

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: Кармишен Е.С.

Преподаватель: Миронов Е.С. (ПМИ)

Оценка: _____

Дата: 14.02.24

Москва, 2024

Постановка задачи

Вариант 17.

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.

Функция 1.

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
---	---	------------------------------	---	-------------------

Функция 2.

5	Расчет значения числа Пи при заданной длине ряда (K)	float Pi(int K)	Ряд Лейбница	Формула Валлиса
---	--	-----------------	--------------	-----------------

Общий метод и алгоритм решения

Использованные системные вызовы:

- `void* dlopen(const char* filename, int flag);` - загружает динамическую библиотеку в память
- `int dlclose(void* handle);` - выгружает динамическую библиотеку из памяти
- `void* dlsym(void* handle, const char* symbol);` - получает адрес символа из библиотеки
- `char* dlerror(void);` - возвращает строку с описанием последней ошибки

Общий алгоритм решения:

`test_static.c`

1. Меню команд:
 - Программа показывает пользователю меню с доступными командами:
 - 1: Вычислить количество простых чисел в диапазоне.
 - 2: Вычислить число Пи с заданной точностью.
 - -1: Выйти из программы.
2. Выбор команды:
 - Пользователь вводит команду.
 - В зависимости от введенной команды, программа выполняет соответствующее действие.
3. Выполнение команд:
 - Команда 1:
 - Пользователь вводит два числа A и B, которые задают диапазон.
 - Программа вызывает функцию `CalculatePrime(A, B)`, которая возвращает количество простых чисел в этом диапазоне.
 - Результат выводится на экран, и информация о вызове функции записывается в отчет.
 - Команда 2:
 - Пользователь вводит число K, которое задает точность вычисления числа Пи.
 - Программа вызывает функцию `CalculatePi(K)`, которая возвращает значение числа Пи с заданной точностью.
 - Результат выводится на экран, и информация о вызове функции записывается в отчет.
 - Команда -1:
 - Программа завершает работу.
4. Запись отчета:
 - Каждый раз, когда вызывается одна из функций (`CalculatePrime` или `CalculatePi`), программа записывает информацию о вызове в файл отчета. Это включает имя функции, аргументы и результат.

test_dynamic.c

1. Загрузка библиотеки:
 - Программа загружает библиотеку с помощью функции `dlopen`.
 - Если загрузка прошла успешно, программа получает указатели на функции `CalculatePrime` и `CalculatePi` с помощью `dlsym`.
 - Если загрузка или получение указателей на функции не удалось, программа выводит сообщение об ошибке.
2. Меню команд:
 - Программа показывает пользователю меню с доступными командами:
 - 0: Переключиться на другую библиотеку.
 - 1: Вычислить количество простых чисел в диапазоне.
 - 2: Вычислить число Пи с заданной точностью.
 - -1: Выйти из программы.
3. Выбор команды:
 - Пользователь вводит команду.
 - В зависимости от введенной команды, программа выполняет соответствующее действие.
4. Выполнение команд:
 - Команда 0:
 - Программа закрывает текущую библиотеку с помощью `dlclose`.
 - Переключается на другую библиотеку (`libImpl1.so` или `libImpl2.so`) и загружает её.
 - Если загрузка новой библиотеки прошла успешно, программа выводит сообщение о успешном переключении.
 - Команда 1:
 - Пользователь вводит два числа A и B, которые задают диапазон.
 - Программа вызывает функцию `CalculatePrime(A, B)` из текущей библиотеки.
 - Результат выводится на экран, и программа сообщает, какая библиотека была использована.
 - Команда 2:
 - Пользователь вводит число K, которое задает точность вычисления числа Пи.
 - Программа вызывает функцию `CalculatePi(K)` из текущей библиотеки.
 - Результат выводится на экран, и программа сообщает, какая библиотека была использована.
 - Команда -1:
 - Программа завершает работу.
5. Завершение программы:
 - При завершении работы программа закрывает текущую библиотеку с помощью `dlclose`.

