

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-209БВ-24

Студент: Кобзев К. А.

Преподаватель: Миронов Е.С.

Оценка: \_\_\_\_\_

Дата: 20.11.25

Москва, 2025

## Постановка задачи

### Вариант 18.

Требуется создать динамические библиотеки, которые реализуют заданный вариантом 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 информацию полученные на этапе компиляции;
- Тестовая программа (программа №2), которая загружает библиотеки, используя только их относительные пути 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.

18 вариант)

3	Подсчёт количества простых чисел на отрезке [A, B] (A, B - натуральные)	Int PrimeCount(int A, int B)	Наивный алгоритм. Проверить делимость текущего числа на все предыдущие числа.	Решето Эратосфена
6	Расчет значения числа e(основание натурального логарифма)	Float E(int x)	$(1 + 1/x)^x$	Сумма ряда по n от 0 до x, где элементы ряда равны: $(1/(n!))$

# Общий метод и алгоритм решения

Использованные инструменты и функции:

- `void* dlsym(void* handle, const char* symbol);` Получает адрес функции или переменной (символа) с именем `symbol` из библиотеки, на которую указывает `handle`.
- `int dlclose(void* handle);` Выгружает динамическую библиотеку, на которую указывает `handle`, из адресного пространства процесса.

Использованные системные вызовы:

- `g++` — компилятор для языка C++.
- флаг `-dynamiclib / -shared` — указывает компилятору создать динамическую (разделяемую) библиотеку из исходного кода.
- флаг `-l<имя>` — указывает компоновщику связать (слинковать) программу с указанной библиотекой.
- флаг `-Wl,-rpath=<путь>` — задает путь для поиска динамических библиотек во время выполнения программы (используется для неявной компоновки).
- `void* dlopen(const char* filename, int flag);` — загружает динамическую библиотеку, указанную в `filename`, в адресное пространство процесса. Возвращает уникальный идентификатор (`handle`) для использования в последующих вызовах.
- `void* dlsym(void* handle, const char* symbol);` — получает адрес функции или переменной (символа) с именем `symbol` из библиотеки, на которую указывает `handle`.
- `int dlclose(void* handle);` — выгружает динамическую библиотеку, на которую указывает `handle`, из адресного пространства процесса.

Алгоритм решения

## 1. Создание динамических библиотек (`libimpl1.so` и `libimpl2.so`)

1.1. Создается общий заголовочный файл (`contracts.h`), в котором объявляются сигнатуры всех реализуемых функций. Для совместимости с `dlsym` объявления оборачиваются в `extern "C"`, чтобы предотвратить искажение (`mangling`) имен функций компилятором C++.

1.2. Создается файл `lib1.cpp`, который включает `contracts.h` и содержит первую реализацию функций (например, "наивный" алгоритм поиска простых чисел).

1.3. Создается файл `lib2.cpp`, который также включает `contracts.h` и содержит вторую, альтернативную реализацию функций (например, "Решето Эратосфена").

1.4. Каждый из файлов (`lib1.cpp`, `lib2.cpp`) компилируется в отдельную динамическую библиотеку с помощью флага `-dynamiclib` (или `-shared`).

## 2. Программа №1 (Неявная компоновка)

2.1. Создается файл `main1.cpp`, который включает заголовочный файл `contracts.h` для доступа к объявлениям функций.

2.2. В коде программы функции из библиотеки вызываются напрямую по их именам (`PrimeCount(...)`, `E(...)`).

2.3. На этапе компиляции программа компоуется с одной из библиотек (например, с `libimpl1.so`) с помощью флагов `-L` (указывает путь к библиотеке) и `-l` (указывает имя библиотеки). Дополнительно используется флаг `-rpath` для того, чтобы исполняемый файл мог найти библиотеку во время запуска.

2.4. Программа организует пользовательский ввод для вызова функций 1 и 2, но не имеет возможности сменить реализацию, так как жестко привязана к одной библиотеке на этапе сборки.

## 3. Программа №2 (Явная динамическая загрузка)

3.1. Создается файл `main2.cpp`, который включает заголовочный файл `<dlfcn.h>`.

3.2. В коде определяются указатели на функции, типы которых в точности соответствуют сигнатурам из contracts.h.

3.3. При запуске программа с помощью dlopen() загружает первую библиотеку (libimpl1.so) и сохраняет ее идентификатор (handle).

3.4. С помощью dlsym() программа получает адреса необходимых функций из загруженной библиотеки и присваивает их соответствующим указателям.

