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-215Б-23

Студент: Лапенко К.А.

Преподаватель: Миронов Е.С.

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: 24.02.24

Москва, 2024

**Постановка задачи**

**Вариант 19.**

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 ее 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 ее выполнения.

19 вариант. 3, 7

| **№** | **Описание** | **Сигнатура** | **Реализация 1** | **Реализация 2** |
| --- | --- | --- | --- | --- |
| 3 | Подсчет количества простых чисел на отрезке [A, B] (A, B - натуральные) | Int PrimeCount(intA, int B) | Наивный алгоритм. Проверить делимость текущего числа на все предыдущие числа. | Решето Эратосфена |
| 7 | Подсчет площади плоской геометрической фигуры по двум сторонам | Float Square(float A, float B) | Фигура прямоугольник | Фигура прямоугольный треугольник |

**Общий метод и алгоритм решения**

**Использованные системные вызовы:**

* pid\_t fork(void); – Создает новый процесс путем копирования текущего процесса. Новый процесс называется дочерним, а исходный процесс называется родительским.
* \*\*int execl(const char *path, const char arg, ...);* – Заменяет текущий образ процесса новым образом, загружаемым из исполняемого файла, указанного в path
* *void dlopen(const char filename, int flag);* – Открывает динамическую библиотеку, указанную в filename, и возвращает дескриптор, который может быть использован для получения адресов функций из этой библиотеки. Для загрузки библиотек liblibrary1.so и liblibrary2.so.
* \**void* dlsym(void *handle, const char symbol);* – Возвращает адрес функции или переменной, указанной в symbol, из динамической библиотеки, открытой с помощью dlopen. Для получения указателей на функции PrimeCount и Square из загруженной библиотеки.
* \**int dlclose(void handle);* – Закрывает динамическую библиотеку, открытую с помощью dlopen, и освобождает связанные с ней ресурсы.
* *char dlerror(void);*\* – Возвращает строку, описывающую последнюю ошибку, произошедшую при вызове функций dlopen, dlsym или dlclose.

### Архитектура и алгоритм работы проекта:

#### 1. Заголовочный файл (library1.h):

* Определяет интерфейс для функций PrimeCount и Square.
* Использует макросы #ifdef \_\_cplusplus и extern "C" для обеспечения совместимости с C и C++.
* Подключается в program1.cpp и program2.cpp для использования объявленных функций.

#### 2. Реализации функций:

* prime\_naive.cpp:
  + Реализует функцию PrimeCount для подсчета простых чисел в диапазоне от A до B с использованием наивного алгоритма.
  + Алгоритм:
    1. Перебирает все числа от A до B.
    2. Для каждого числа проверяет, является ли оно простым, путем деления на все числа от 2 до квадратного корня из числа.
    3. Если число простое, увеличивает счетчик.
    4. Возвращает количество простых чисел.
* prime\_erat.cpp:
  + Реализует функцию PrimeCount с использованием алгоритма "Решето Эратосфена".
  + Алгоритм:
    1. Создает массив isPrime, где isPrime[i] указывает, является ли число i простым.
    2. Инициализирует все элементы массива как true, кроме 0 и 1.
    3. Перебирает числа от 2 до квадратного корня из B и помечает кратные числа как не простые. Подсчитывает количество простых чисел в диапазоне от A до B.
    4. Возвращает количество простых чисел.
* area\_triangle.cpp:
  + Реализует функцию Square для вычисления площади треугольника по формуле: 0.5 \* A \* B.
  + Возвращает результат вычисления.
* area\_rect.cpp:
  + Реализует функцию Square для вычисления площади прямоугольника по формуле: A \* B.
  + Возвращает результат вычисления.

#### 3. Описание основных программ:

* program1.cpp:
  + Использует функции PrimeCount и Square из статически подключенной библиотеки.
  + Алгоритм:
    1. Входит в бесконечный цикл, где пользователь может выбрать команду:
       - Команда 1: Запрашивает у пользователя диапазон A и B, затем вызывает PrimeCount и выводит количество простых чисел.
       - Команда 2: Запрашивает у пользователя стороны A и B, затем вызывает Squareи выводит площадь.
       - Любая другая команда: завершает программу.
* program2.cpp:
  + Динамически загружает библиотеки (liblibrary1.so и liblibrary2.so) и переключается между ними.
  + Алгоритм:
    1. Загружает первую библиотеку (liblibrary1.so) с помощью dlopen.
    2. Получает указатели на функции PrimeCount и Square с помощью dlsym.
    3. Входит в бесконечный цикл, где пользователь может выбрать команду:
       - Команда 0: Переключает между библиотеками liblibrary1.so и liblibrary2.so, закрывая текущую библиотеку и загружая новую.
       - Команда 1: Запрашивает у пользователя диапазон A и B, затем вызывает PrimeCount и выводит количество простых чисел.
       - Команда 2: Запрашивает у пользователя стороны A и B, затем вызывает Squareи выводит площадь.
       - Команда 4: Завершает программу.