Код программы

pi_leibniz.c

```
#include <stdio.h>

// Ряд Лейбница

float CalculatePi(int K) {

float pi = 0.0;

for (int i = 0; i < K; i++) {

pi += (i % 2 == 0 ? 1.0 : -1.0) / (2 * i + 1);

}

return 4 * pi;

}
```

pi_wallis.c

```
#include <stdio.h>

// Формула Валлиса

float CalculatePi(int K) {

float pi = 1.0;

for (int i = 1; i <= K; i++) {

pi *= (4.0 * i * i) / (4.0 * i * i - 1);

}

return 2 * pi;

}
```

prime_eratosthenus.c

```
#include <stdio.h>

#include <math.h>

// Решето Эратосфена

int CalculatePrime(int A, int B) {

if (B < 2)
```

```

return 0;

int size = B + 1;

int sieve[size];

for (int i = 0; i < size; i++) sieve[i] = 1;

sieve[0] = sieve[1] = 0;

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

    if (sieve[i]) {

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

            sieve[j] = 0;

        }

    }

}

int count = 0;

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

    if (sieve[i]) count++;

}

return count;

}

```

prime_naive.c

```

#include <stdio.h>

#include <math.h>

// Наивный алгоритм

int CalculatePrime(int A, int B) {

    int count = 0;

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

        if (i < 2) continue;

```

```

int is_prime = 1;

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

    if (i % j == 0) {

        is_prime = 0;

        break;

    }

}

if (is_prime) count++;

}

return count;

}

```

test_dynamic.c

```

#include <stdio.h>

#include <stdlib.h>

#include <dlfcn.h>

typedef int (*tCalculatePrime)(int, int);

typedef float (*tCalculatePi)(int);

typedef struct {

    tCalculatePi calculatePi;

    tCalculatePrime calculatePrime;

    void* library;

} tFuncLibrary;

tFuncLibrary load_library(char* filename) {

    // Загрузка библиотеки

    tFuncLibrary result;

    result.library = dlopen(filename, RTLD_LAZY); // RTLD_LAZY - отложенная загрузка

    if (!result.library) {

```

```

fprintf(stderr, "Ошибка загрузки библиотек: %s\n", dlerror());

return result;
}

// Загрузка функций

result.calculatePrime = dlsym(result.library, "CalculatePrime");

result.calculatePi = dlsym(result.library, "CalculatePi");

if (!result.calculatePrime || !result.calculatePi) {

fprintf(stderr, "Ошибка загрузки функций из библиотеки: %s\n", dlerror());

dlclose(result.library);

result.library = NULL;

return result;
}

return result;
}

int main() {

tFuncLibrary funcLib = load_library("./libImpl1.so");

if (funcLib.library == NULL) {

return 1;
}

int lib_index = 0;

int command;

while (1) {

printf("Input program code:\n");

printf(" 0 -> Library switch\n");

printf(" 1 -> PrimeCount\n");

printf(" 2 -> Pi\n");

```



```
printf("-1 -> Exit\n");

scanf("%d", &command);

if (command == -1) {

break;

} else if (command == 0) {

dlclose(funcLib.library);

lib_index = lib_index == 0 ? 1 : 0;

funcLib = load_library(lib_index == 0 ? "./libImpl1.so" : "./libImpl2.so");

if (funcLib.library == NULL) {

continue;

}

printf("Library switched successfully!\n");

printf("Current lib: %s\n", lib_index == 0 ? "./libImpl1.so" : "./libImpl2.so");

} else if (command == 1) {

int A, B;

printf("Enter A and B: ");

scanf("%d %d", &A, &B);

printf("PrimeCount(%d, %d) = %d\n", A, B, funcLib.calculatePrime(A, B));

printf("Implementation used: %s\n", lib_index == 0 ? "./libImpl1.so" : "./libImpl2.so");

} else if (command == 2) {

int K;

printf("Enter K: ");

scanf("%d", &K);

printf("Pi(%d) = %f\n", K, funcLib.calculatePi(K));

printf("Implementation used: %s\n", lib_index == 0 ? "./libImpl1.so" : "./libImpl2.so");

} else {

printf("Invalid command\n");

}

}
```

```
dlclose(funcLib.library);

return 0;

}
```