3.5. В основном цикле программа обрабатывает команды пользователя:

- При вводе команд 1 или 2 вызов соответствующей функции происходит через ранее полученные указатели.
- При вводе команды 0 (смена реализации) программа выполняет следующие действия:
  - Вызывает dlclose() для текущей библиотеки, чтобы выгрузить ее из памяти.
  - Вызывает dlopen() с именем второй библиотеки (libimpl2.so).
  - Снова использует dlsym(), чтобы обновить указатели на функции, которые теперь будут ссылаться на код из новой библиотеки.

3.6. Перед завершением работы программа вызывает dlclose(), чтобы корректно освободить ресурсы последней использованной библиотеки.

## Код программы

### contracts/contracts.h

```
#ifndef CONTRACTS_H
```

```
#define CONTRACTS_H
```

```
#ifdef __cplusplus
```

```
extern "C"
```

```
{
```

```
#endif
```

```
    int PrimeCount(int A, int B); // Подсчёт количества простых чисел в [A, B]
```

```
    float E(int x);              // Расчёт значения числа e
```

```
#ifdef __cplusplus
```

```
}
```

```
#endif
```

```
#endif
```

### lib1/lib1.cpp

```
#include "../contracts/contracts.h"
```

```
#include <cmath>
```

// Является ли число простым

```
bool isPrime(int n)
{
    if (n <= 1)
    {
        return false;
    }
    for (int i = 2; i * i <= n; i++)
    {
        if (n % i == 0)
        {
            return false;
        }
    }
    return true;
}
```

// Наивный подсчет

```
int PrimeCount(int A, int B)
{
    int count = 0;
    for (int i = A; i <= B; ++i)
    {
        if (isPrime(i))
        {
            count++;
        }
    }
    return count;
}
```

```
}
```

```
// Вычисление e через предел
```

```
float E(int x)
```

```
{
```

```
    if (x <= 0)
```

```
    {
```

```
        return 1.0f;
```

```
    }
```

```
    return pow(1.0f + 1.0f / x, x);
```

```
}
```

```
lib2/lib2.cpp
```

```
#include "../contracts/contracts.h"
```

```
#include <vector>
```

```
// Подсчет простых чисел с помощью Решета Эратосфена
```

```
int PrimeCount(int A, int B)
```

```
{
```

```
    if (B < 2)
```

```
    {
```

```
        return 0;
```

```
    }
```

```
    std::vector<bool> is_prime(B + 1, true);
```

```
    is_prime[0] = is_prime[1] = false;
```

```
    for (int p = 2; p * p <= B; p++)
```

```
    {
```

```
        if (is_prime[p])
```

```
        {
```

```
            for (int i = p * p; i <= B; i += p)
```

```
                is_prime[i] = false;
```

```

    }
}

int count = 0;
for (int i = A; i <= B; i++)
{
    if (i >= 0 && i <= B && is_prime[i])
    {
        count++;
    }
}
return count;
}

```

// Вычисление e через сумму ряда  $1/n!$

```

float E(int x)
{
    float sum = 0.0f;
    long double factorial = 1.0;
    for (int i = 0; i <= x; i++)
    {
        if (i == 0)
        {
            factorial = 1.0;
        }
        else
        {
            factorial *= i;
        }
        sum += 1.0f / factorial;
    }
}

```

```

    }

    return sum;
}

program1/main1.cpp

#include "../contracts/contracts.h"

#include <iostream>

using namespace std;

void print_usage()
{
    cout << "Использование программы:" << endl;
    cout << "1 <A> <B> - Подсчёт простых чисел в [A, B]" << endl;
    cout << "2 <X> - Вычислить e с точностью X" << endl;
    cout << "exit - Выход" << endl;
}

```

```

int main()
{
    cout << "Программа 1 (Статическая компоновка с реализацией 1)" << endl;
    print_usage();

    int command;
    while (cin >> command)
    {
        if (command == 1)
        {
            int a, b;

            cin >> a >> b;

            cout << "Результат: " << PrimeCount(a, b) << endl;
        }
    }
}

```



```

        else if (command == 2)
        {
            int x;

            cin >> x;

            cout << "Результат: " << E(x) << endl;
        }
        else
        {
            cout << "Неизвестная команда" << endl;

            print_usage();
        }
    }

    return 0;
}

```

### **program2/main2.cpp**

```

#include <iostream>

#include <dlfcn.h> // Для dlopen, dlsym, dlclose

using namespace std;

typedef int (*PrimeCount_t)(int, int);

typedef float (*E_t)(int);

void print_usage()
{
    cout << "Использование программы:" << endl;

    cout << "0      - Переключение реализации" << endl;

    cout << "1 <A> <B>  - Подсчёт простых чисел в [A, B]" << endl;

    cout << "2 <X>    - Вычислить e с точностью X" << endl;

    cout << "exit     - Выход" << endl;
}