После завершения цикла программа закрывает текущую библиотеку с помощью dlclose.

**Код программы**

**program1.cpp**

#include <iostream>

#include "library1.h"

int main() {

int choice;

while (true) {

std::cout << "Enter a command (1 or 2): ";

std::cin >> choice;

if (choice == 1) {

int A, B;

std::cout << "Enter A and B to count prime numbers: ";

std::cin >> A >> B;

std::cout << "Number of primes: " << PrimeCount(A, B) << std::endl;

} else if (choice == 2) {

float A, B;

std::cout << "Enter sides A and B: ";

std::cin >> A >> B;

std::cout << "Area: " << Square(A, B) << std::endl;

} else {

break;

}

}

return 0;

}

**program2.cpp**

#include <iostream>

#include <dlfcn.h>

typedef int (\*PrimeCountFunc)(int, int);

typedef float (\*SquareFunc)(float, float);

int main() {

void\* hLib = nullptr;

PrimeCountFunc PrimeCount = nullptr;

SquareFunc Square = nullptr;

int currentLibrary = 1;

hLib = dlopen("./liblibrary1.so", RTLD\_LAZY);

if (!hLib) {

std::cerr << "Failed to load library1.so: " << dlerror() << std::endl;

return 1;

}

PrimeCount = (PrimeCountFunc)dlsym(hLib, "PrimeCount");

Square = (SquareFunc)dlsym(hLib, "Square");

int choice;

while (true) {

std::cout << "Enter a command (0, 1, or 2) or 4 for exit: ";

std::cin >> choice;

std::cout << choice;

if (choice == 0) {

dlclose(hLib);

currentLibrary = (currentLibrary == 1) ? 2 : 1;

hLib = dlopen(currentLibrary == 1 ? "./liblibrary1.so" : "./liblibrary2.so", RTLD\_LAZY);

if (!hLib) {

std::cerr << "Failed to load the library: " << dlerror() << std::endl;

return 1;

}

PrimeCount = (PrimeCountFunc)dlsym(hLib, "PrimeCount");

Square = (SquareFunc)dlsym(hLib, "Square");

std::cout << "Switched to " << (currentLibrary == 1 ? "library1.so" : "library2.so") << std::endl;

} else if (choice == 1) {

int A, B;

std::cout << "Enter A and B to count prime numbers: ";

std::cin >> A >> B;

std::cout << "Number of primes: " << PrimeCount(A, B) << std::endl;

} else if (choice == 2) {

float A, B;

std::cout << "Enter sides A and B: ";

std::cin >> A >> B;

std::cout << "Area: " << Square(A, B) << std::endl;

} else if (choice == 4) {

break;

}

}

dlclose(hLib);

return 0;

}

**area\_triangle.cpp**

extern "C" float Square(float A, float B) {

return 0.5f \* A \* B;

}

**area\_rect.cpp**

extern "C" float Square(float A, float B) {

return A \* B;

}

**prime\_erat.cpp**

#include <vector>

extern "C" int PrimeCount(int A, int B) {

if (B < 2) return 0;

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

isPrime[0] = isPrime[1] = false;

for (int i = 2; i \* i <= B; i++) {

if (isPrime[i]) {

for (int j = i \* i; j <= B; j += i) {

isPrime[j] = false;

}

}

}

int count = 0;

for (int i = A; i <= B; i++) {

if (isPrime[i]) count++;

}

return count;

}

**prime\_naive.cpp**

extern "C" int PrimeCount(int A, int B) {

int count = 0;

for (int i = A; i <= B; i++) {

bool isPrime = true;

if (i < 2) continue;

for (int j = 2; j \* j <= i; j++) {

if (i % j == 0) {

isPrime = false;

break;

}

}

if (isPrime) count++;

}

return count;

}

**library1.h**

#ifndef LIBRARY1\_H

#define LIBRARY1\_H

#ifdef \_\_cplusplus

extern "C" {

#endif

int PrimeCount(int A, int B);

float Square(float A, float B);