test_static.c

```
#include <stdio.h>

// Объявление функций из библиотек

extern int CalculatePrime(int, int);

extern float CalculatePi(int);

int main() {

int command;

while (1) {

printf("Input program code:\n");

printf(" 1 -> PrimeCount\n");

printf(" 2 -> Pi\n");

printf("-1 -> Exit\n");

scanf("%d", &command);

if (command == -1) {

break;

} else if (command == 1) {

int A, B;

printf("Enter A and B: ");

scanf("%d %d", &A, &B);

printf("PrimeCount(%d, %d) = %d\n", A, B, CalculatePrime(A, B));

} else if (command == 2) {

int K;

printf("Enter K: ");
```

```
scanf("%d", &K);

printf("Pi(%d) = %f\n", K, CalculatePi(K));

} else {

printf("Invalid command\n");

}

}

return 0;

}
```

Протокол работы программы

Тестирование:

test_dynamic.c:

```
Input program code:
0 -> Library switch
1 -> PrimeCount
2 -> Pi
-1 -> Exit
1
Enter A and B: 1 10
PrimeCount(1, 10) = 4
Implementation used: ./libImpl1.so
Input program code:
0 -> Library switch
1 -> PrimeCount
2 -> Pi
-1 -> Exit
0
Library switched successfully!
Current lib: ./libImpl2.so
Input program code:
0 -> Library switch
1 -> PrimeCount
2 -> Pi
-1 -> Exit
1
Enter A and B: 1 10
PrimeCount(1, 10) = 4
Implementation used: ./libImpl2.so
Input program code:
0 -> Library switch
1 -> PrimeCount
2 -> Pi
-1 -> Exit
-1
```

test_static.c:

```
apple@MacBook-Pro-apple src % ./test_static
Input program code:
1 -> PrimeCount
2 -> Pi
```