```

```

int main()
{
    void *handle = nullptr;

    PrimeCount_t primeCountFunc = nullptr;

    E_t eFunc = nullptr;

    int current_lib = 1;

#ifdef __APPLE__
    const char *lib_ext = ".dylib";
#elif __linux__
    const char *lib_ext = ".so";
#endif

    string lib1_name = "./libimpl1";
    lib1_name += lib_ext;

    handle = dlopen(lib1_name.c_str(), RTLD_LAZY);

    if (!handle)
    {
        cerr << "Не могу открыть библиотеку: " << dlerror() << endl;

        return 1;
    }

    primeCountFunc = (PrimeCount_t)dlsym(handle, "PrimeCount");
    eFunc = (E_t)dlsym(handle, "E");

    if (!primeCountFunc || !eFunc)
    {
        cerr << "Невозможно загрузить символы: " << dlerror() << endl;

        dlclose(handle);
    }
}

```

```

        return 1;
    }

    cout << "Программа 2 (Динамическая загрузка). Текущая реализация: " << current_lib <<
endl;

    print_usage();

    int command;
    while (cin >> command)
    {
        if (command == 0)
        {
            dlclose(handle);
            current_lib = (current_lib == 1) ? 2 : 1;

            string next_lib_name = (current_lib == 1) ? "./libimpl1" : "./libimpl2";
            next_lib_name += lib_ext;

            handle = dlopen(next_lib_name.c_str(), RTLD_LAZY);
            if (!handle)
            {
                cerr << "Не могу открыть библиотеку: " << dlerror() << endl;
                return 1;
            }

            primeCountFunc = (PrimeCount_t)dlsym(handle, "PrimeCount");
            eFunc = (E_t)dlsym(handle, "E");
            if (!primeCountFunc || !eFunc)
            {
                cerr << "Невозможно загрузить символы: " << dlerror() << endl;

```

```

        dlclose(handle);

        return 1;
    }

    cout << "Перешёл к реализации " << current_lib << endl;
}

else if (command == 1)
{
    int a, b;

    cin >> a >> b;

    if (primeCountFunc)
    {
        cout << "Результат: " << primeCountFunc(a, b) << endl;
    }

    else
    {
        cerr << "Функция PrimeCount не загружена." << endl;
    }
}

else if (command == 2)
{
    int x;

    cin >> x;

    if (eFunc)
    {
        cout << "Результат: " << eFunc(x) << endl;
    }

    else
    {
        cerr << "Функция E не загружена." << endl;
    }
}

```

```

    }

    else if (cin.eof() || cin.fail())

    {

        cout << "Выход." << endl;

        break;

    }

    else

    {

        cout << "Неизвестная команда" << endl;

        print_usage();

        cin.clear();

    }

}

dlclose(handle);

return 0;

}

```

## **Makefile**

CXX = g++

CXXFLAGS = -Wall -std=c++11 -fPIC

BUILD\_DIR = build

UNAME\_S := \$(shell uname -s)

ifeq (\$(UNAME\_S),Darwin)

SHARED\_FLAG = -dynamiclib -install\_name @rpath/\$(@F)

LIB\_EXT = dylib

RPATH\_FLAG = -Wl,-rpath,@executable\_path

else

SHARED\_FLAG = -shared -Wl,-soname,\$(@F)

LIB\_EXT = so

RPATH\_FLAG = -Wl,-rpath,\$(BUILD\_DIR)

endif

LIB1\_TARGET\_NAME = libimpl1.\$(LIB\_EXT)

LIB2\_TARGET\_NAME = libimpl2.\$(LIB\_EXT)

LIB1 = \$(BUILD\_DIR)/\$(LIB1\_TARGET\_NAME)

LIB2 = \$(BUILD\_DIR)/\$(LIB2\_TARGET\_NAME)

PROG1 = \$(BUILD\_DIR)/program1

PROG2 = \$(BUILD\_DIR)/program2

all: \$(PROG1) \$(PROG2)

\$(BUILD\_DIR):

mkdir -p \$(BUILD\_DIR)

\$(LIB1): lib1/lib1.cpp contracts/contracts.h | \$(BUILD\_DIR)

\$(CXX) \$(CXXFLAGS) \$(SHARED\_FLAG) -o \$@ \$<

\$(LIB2): lib2/lib2.cpp contracts/contracts.h | \$(BUILD\_DIR)

\$(CXX) \$(CXXFLAGS) \$(SHARED\_FLAG) -o \$@ \$<

\$(PROG1): program1/main1.cpp \$(LIB1) | \$(BUILD\_DIR)

\$(CXX) \$(CXXFLAGS) -o \$@ \$< -L\$(BUILD\_DIR) -limpl1 \$(RPATH\_FLAG)

\$(PROG2): program2/main2.cpp \$(LIB1) \$(LIB2) | \$(BUILD\_DIR)