#ifdef \_\_cplusplus

}

#endif

#endif

**Протокол работы программы**

Примеры вызовов открывающих/изменяющих библиотеки:

* openat(AT\_FDCWD, "./liblibrary1.so", O\_RDONLY|O\_CLOEXEC) = 3 – открытие файла библиотеки
* mmap(0xffff8b3a0000, 69664, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff8b3a0000 – отображение файла в память
* munmap(0xffff8b39e000, 8192) = 0 – освобождение отображенной памяти
* mprotect(0xffff8b3a1000, 61440, PROT\_NONE) = 0 – изменение прав доступа к памяти

**Strace для program1.cpp:**

execve("./program1", ["./program1"], 0xffffd38b0800 /\* 11 vars \*/) = 0

brk(NULL) = 0xaaaafd070000

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff9926e000

faccessat(AT\_FDCWD, "/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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=12067, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 12067, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0xffff9926b000

close(3) = 0

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

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=2190752, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 2332704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff99000000

mmap(0xffff99000000, 2267168, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff99000000

munmap(0xffff9922a000, 63520) = 0

mprotect(0xffff9920a000, 61440, PROT\_NONE) = 0

mmap(0xffff99219000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x209000) = 0xffff99219000

mmap(0xffff99227000, 10272, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0xffff99227000

close(3) = 0

openat(AT\_FDCWD, "/lib/aarch64-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\267\0\1\0\0\0\340u\2\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=1637400, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 1805928, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff98e47000

mmap(0xffff98e50000, 1740392, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff98e50000

munmap(0xffff98e47000, 36864) = 0

munmap(0xffff98ff9000, 28264) = 0

mprotect(0xffff98fd8000, 61440, PROT\_NONE) = 0

mmap(0xffff98fe7000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x187000) = 0xffff98fe7000

mmap(0xffff98fed000, 48744, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0xffff98fed000

close(3) = 0

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

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=551064, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 680048, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff98da9000

mmap(0xffff98db0000, 614512, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff98db0000

munmap(0xffff98da9000, 28672) = 0

munmap(0xffff98e47000, 32880) = 0

mprotect(0xffff98e36000, 61440, PROT\_NONE) = 0

mmap(0xffff98e45000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x85000) = 0xffff98e45000

close(3) = 0

openat(AT\_FDCWD, "/lib/aarch64-linux-gnu/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = 3

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=84296, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 213704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff98d7b000

mmap(0xffff98d80000, 148168, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff98d80000

munmap(0xffff98d7b000, 20480) = 0

munmap(0xffff98da5000, 41672) = 0

mprotect(0xffff98d94000, 61440, PROT\_NONE) = 0

mmap(0xffff98da3000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x13000) = 0xffff98da3000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff99269000

set\_tid\_address(0xffff99269af0) = 109

set\_robust\_list(0xffff99269b00, 24) = 0

rseq(0xffff9926a1c0, 0x20, 0, 0xd428bc00) = 0

mprotect(0xffff98fe7000, 16384, PROT\_READ) = 0

mprotect(0xffff98da3000, 4096, PROT\_READ) = 0

mprotect(0xffff98e45000, 4096, PROT\_READ) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff99267000

mprotect(0xffff99219000, 45056, PROT\_READ) = 0

mprotect(0xaaaae78f1000, 4096, PROT\_READ) = 0

mprotect(0xffff99274000, 8192, PROT\_READ) = 0

prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

munmap(0xffff9926b000, 12067) = 0

getrandom("\xc1\xcb\x22\x4d\x63\x9b\x6e\x80", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0xaaaafd070000

brk(0xaaaafd091000) = 0xaaaafd091000

futex(0xffff992277a4, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

newfstatat(1, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}, AT\_EMPTY\_PATH) = 0

write(1, "Enter a command (1 or 2): ", 26) = 26

newfstatat(0, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}, AT\_EMPTY\_PATH) = 0