1

```
execve("/test_static", ["/test_static"], 0x7ffc3fb39040 /* 34 vars */) = 0
brk(NULL) = 0x558b99efb000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffc24c602c0) = -1 EINVAL (Invalid argument)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7ff2e41af000
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/glibc-hwcap/x86-64-v3/libImpl1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/glibc-hwcap/x86-64-v2/libImpl1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/tls/x86_64/x86_64/libImpl1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/tls/x86_64/libImpl1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/tls/x86_64/libImpl1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/tls/x86_64/libImpl1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/x86_64/x86_64/libImpl1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/x86_64/libImpl1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/x86_64/libImpl1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/libImpl1.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
newfstatat(3, "", {st_mode=S_IFREG|0777, st_size=15648, ...}, AT_EMPTY_PATH) = 0
getcwd("/mnt/d/Projects/egorka_os/lab4/src", 128) = 35
mmap(NULL, 16432, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7ff2e41aa000
mmap(0x7ff2e41ab000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x7ff2e41ab000
mmap(0x7ff2e41ac000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7ff2e41ac000
mmap(0x7ff2e41ad000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7ff2e41ad000
close(3) = 0
openat(AT_FDCWD, "/glibc-hwcap/x86-64-v3/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/glibc-hwcap/x86-64-v2/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/tls/x86_64/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/tls/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/x86_64/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/x86_64/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
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=22187, ...}, AT_EMPTY_PATH) = 0
```

```

mmap(NULL, 22187, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7ff2e41a4000
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\0P\237\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"... , 784, 64) = 784
pread64(3, "\4\0\0\0\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0"... , 48, 848) = 48
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\17\357\204\3$\f\221\2039x\324\224\323\236S"... , 68, 896) = 68
newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2220400, ...}, AT_EMPTY_PATH) = 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, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7ff2e3f7b000
mprotect(0x7ff2e3fa3000, 2023424, PROT_NONE) = 0
mmap(0x7ff2e3fa3000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) =
0x7ff2e3fa3000
mmap(0x7ff2e4138000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7ff2e4138000
mmap(0x7ff2e4191000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) =
0x7ff2e4191000
mmap(0x7ff2e4197000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7ff2e4197000
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\0\0\0"... , 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=940560, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 942344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7ff2e3e94000
mmap(0x7ff2e3ea2000, 507904, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) = 0x7ff2e3ea2000
mmap(0x7ff2e3f1e000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8a000) = 0x7ff2e3f1e000
mmap(0x7ff2e3f79000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe4000) = 0x7ff2e3f79000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7ff2e3e91000
arch_prctl(ARCH_SET_FS, 0x7ff2e3e91740) = 0
set_tid_address(0x7ff2e3e91a10) = 57510
set_robust_list(0x7ff2e3e91a20, 24) = 0
rseq(0x7ff2e3e920e0, 0x20, 0, 0x53053053) = 0
mprotect(0x7ff2e4191000, 16384, PROT_READ) = 0
mprotect(0x7ff2e3f79000, 4096, PROT_READ) = 0
mprotect(0x7ff2e41ad000, 4096, PROT_READ) = 0
mprotect(0x558b7addf000, 4096, PROT_READ) = 0
mprotect(0x7ff2e41e9000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7ff2e41a4000, 22187) = 0
newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x7), ...}, AT_EMPTY_PATH) = 0
getrandom("\x13\xf4\xf5\x10\xee\x37\xce\x3", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x558b99efb000
brk(0x558b99f1c000) = 0x558b99f1c000
write(1, "Input program code:\n", 20) = 20
write(1, " 1 -> PrimeCount\n", 17) = 17
write(1, " 2 -> Pi\n", 9) = 9
write(1, "-1 -> Exit\n", 11) = 11
newfstatat(0, "", {st_mode=S_IFIFO|0600, st_size=0, ...}, AT_EMPTY_PATH) = 0
read(0, "\1\n0 10\n2\n100\n1\n", 4096) = 16
write(1, "Enter A and B: PrimeCount(0, 10)"..., 37) = 37
write(1, "Input program code:\n", 20) = 20
write(1, " 1 -> PrimeCount\n", 17) = 17
write(1, " 2 -> Pi\n", 9) = 9
write(1, "-1 -> Exit\n", 11) = 11
write(1, "Enter K: Pi(100) = 3.131593\n", 28) = 28

```

```

write(1, "Input program code:\n", 20) = 20
write(1, " 1 -> PrimeCount\n", 17) = 17
write(1, " 2 -> Pi\n", 9) = 9
write(1, "-1 -> Exit\n", 11) = 11
lseek(0, -1, SEEK_CUR) = -1 ESPIPE (Illegal seek)
exit_group(0) = ?
+++ exited with 0 +++