\$(CXX) \$(CXXFLAGS) -o \$@ \$< -L\$(BUILD\_DIR) -limpl1 -limpl2 \$(RPATH\_FLAG)

clean:

```
rm -rf $(BUILD_DIR)
```

.PHONY: all clean

## Протокол работы программы

### Тестирование:

→ build git:(main) X ./program1

Программа 1 (Статическая компоновка с реализацией 1)

Использование программы:

1 <A> <B> - Подсчёт простых чисел в [A, B]

2 <X> - Вычислить  $e$  с точностью X

exit - Выход

1 10 100

Результат: 21

2 10

Результат: 2.59374

exit

→ build git:(main) X ./program2

Программа 2 (Динамическая загрузка). Текущая реализация: 1

Использование программы:

0 - Переключение реализации

1 <A> <B> - Подсчёт простых чисел в [A, B]

2 <X> - Вычислить  $e$  с точностью X

exit - Выход

1 1 10

Результат: 4

2 5

Результат: 2.48832

0

Перешёл к реализации 2

1 1 10

Результат: 4

2 9

Результат: 2.71828

exit

**strace**

**strace program1.txt**

211 execve("./program1", ["/program1"], 0xffffced24f98 /\* 13 vars \*/) = 0

211 brk(NULL) = 0x19e3e000

211 mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8e640000

211 faccessat(AT\_FDCWD, "/etc/ld.so.preload", R\_OK) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "/workspace/lab4/src/build/tls/aarch64/atomics/libimpl1.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 newfstatat(AT\_FDCWD, "/workspace/lab4/src/build/tls/aarch64/atomics", 0xffffc2c6bad0, 0) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "/workspace/lab4/src/build/tls/aarch64/libimpl1.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 newfstatat(AT\_FDCWD, "/workspace/lab4/src/build/tls/aarch64", 0xffffc2c6bad0, 0) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "/workspace/lab4/src/build/tls/atomics/libimpl1.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 newfstatat(AT\_FDCWD, "/workspace/lab4/src/build/tls/atomics", 0xffffc2c6bad0, 0) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "/workspace/lab4/src/build/tls/libimpl1.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 newfstatat(AT\_FDCWD, "/workspace/lab4/src/build/tls", 0xffffc2c6bad0, 0) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "/workspace/lab4/src/build/aarch64/atomics/libimpl1.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 newfstatat(AT\_FDCWD, "/workspace/lab4/src/build/aarch64/atomics", 0xffffc2c6bad0, 0) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "/workspace/lab4/src/build/aarch64/libimpl1.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 newfstatat(AT\_FDCWD, "/workspace/lab4/src/build/aarch64", 0xffffc2c6bad0, 0) = -1 ENOENT (No such file or directory)



211 openat(AT\_FDCWD, "/workspace/lab4/src/build/atomics/libimpl1.so",  
O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 newfstatat(AT\_FDCWD, "/workspace/lab4/src/build/atomics", 0xffffc2c6bad0, 0) = -1  
ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "/workspace/lab4/src/build/libimpl1.so", O\_RDONLY|O\_CLOEXEC)  
= 3

211 read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\0\0\0\0\0\0\0", 832) = 832

211 newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=69536, ...}, AT\_EMPTY\_PATH) = 0

211 mmap(NULL, 196656, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) =  
0xffff8e5d6000

211 mmap(0xffff8e5e0000, 131120, PROT\_READ|PROT\_EXEC,  
MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff8e5e0000

211 munmap(0xffff8e5d6000, 40960) = 0

211 munmap(0xffff8e601000, 20528) = 0

211 mprotect(0xffff8e5e1000, 122880, PROT\_NONE) = 0

211 mmap(0xffff8e5ff000, 8192, PROT\_READ|PROT\_WRITE,  
MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xf000) = 0xffff8e5ff000

211 close(3) = 0

211 openat(AT\_FDCWD, "/workspace/lab4/src/build/libstdc++.so.6",  
O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/tls/aarch64/atomics/libstdc++.so.6",  
O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/tls/aarch64/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC) = -1  
ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/tls/atomics/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC) = -1  
ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/tls/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT  
(No such file or directory)

211 openat(AT\_FDCWD, "build/aarch64/atomics/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC)  
= -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/aarch64/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC) = -1  
ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/atomics/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC) = -1  
ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT  
(No such file or directory)

211 openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

211 newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=25959, ...}, AT\_EMPTY\_PATH) = 0

211 mmap(NULL, 25959, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0xffff8e639000

211 close(3) = 0

211 openat(AT\_FDCWD, "/usr/local/lib64/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC) = 3

211 read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\3\0\267\0\1\0\0\0\0\0\0\0\0"..., 832) = 832