read(0, "1\n", 1024) = 2

write(1, "Enter A and B to count prime num"..., 38) = 38

read(0, "1 10\n", 1024) = 5

write(1, "Number of primes: 4\n", 20) = 20

write(1, "Enter a command (1 or 2): ", 26) = 26

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

write(1, "Enter sides A and B: ", 21) = 21

read(0, "1 414\n", 1024) = 6

write(1, "Area: 414\n", 10) = 10

write(1, "Enter a command (1 or 2): ", 26) = 26

read(0, 0xaaaafd0822c0, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

--- SIGINT {si\_signo=SIGINT, si\_code=SI\_KERNEL} ---

+++ killed by SIGINT +++

**Strace для program2.cpp:**

execve("./program2", ["./program2"], 0xffffd40a2670 /\* 11 vars \*/) = 0

brk(NULL) = 0xaaab06d93000

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8b8a9000

faccessat(AT\_FDCWD, "/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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=12067, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 12067, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0xffff8b8a6000

close(3) = 0

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

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=2190752, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 2332704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8b63b000

mmap(0xffff8b640000, 2267168, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff8b640000

munmap(0xffff8b63b000, 20480) = 0

munmap(0xffff8b86a000, 43040) = 0

mprotect(0xffff8b84a000, 61440, PROT\_NONE) = 0

mmap(0xffff8b859000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x209000) = 0xffff8b859000

mmap(0xffff8b867000, 10272, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0xffff8b867000

close(3) = 0

openat(AT\_FDCWD, "/lib/aarch64-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\267\0\1\0\0\0\340u\2\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=1637400, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 1805928, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8b487000

mmap(0xffff8b490000, 1740392, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff8b490000

munmap(0xffff8b487000, 36864) = 0

munmap(0xffff8b639000, 28264) = 0

mprotect(0xffff8b618000, 61440, PROT\_NONE) = 0

mmap(0xffff8b627000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x187000) = 0xffff8b627000

mmap(0xffff8b62d000, 48744, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0xffff8b62d000

close(3) = 0

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

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=551064, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 680048, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8b3e9000

mmap(0xffff8b3f0000, 614512, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff8b3f0000

munmap(0xffff8b3e9000, 28672) = 0

munmap(0xffff8b487000, 32880) = 0

mprotect(0xffff8b476000, 61440, PROT\_NONE) = 0

mmap(0xffff8b485000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x85000) = 0xffff8b485000

close(3) = 0

openat(AT\_FDCWD, "/lib/aarch64-linux-gnu/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = 3

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=84296, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 213704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8b3bb000

mmap(0xffff8b3c0000, 148168, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff8b3c0000

munmap(0xffff8b3bb000, 20480) = 0

munmap(0xffff8b3e5000, 41672) = 0

mprotect(0xffff8b3d4000, 61440, PROT\_NONE) = 0

mmap(0xffff8b3e3000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x13000) = 0xffff8b3e3000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8b8a4000

set\_tid\_address(0xffff8b8a4af0) = 117

set\_robust\_list(0xffff8b8a4b00, 24) = 0

rseq(0xffff8b8a51c0, 0x20, 0, 0xd428bc00) = 0

mprotect(0xffff8b627000, 16384, PROT\_READ) = 0

mprotect(0xffff8b3e3000, 4096, PROT\_READ) = 0

mprotect(0xffff8b485000, 4096, PROT\_READ) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8b8a2000

mprotect(0xffff8b859000, 45056, PROT\_READ) = 0

mprotect(0xaaaadba61000, 4096, PROT\_READ) = 0

mprotect(0xffff8b8af000, 8192, PROT\_READ) = 0

prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

munmap(0xffff8b8a6000, 12067) = 0

getrandom("\x18\x95\xe5\xbf\x80\xb5\x35\x10", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0xaaab06d93000

brk(0xaaab06db4000) = 0xaaab06db4000

futex(0xffff8b8677a4, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

openat(AT\_FDCWD, "./liblibrary1.so", O\_RDONLY|O\_CLOEXEC) = 3

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

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=7928, ...}, AT\_EMPTY\_PATH) = 0

getcwd("/workspace/lab4/src", 128) = 20

mmap(NULL, 135200, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8b39e000

mmap(0xffff8b3a0000, 69664, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff8b3a0000

munmap(0xffff8b39e000, 8192) = 0

munmap(0xffff8b3b2000, 53280) = 0

mprotect(0xffff8b3a1000, 61440, PROT\_NONE) = 0

mmap(0xffff8b3b0000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff8b3b0000

close(3) = 0

mprotect(0xffff8b3b0000, 4096, PROT\_READ) = 0

newfstatat(1, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}, AT\_EMPTY\_PATH) = 0

write(1, "Enter a command (0, 1, or 2) or "..., 44) = 44

newfstatat(0, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}, AT\_EMPTY\_PATH) = 0