```

test_dynamic.c:

```

execve("./test_dynamic", [ "./test_dynamic" ], 0x7ffe7c042fc0 /* 34 vars */) = 0
brk(NULL) = 0x563b25b55000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffcd595b920) = -1 EINVAL (Invalid argument)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f84073e2000
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
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=22187, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 22187, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f84073dc000
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\0P\237\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
pread64(3, "\4\0\0\0 \0\0\0\5\0\0\0GNU\02\0\0\300\4\0\0\0\3\0\0\0\0\0\0"... , 48, 848) = 48
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0I\17\357\204\3$\f\221\2039x\324\224\323\236S"... , 68, 896) = 68
newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2220400, ...}, AT_EMPTY_PATH) = 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, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f84071b3000
mprotect(0x7f84071db000, 2023424, PROT_NONE) = 0
mmap(0x7f84071db000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) =
0x7f84071db000
mmap(0x7f8407370000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7f8407370000
mmap(0x7f84073c9000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) =
0x7f84073c9000
mmap(0x7f84073cf000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) =
0x7f84073cf000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f84071b0000
arch_prctl(ARCH_SET_FS, 0x7f84071b0740) = 0
set_tid_address(0x7f84071b0a10) = 62096
set_robust_list(0x7f84071b0a20, 24) = 0
rseq(0x7f84071b10e0, 0x20, 0, 0x53053053) = 0
mprotect(0x7f84073c9000, 16384, PROT_READ) = 0
mprotect(0x563b11e2d000, 4096, PROT_READ) = 0
mprotect(0x7f840741c000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f84073dc000, 22187) = 0
getrandom("\xa2\x51\x02\x5f\x26\xb4\x3a\x43", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x563b25b55000
brk(0x563b25b76000) = 0x563b25b76000
openat(AT_FDCWD, "./libImpl1.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\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
newfstatat(3, "", {st_mode=S_IFREG|0777, st_size=15648, ...}, AT_EMPTY_PATH) = 0
getcwd("/mnt/d/Projects/egorka_os/lab4/src", 128) = 35

```

```

mmap(NULL, 16432, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f84073dd000
mmap(0x7f84073de000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) =
0x7f84073de000
mmap(0x7f84073df000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f84073df000
mmap(0x7f84073e0000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) =
0x7f84073e0000
close(3) = 0
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=22187, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 22187, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f84071aa000
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"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=940560, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 942344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f84070c3000
mmap(0x7f84070d1000, 507904, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) =
0x7f84070d1000
mmap(0x7f840714d000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8a000) = 0x7f840714d000
mmap(0x7f84071a8000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe4000) =
0x7f84071a8000
close(3) = 0
mprotect(0x7f84071a8000, 4096, PROT_READ) = 0
mprotect(0x7f84073e0000, 4096, PROT_READ) = 0
munmap(0x7f84071aa000, 22187) = 0
newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x7), ...}, AT_EMPTY_PATH) = 0
write(1, "Input program code:\n", 20) = 20
write(1, " 0 -> Library switch\n", 21) = 21
write(1, " 1 -> PrimeCount\n", 17) = 17
write(1, " 2 -> Pi\n", 9) = 9
write(1, "-1 -> Exit\n", 11) = 11
newfstatat(0, "", {st_mode=S_IFIFO|0600, st_size=0, ...}, AT_EMPTY_PATH) = 0
read(0, "1\n1 10\n2\n100\n0\n1\n1 10\n2\n100\n0\n2\n"..., 4096) = 38
write(1, "Enter A and B: PrimeCount(1, 10)"..., 37) = 37
write(1, "Implementation used: ./libImpl1."..., 35) = 35
write(1, "Input program code:\n", 20) = 20
write(1, " 0 -> Library switch\n", 21) = 21
write(1, " 1 -> PrimeCount\n", 17) = 17
write(1, " 2 -> Pi\n", 9) = 9
write(1, "-1 -> Exit\n", 11) = 11
write(1, "Enter K: Pi(100) = 3.131593\n", 28) = 28
write(1, "Implementation used: ./libImpl1."..., 35) = 35
write(1, "Input program code:\n", 20) = 20
write(1, " 0 -> Library switch\n", 21) = 21
write(1, " 1 -> PrimeCount\n", 17) = 17
write(1, " 2 -> Pi\n", 9) = 9
write(1, "-1 -> Exit\n", 11) = 11
munmap(0x7f84073dd000, 16432) = 0
munmap(0x7f84070c3000, 942344) = 0
openat(AT_FDCWD, "./libImpl2.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0777, st_size=15672, ...}, AT_EMPTY_PATH) = 0
getcwd("/mnt/d/Projects/egorka_os/lab4/src", 128) = 35
mmap(NULL, 16432, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f84073dd000