211 newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=2436000, ...}, AT\_EMPTY\_PATH) = 0

211 mmap(NULL, 2583688, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8e200000

211 mmap(0xffff8e200000, 2518152, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff8e200000

211 munmap(0xffff8e467000, 64648) = 0

211 mprotect(0xffff8e445000, 65536, PROT\_NONE) = 0

211 mmap(0xffff8e455000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x245000) = 0xffff8e455000

211 mmap(0xffff8e463000, 15496, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0xffff8e463000

211 close(3) = 0

211 openat(AT\_FDCWD, "/workspace/lab4/src/build/libm.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/tls/aarch64/atomics/libm.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/tls/aarch64/libm.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/tls/atomics/libm.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/tls/libm.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/aarch64/atomics/libm.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/aarch64/libm.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/atomics/libm.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/libm.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "/lib/aarch64-linux-gnu/libm.so.6", O\_RDONLY|O\_CLOEXEC) = 3

211 read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\0\0\0\0"..., 832) = 832

211 newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=591960, ...}, AT\_EMPTY\_PATH) = 0

211 mmap(NULL, 655472, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8e53f000



211 close(3) = 0

211 openat(AT\_FDCWD, "/workspace/lab4/src/build/libc.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/tls/aarch64/atomics/libc.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/tls/aarch64/libc.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/tls/atomics/libc.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/tls/libc.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/aarch64/atomics/libc.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/aarch64/libc.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/atomics/libc.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "build/libc.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

211 openat(AT\_FDCWD, "/lib/aarch64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3

211 read(3, "\177ELF\2\1\13\0\0\0\0\0\0\3\0\267\0\1\0\0\0000y\2\0\0\0\0\0"..., 832) = 832

211 newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=1651408, ...}, AT\_EMPTY\_PATH) = 0

211 mmap(NULL, 1826912, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8e041000

211 mmap(0xffff8e050000, 1761376, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff8e050000

211 munmap(0xffff8e041000, 61440) = 0

211 munmap(0xffff8e1ff000, 96) = 0

211 mprotect(0xffff8e1d7000, 86016, PROT\_NONE) = 0

211 mmap(0xffff8e1ec000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x18c000) = 0xffff8e1ec000

211 mmap(0xffff8e1f2000, 49248, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0xffff8e1f2000

211 close(3) = 0

211 mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8e637000

211 mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8e635000

211 set\_tid\_address(0xffff8e6350f0) = 211

211 set\_robust\_list(0xffff8e635100, 24) = 0  
211 rseq(0xffff8e635740, 0x20, 0, 0xd428bc00) = 0  
211 mprotect(0xffff8e1ec000, 16384, PROT\_READ) = 0  
211 mprotect(0xffff8e52f000, 4096, PROT\_READ) = 0  
211 mprotect(0xffff8e5cf000, 4096, PROT\_READ) = 0  
211 mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8e633000  
211 mprotect(0xffff8e455000, 45056, PROT\_READ) = 0  
211 mprotect(0xffff8e5ff000, 4096, PROT\_READ) = 0  
211 mprotect(0x41f000, 4096, PROT\_READ) = 0  
211 mprotect(0xffff8e645000, 8192, PROT\_READ) = 0  
211 prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0  
211 munmap(0xffff8e639000, 25959) = 0  
211 futex(0xffff8e4637ec, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0  
211 getrandom("\x06\x68\xec\x53\x75\xca\xfl\x8b", 8, GRND\_NONBLOCK) = 8  
211 brk(NULL) = 0x19e3e000  
211 brk(0x19e5f000) = 0x19e5f000  
211 newfstatat(1, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}, AT\_EMPTY\_PATH) = 0  
211 write(1, "\320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260 1 (\320\241\321\202\320\260\321\202\320\270"..., 95) = 95  
211 write(1, "\320\230\321\201\320\277\320\276\320\273\321\214\320\267\320\276\320\262\320\260\320\275\320\270\320\265 \320\277\321\200\320"..., 47) = 47  
211 write(1, "1 <A> <B> - \320\237\320\276\320\264\321\201\321\207\321\221\321\202 \320\277\321"..., 65) = 65  
211 write(1, "2 <X> - \320\222\321\213\321\207\320\270\321\201\320\273\320\270\321\202\321\214"..., 59) = 59  
211 write(1, "exit - \320\222\321\213\321\205\320\276\320\264\n", 25) = 25  
211 newfstatat(0, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}, AT\_EMPTY\_PATH) = 0  
211 read(0, "1 10 100\n", 1024) = 9  
211 write(1, "\320\240\320\265\320\267\321\203\320\273\321\214\321\202\320\260\321\202: 21\n", 23) = 23  
211 read(0, "2 10\n", 1024) = 5

211 write(1, "\320\240\320\265\320\267\321\203\320\273\321\214\321\202\320\260\321\202:2.59374\n", 28) = 28

211 read(0, "exit\n", 1024) = 5

211 lseek(0, -5, SEEK\_CUR) = -1 ESPIPE (Illegal seek)

211 exit\_group(0) = ?