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

write(1, "2Enter sides A and B: ", 22) = 22

read(0, "1 10\n", 1024) = 5

write(1, "Area: 10\n", 9) = 9

write(1, "Enter a command (0, 1, or 2) or "..., 44) = 44

read(0, "1\n", 1024) = 2

write(1, "1Enter A and B to count prime nu"..., 39) = 39

read(0, "1 1000\n", 1024) = 7

write(1, "Number of primes: 168\n", 22) = 22

write(1, "Enter a command (0, 1, or 2) or "..., 44) = 44

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

munmap(0xffff8b3a0000, 69664) = 0

openat(AT\_FDCWD, "./liblibrary2.so", O\_RDONLY|O\_CLOEXEC) = 3

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

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=26320, ...}, AT\_EMPTY\_PATH) = 0

getcwd("/workspace/lab4/src", 128) = 20

mmap(NULL, 147928, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8b39b000

mmap(0xffff8b3a0000, 82392, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff8b3a0000

munmap(0xffff8b39b000, 20480) = 0

munmap(0xffff8b3b5000, 41432) = 0

mprotect(0xffff8b3a4000, 61440, PROT\_NONE) = 0

mmap(0xffff8b3b3000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0xffff8b3b3000

close(3) = 0

mprotect(0xffff8b3b3000, 4096, PROT\_READ) = 0

write(1, "0Switched to library2.so\n", 25) = 25

write(1, "Enter a command (0, 1, or 2) or "..., 44) = 44

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

munmap(0xffff8b3a0000, 82392) = 0

openat(AT\_FDCWD, "./liblibrary1.so", O\_RDONLY|O\_CLOEXEC) = 3

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

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=7928, ...}, AT\_EMPTY\_PATH) = 0

getcwd("/workspace/lab4/src", 128) = 20

mmap(NULL, 135200, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8b39e000

mmap(0xffff8b3a0000, 69664, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff8b3a0000

munmap(0xffff8b39e000, 8192) = 0

munmap(0xffff8b3b2000, 53280) = 0

mprotect(0xffff8b3a1000, 61440, PROT\_NONE) = 0

mmap(0xffff8b3b0000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff8b3b0000

close(3) = 0

mprotect(0xffff8b3b0000, 4096, PROT\_READ) = 0

write(1, "0Switched to library1.so\n", 25) = 25

write(1, "Enter a command (0, 1, or 2) or "..., 44) = 44

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

munmap(0xffff8b3a0000, 69664) = 0

openat(AT\_FDCWD, "./liblibrary2.so", O\_RDONLY|O\_CLOEXEC) = 3

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

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=26320, ...}, AT\_EMPTY\_PATH) = 0

getcwd("/workspace/lab4/src", 128) = 20

mmap(NULL, 147928, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8b39b000

mmap(0xffff8b3a0000, 82392, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff8b3a0000

munmap(0xffff8b39b000, 20480) = 0

munmap(0xffff8b3b5000, 41432) = 0

mprotect(0xffff8b3a4000, 61440, PROT\_NONE) = 0

mmap(0xffff8b3b3000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0xffff8b3b3000

close(3) = 0

mprotect(0xffff8b3b3000, 4096, PROT\_READ) = 0

write(1, "0Switched to library2.so\n", 25) = 25

write(1, "Enter a command (0, 1, or 2) or "..., 44) = 44

read(0, "1\n", 1024) = 2

write(1, "1Enter A and B to count prime nu"..., 39) = 39

read(0, "1 1000\n", 1024) = 7

write(1, "Number of primes: 168\n", 22) = 22

write(1, "Enter a command (0, 1, or 2) or "..., 44) = 44

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

write(1, "2Enter sides A and B: ", 22) = 22

read(0, "1 2134124\n", 1024) = 10

write(1, "Area: 1.06706e+06\n", 18) = 18

write(1, "Enter a command (0, 1, or 2) or "..., 44) = 44

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

write(1, "2Enter sides A and B: ", 22) = 22

read(0, "2 3\n", 1024) = 4

write(1, "Area: 3\n", 8) = 8

write(1, "Enter a command (0, 1, or 2) or "..., 44) = 44

read(0, "4\n", 1024) = 2

munmap(0xffff8b3a0000, 82392) = 0

write(1, "4", 1) = 1

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

exit\_group(0) = ?

+++ exited with 0 +++

**Вывод**

Было интересно разобраться с динамическими и статическими библиотеками. Поняла, как и в какой ситуации следует использовать каждую из них.