```

```

0x7f84073de000 mmap(0x7f84073de000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) =
0x7f84073df000 mmap(0x7f84073df000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f84073df000
0x7f84073e0000 mmap(0x7f84073e0000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) =
0x7f84073e0000
close(3) = 0
mprotect(0x7f84073e0000, 4096, PROT_READ) = 0
write(1, "Library switched successfully!\n", 31) = 31
write(1, "Current lib: ./libImpl2.so\n", 27) = 27
write(1, "Input program code:\n", 20) = 20
write(1, " 0 -> Library switch\n", 21) = 21
write(1, " 1 -> PrimeCount\n", 17) = 17
write(1, " 2 -> Pi\n", 9) = 9
write(1, "-1 -> Exit\n", 11) = 11
write(1, "Enter A and B: PrimeCount(1, 10)"..., 37) = 37
write(1, "Implementation used: ./libImpl2."..., 35) = 35
write(1, "Input program code:\n", 20) = 20
write(1, " 0 -> Library switch\n", 21) = 21
write(1, " 1 -> PrimeCount\n", 17) = 17
write(1, " 2 -> Pi\n", 9) = 9
write(1, "-1 -> Exit\n", 11) = 11
write(1, "Enter K: Pi(100) = 3.133787\n", 28) = 28
write(1, "Implementation used: ./libImpl2."..., 35) = 35
write(1, "Input program code:\n", 20) = 20
write(1, " 0 -> Library switch\n", 21) = 21
write(1, " 1 -> PrimeCount\n", 17) = 17
write(1, " 2 -> Pi\n", 9) = 9
write(1, "-1 -> Exit\n", 11) = 11
munmap(0x7f84073dd000, 16432) = 0
openat(AT_FDCWD, "./libImpl1.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\3\0\>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0777, st_size=15648, ...}, AT_EMPTY_PATH) = 0
getcwd("/mnt/d/Projects/egorka_os/lab4/src", 128) = 35
mmap(NULL, 16432, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f84073dd000
0x7f84073de000 mmap(0x7f84073de000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) =
0x7f84073df000 mmap(0x7f84073df000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f84073df000
0x7f84073e0000 mmap(0x7f84073e0000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) =
0x7f84073e0000
close(3) = 0
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=22187, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 22187, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f84071aa000
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\3\0\>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=940560, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 942344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f84070c3000
0x7f84070d1000 mmap(0x7f84070d1000, 507904, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) =
0x7f840714d000 mmap(0x7f840714d000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8a000) = 0x7f840714d000
0x7f84071a8000 mmap(0x7f84071a8000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe4000) =
0x7f84071a8000
close(3) = 0
mprotect(0x7f84071a8000, 4096, PROT_READ) = 0
mprotect(0x7f84073e0000, 4096, PROT_READ) = 0
munmap(0x7f84071aa000, 22187) = 0

```



```

write(1, "Library switched successfully!\n", 31) = 31
write(1, "Current lib: ./libImpl1.so\n", 27) = 27
write(1, "Input program code:\n", 20) = 20
write(1, " 0 -> Library switch\n", 21) = 21
write(1, " 1 -> PrimeCount\n", 17) = 17
write(1, " 2 -> Pi\n", 9) = 9
write(1, "-1 -> Exit\n", 11) = 11
write(1, "Enter K: Pi(100) = 3.131593\n", 28) = 28
write(1, "Implementation used: ./libImpl1."..., 35) = 35
write(1, "Input program code:\n", 20) = 20
write(1, " 0 -> Library switch\n", 21) = 21
write(1, " 1 -> PrimeCount\n", 17) = 17
write(1, " 2 -> Pi\n", 9) = 9
write(1, "-1 -> Exit\n", 11) = 11
read(0, "", 4096) = 0
munmap(0x7f84073dd000, 16432) = 0
munmap(0x7f84070c3000, 942344) = 0
exit_group(0) = ?
+++ exited with 0 +++

```

Вывод

Работа демонстрирует принципы создания и использования динамических библиотек, а также различия между статической и динамической линковкой.