211 +++ exited with 0 +++

### **strace program2.txt**

216 execve("./program2", [ "./program2" ], 0xffffed959a28 /\* 13 vars \*/) = 0

216 brk(NULL) = 0x40337000

216 mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffffface3b000

216 faccessat(AT\_FDCWD, "/etc/ld.so.preload", R\_OK) = -1 ENOENT (No such file or directory)

216 openat(AT\_FDCWD, "/workspace/lab4/src/build/tls/aarch64/atomics/libimpl1.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

216 newfstatat(AT\_FDCWD, "/workspace/lab4/src/build/tls/aarch64/atomics", 0xffffe49c2ad0, 0) = -1 ENOENT (No such file or directory)

216 openat(AT\_FDCWD, "/workspace/lab4/src/build/tls/aarch64/libimpl1.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

216 newfstatat(AT\_FDCWD, "/workspace/lab4/src/build/tls/aarch64", 0xffffe49c2ad0, 0) = -1 ENOENT (No such file or directory)

216 openat(AT\_FDCWD, "/workspace/lab4/src/build/tls/atomics/libimpl1.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

216 newfstatat(AT\_FDCWD, "/workspace/lab4/src/build/tls/atomics", 0xffffe49c2ad0, 0) = -1 ENOENT (No such file or directory)

216 openat(AT\_FDCWD, "/workspace/lab4/src/build/tls/libimpl1.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

216 newfstatat(AT\_FDCWD, "/workspace/lab4/src/build/tls", 0xffffe49c2ad0, 0) = -1 ENOENT (No such file or directory)

216 openat(AT\_FDCWD, "/workspace/lab4/src/build/aarch64/atomics/libimpl1.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

216 newfstatat(AT\_FDCWD, "/workspace/lab4/src/build/aarch64/atomics", 0xffffe49c2ad0, 0) = -1 ENOENT (No such file or directory)

216 openat(AT\_FDCWD, "/workspace/lab4/src/build/aarch64/libimpl1.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

216 newfstatat(AT\_FDCWD, "/workspace/lab4/src/build/aarch64", 0xffffe49c2ad0, 0) = -1 ENOENT (No such file or directory)

216 openat(AT\_FDCWD, "/workspace/lab4/src/build/atomics/libimpl1.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

```

216 newfstatat(AT_FDCWD, "/workspace/lab4/src/build/atomics", 0xffffe49c2ad0, 0) = -1
ENOENT (No such file or directory)

216 openat(AT_FDCWD, "/workspace/lab4/src/build/libimpl1.so", O_RDONLY|O_CLOEXEC)
= 3

216 read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

216 newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=69536, ...}, AT_EMPTY_PATH) = 0

216 mmap(NULL, 196656, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0xffffacdd1000

216 mmap(0xffffacde0000, 131120, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0) = 0xffffacde0000

216 munmap(0xffffacdd1000, 61440) = 0

216 munmap(0xfffface01000, 48) = 0

216 mprotect(0xffffacde1000, 122880, PROT_NONE) = 0

216 mmap(0xffffacdff000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xf000) = 0xffffacdff000

216 close(3) = 0

216 openat(AT_FDCWD, "/workspace/lab4/src/build/libimpl2.so", O_RDONLY|O_CLOEXEC)
= 3

216 read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

216 newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=74400, ...}, AT_EMPTY_PATH) = 0

216 mmap(NULL, 197032, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0xffffacdaf000

216 mmap(0xffffacdb0000, 131496, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0) = 0xffffacdb0000

216 munmap(0xffffacdaf000, 4096) = 0

216 munmap(0xffffacdd1000, 57768) = 0

216 mprotect(0xffffacdb4000, 110592, PROT_NONE) = 0

216 mmap(0xffffacdcf000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xf000) = 0xffffacdcf000

216 close(3) = 0

216 openat(AT_FDCWD, "/workspace/lab4/src/build/libstdc++.so.6",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

216 openat(AT_FDCWD, "build/tls/aarch64/atomics/libstdc++.so.6",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

216 openat(AT_FDCWD, "build/tls/aarch64/libstdc++.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)

216 openat(AT_FDCWD, "build/tls/atomics/libstdc++.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)

```





216 openat(AT\_FDCWD, "build/aarch64/atomics/libm.so.6", O\_RDONLY|O\_CLOEXEC) = -1  
ENOENT (No such file or directory)

