результат= 0.459653

) = 53

write(1, "> ", 2>) = 2

read(0, 2 5

"2 5\n", 4095) = 4

write(1, "e
\321\200\320\265\320\267\321\203\320\273\321\214\321\202\320\260\321\202 =
2.4883205"..., 36e результат = 2.4883205891

) = 36

write(1, "> ", 2>) = 2

read(0, q

"q\n", 4095) = 2

write(1, "Exit\n\0", 6Exit

) = 6

exit_group(0) = ?

+++ exited with 0 +++

---

kriasatri@kriasa2006:/mnt/f/2_kurs/1_sem/os/lab_4/src$ strace -f ./program2

execve("./program2", ["./program2"], 0x7ffeff18c698 /* 36 vars */) = 0

brk(NULL) = 0x5bc97c11a000

mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x73b53eb0e000

access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

fstat(3, {st_mode=S_IFREG|0644, st_size=41691, ...}) = 0

mmap(NULL, 41691, PROT_READ, MAP_PRIVATE, 3, 0) = 0x73b53eb03000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0"..., 832) =
832

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) =
784

fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) =
784

mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x73b53e800000

mmap(0x73b53e828000, 1605632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x73b53e828000

mmap(0x73b53e9b0000, 323584, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000) = 0x73b53e9b0000

mmap(0x73b53e9ff000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x73b53e9ff000

mmap(0x73b53ea05000, 52624, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x73b53ea05000

close(3) = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x73b53eb00000

arch_prctl(ARCH_SET_FS, 0x73b53eb00740) = 0

set_tid_address(0x73b53eb00a10) = 159233

set_robust_list(0x73b53eb00a20, 24) = 0

rseq(0x73b53eb01060, 0x20, 0, 0x53053053) = 0

mprotect(0x73b53e9ff000, 16384, PROT_READ) = 0

mprotect(0x5bc960832000, 4096, PROT_READ) = 0

mprotect(0x73b53eb46000, 8192, PROT_READ) = 0

prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0

munmap(0x73b53eb03000, 41691) = 0

getrandom("\x02\x7e\xc3\xff\x53\xab\x3d\xed", 8, GRND_NONBLOCK) = 8

brk(NULL) = 0x5bc97c11a000

brk(0x5bc97c13b000) = 0x5bc97c13b000

openat(AT_FDCWD, "./libmath1.so", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st_mode=S_IFREG|0777, st_size=15616, ...}) = 0

getcwd("/mnt/f/2_kurs/1_sem/os/lab_4/src", 128) = 33

mmap(NULL, 16416, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x73b53eb09000

**mmap(0x73b53eb0a000, 4096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x73b53eb0a000**

**mmap(0x73b53eb0b000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x73b53eb0b000**

mmap(0x73b53eb0c000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x73b53eb0c000

close(3) = 0

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

fstat(3, {st_mode=S_IFREG|0644, st_size=41691, ...}) = 0

mmap(NULL, 41691, PROT_READ, MAP_PRIVATE, 3, 0) = 0x73b53eaf5000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.so.6", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st_mode=S_IFREG|0644, st_size=952616, ...}) = 0

mmap(NULL, 950296, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x73b53e717000

mmap(0x73b53e727000, 520192, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x10000) = 0x73b53e727000

mmap(0x73b53e7a6000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8f000) = 0x73b53e7a6000

mmap(0x73b53e7fe000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe7000) = 0x73b53e7fe000

close(3) = 0

mprotect(0x73b53e7fe000, 4096, PROT_READ) = 0

mprotect(0x73b53eb0c000, 4096, PROT_READ) = 0

munmap(0x73b53eaf5000, 41691) = 0

write(1, "\320\261\320\270\320\261\320\273\320\270\320\276\321\202\320\265\320\272\320\260 './libmath1"..., 56библиотека './libmath1.so' загружена

) = 56

write(1, "\320\224\320\270\320\275\320\260\320\274\320\270\321\207\320\265\321\201\320\272\320\260\321\217 \320\267\320\260\320\263\321"..., 76Динамическая загрузка

0 (switch) / 1 a b e / 2 x / q

>) = 76

read(0, 1 0 1 0.0001

"1 0 1 0.0001\n", 4095) = 13

write(1, "\320\270\320\275\321\202\320\265\320\263\321\200\320\260\320\273 sin(x) \321\200\320\265\320\267\321\203"..., 54интеграл sin(x) результат = 0.459653

) = 54

write(1, "> ", 2>) = 2

read(0, 0

"0\n", 4095) = 2

munmap(0x73b53eb09000, 16416) = 0

munmap(0x73b53e717000, 950296) = 0

openat(AT_FDCWD, "./libmath2.so", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st_mode=S_IFREG|0777, st_size=15568, ...}) = 0

getcwd("/mnt/f/2_kurs/1_sem/os/lab_4/src", 128) = 33

mmap(NULL, 16408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x73b53eb09000

mmap(0x73b53eb0a000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x73b53eb0a000

mmap(0x73b53eb0b000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x73b53eb0b000

mmap(0x73b53eb0c000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x73b53eb0c000

close(3) = 0

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

fstat(3, {st_mode=S_IFREG|0644, st_size=41691, ...}) = 0

mmap(NULL, 41691, PROT_READ, MAP_PRIVATE, 3, 0) = 0x73b53eaf5000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.so.6", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st_mode=S_IFREG|0644, st_size=952616, ...}) = 0

mmap(NULL, 950296, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x73b53e717000

mmap(0x73b53e727000, 520192, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x10000) = 0x73b53e727000

mmap(0x73b53e7a6000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8f000) = 0x73b53e7a6000

mmap(0x73b53e7fe000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe7000) = 0x73b53e7fe000

close(3)   = 0

mprotect(0x73b53e7fe000, 4096, PROT_READ) = 0

mprotect(0x73b53eb0c000, 4096, PROT_READ) = 0

munmap(0x73b53eaf5000, 41691)   = 0

write(1, "\320\261\320\270\320\261\320\273\320\270\320\276\321\202\320\265\320\272\320\260 './libmath2"..., 56библиотека './libmath2.so' загружена

) = 56

write(1, "> ", 2>)   = 2

read(0, 1 0 1 0.0001

"1 0 1 0.0001\n", 4095)   = 13

write(1, "\320\270\320\275\321\202\320\265\320\263\321\200\320\260\320\273 sin(x) \321\200\320\265\320\267\321\203"..., 54интеграл sin(x) результат = 0.459653

) = 54

write(1, "> ", 2>)   = 2

read(0, 2 5

"2 5\n", 4095)   = 4

```
write(1, "e
\321\200\320\265\320\267\321\203\320\273\321\214\321\202\320\260\321\202 =
2.7166669"..., 36е результат = 2.7166669369

) = 36

write(1, "> ", 2> )                    = 2

read(0, q

"q\n", 4095)                  = 2

munmap(0x73b53eb09000, 16408)          = 0

munmap(0x73b53e717000, 950296)         = 0

write(1, "Exit\n", 5Exit

)                = 5

exit_group(0)                      = ?

+++ exited with 0 +++
```

Вывод

В ходе выполнения лабораторной работы были успешно разработаны две динамические библиотеки с альтернативными реализациями (для расчёта интеграла sin(x) и числа e) в среде Linux. Продемонстрировано их использование двумя ключевыми методами связывания: статическая компоновка (link-time), динамическая подгрузка во время исполнения (runtime loading) с использованием системного интерфейса dlfcn.h, обеспечившая возможность выбора функционала без перезапуска программы.