216 openat(AT\_FDCWD, "build/aarch64/libm.so.6", O\_RDONLY|O\_CLOEXEC) = -1  
ENOENT (No such file or directory)

216 openat(AT\_FDCWD, "build/atomics/libm.so.6", O\_RDONLY|O\_CLOEXEC) = -1  
ENOENT (No such file or directory)

216 openat(AT\_FDCWD, "build/libm.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No  
such file or directory)

216 openat(AT\_FDCWD, "/lib/aarch64-linux-gnu/libm.so.6", O\_RDONLY|O\_CLOEXEC) = 3

216 read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

216 newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=591960, ...}, AT\_EMPTY\_PATH) = 0

216 mmap(NULL, 655472, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) =  
0xffffacd0f000

216 mmap(0xffffacd10000, 589936, PROT\_READ|PROT\_EXEC,  
MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffffacd10000

216 munmap(0xffffacd0f000, 4096) = 0

216 munmap(0xffffacda1000, 57456) = 0

216 mprotect(0xffffacd90000, 61440, PROT\_NONE) = 0

216 mmap(0xffffacd9f000, 8192, PROT\_READ|PROT\_WRITE,  
MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x8f000) = 0xffffacd9f000

216 close(3) = 0

216 openat(AT\_FDCWD, "/workspace/lab4/src/build/libgcc\_s.so.1",  
O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

216 openat(AT\_FDCWD, "build/tls/aarch64/atomics/libgcc\_s.so.1",  
O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

216 openat(AT\_FDCWD, "build/tls/aarch64/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = -1  
ENOENT (No such file or directory)

216 openat(AT\_FDCWD, "build/tls/atomics/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = -1  
ENOENT (No such file or directory)

216 openat(AT\_FDCWD, "build/tls/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT  
(No such file or directory)

216 openat(AT\_FDCWD, "build/aarch64/atomics/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) =  
-1 ENOENT (No such file or directory)

216 openat(AT\_FDCWD, "build/aarch64/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = -1  
ENOENT (No such file or directory)

216 openat(AT\_FDCWD, "build/atomics/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = -1  
ENOENT (No such file or directory)

216 openat(AT\_FDCWD, "build/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No  
such file or directory)

```

216 openat(AT_FDCWD, "/usr/local/lib64/libgcc_s.so.1", O_RDONLY|O_CLOEXEC) = 3
216 read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\0\0\0\0\0"..., 832) = 832
216 newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=726416, ...}, AT_EMPTY_PATH) = 0
216 mmap(NULL, 263104, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0xffffacccf000
216 mmap(0xffffaccd0000, 197568, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0) = 0xffffaccd0000
216 munmap(0xffffacccf000, 4096) = 0
216 munmap(0xffffacd01000, 58304) = 0
216 mprotect(0xffffaccef000, 65536, PROT_NONE) = 0
216 mmap(0xffffaccff000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1f000) = 0xffffaccff000
216 close(3) = 0
216 openat(AT_FDCWD, "/workspace/lab4/src/build/libc.so.6", O_RDONLY|O_CLOEXEC) = -
1 ENOENT (No such file or directory)
216 openat(AT_FDCWD, "build/tls/aarch64/atomics/libc.so.6", O_RDONLY|O_CLOEXEC) = -
1 ENOENT (No such file or directory)
216 openat(AT_FDCWD, "build/tls/aarch64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
216 openat(AT_FDCWD, "build/tls/atomics/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
216 openat(AT_FDCWD, "build/tls/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
216 openat(AT_FDCWD, "build/aarch64/atomics/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
216 openat(AT_FDCWD, "build/aarch64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
216 openat(AT_FDCWD, "build/atomics/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
216 openat(AT_FDCWD, "build/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
216 openat(AT_FDCWD, "/lib/aarch64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
216 read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\3\0\267\0\1\0\0\0000y\2\0\0\0\0"..., 832) = 832
216 newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=1651408, ...}, AT_EMPTY_PATH) = 0
216 mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xfffface32000
216 mmap(NULL, 1826912, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0xffffac841000

```

```

216 mmap(0xfffffac850000, 1761376, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0) = 0xfffffac850000

216 munmap(0xfffffac841000, 61440) = 0

216 munmap(0xfffffac9ff000, 96) = 0

216 mprotect(0xfffffac9d7000, 86016, PROT_NONE) = 0

216 mmap(0xfffffac9ec000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x18c000) = 0xfffffac9ec000

216 mmap(0xfffffac9f2000, 49248, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0xfffffac9f2000

216 close(3) = 0

216 mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xffffface30000

216 set_tid_address(0xffffface300f0) = 216

216 set_robust_list(0xffffface30100, 24) = 0

216 rseq(0xffffface30740, 0x20, 0, 0xd428bc00) = 0

216 mprotect(0xfffffac9ec000, 16384, PROT_READ) = 0

216 mprotect(0xfffffaccff000, 4096, PROT_READ) = 0

216 mprotect(0xfffffacd9f000, 4096, PROT_READ) = 0

216 mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xffffface2e000

216 mprotect(0xfffffacc55000, 45056, PROT_READ) = 0

216 mprotect(0xfffffacdcf000, 4096, PROT_READ) = 0

216 mprotect(0xfffffacdff000, 4096, PROT_READ) = 0

216 mprotect(0x41f000, 4096, PROT_READ) = 0

216 mprotect(0xffffface40000, 8192, PROT_READ) = 0

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

216 munmap(0xffffface34000, 25959) = 0

216 futex(0xfffffacc637ec, FUTEX_WAKE_PRIVATE, 2147483647) = 0

216 getrandom("\x7b\x84\xa6\xee\xfb\x08\x25\x13", 8, GRND_NONBLOCK) = 8

216 brk(NULL) = 0x40337000

216 brk(0x40358000) = 0x40358000

216 openat(AT_FDCWD, "./libimpl1.so", O_RDONLY|O_CLOEXEC) = 3

216 read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\0\0\0\0\0\0"..., 832) = 832

216 newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=69536, ...}, AT_EMPTY_PATH) = 0

```

```

216 close(3) = 0

216 newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...},
AT_EMPTY_PATH) = 0

216 write(1, "\320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260 2
(\320\224\320\270\320\275\320\260\320\274"..., 105) = 105

216 write(1,
"\320\230\321\201\320\277\320\276\320\273\321\214\320\267\320\276\320\262\320\260\320\275\320\2
70\320\265 \320\277\321\200\320"..., 47) = 47

216 write(1, "0 -
\320\237\320\265\321\200\320\265\320\272\320\273\321\216\321\207\320\265"..., 60) = 60

216 write(1, "1 <A> <B> - \320\237\320\276\320\264\321\201\321\207\321\221\321\202
\320\277\321"..., 65) = 65

216 write(1, "2 <X> -
\320\222\321\213\321\207\320\270\321\201\320\273\320\270\321\202\321\214"..., 59) = 59

216 write(1, "exit - \320\222\321\213\321\205\320\276\320\264\n", 25) = 25

216 newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...},
AT_EMPTY_PATH) = 0

216 read(0, "1 10 100\n", 1024) = 9

216 write(1, "\320\240\320\265\320\267\321\203\320\273\321\214\321\202\320\260\321\202:
21\n", 23) = 23

216 read(0, "2 10\n", 1024) = 5

216 write(1, "\320\240\320\265\320\267\321\203\320\273\321\214\321\202\320\260\321\202:
2.59374\n", 28) = 28

216 read(0, "0\n", 1024) = 2

216 openat(AT_FDCWD, "/libimpl2.so", O_RDONLY|O_CLOEXEC) = 3

216 read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\0\0\0\0\0\0"..., 832) = 832

216 newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=74400, ...}, AT_EMPTY_PATH) = 0

216 close(3) = 0

216 write(1, "\320\237\320\265\321\200\320\265\321\210\321\221\320\273 \320\272
\321\200\320\265\320\260\320\273\320\270\320\267\320\260"..., 41) = 41

216 read(0, "1 10 100\n", 1024) = 9

216 write(1, "\320\240\320\265\320\267\321\203\320\273\321\214\321\202\320\260\321\202:
21\n", 23) = 23

216 read(0, "2 10\n", 1024) = 5

216 write(1, "\320\240\320\265\320\267\321\203\320\273\321\214\321\202\320\260\321\202:
2.71828\n", 28) = 28

216 read(0, "exit\n", 1024) = 5

216 lseek(0, -5, SEEK_CUR) = -1 ESPIPE (Illegal seek)

```

```
216 exit_group(0)          = ?
```

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

## Вывод

В результате выполнения лабораторной работы были разработаны и отлажены программы, демонстрирующие два основных способа работы с динамическими библиотеками.

Была создана динамическая библиотека (.so) с двумя различными реализациями заданных функций и две тестовые программы. Первая программа связывалась с библиотекой на этапе компиляции. Анализ вывода утилиты strace подтвердил, что динамический компоновщик операционной системы автоматически загружает необходимую библиотеку и все её зависимости при старте программы.

Вторая программа продемонстрировала явную динамическую загрузку во время исполнения. Для этого был использован программный интерфейс dlfcn.h, в частности функции dlopen(), dlsym() и dlclose(). Данный подход позволил реализовать функционал переключения между двумя различными реализациями функций непосредственно во время работы программы по команде пользователя, что также было подтверждено трассировкой системных вызовов.