

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: Закарейшвили Г. М.

Преподаватель: Миронов Е.С. (ПМИ)

Оценка: _____

Дата: 07.03.24

Москва, 2024

Постановка задачи

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 использует одну из библиотек, используя информацию полученную 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 обеих 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:

4	Подсчёт наибольшего общего делителя для двух натуральных чисел	Int GCF(int A, int B)	Алгоритм Евклида	Наивный алгоритм. Пытаться разделить числа на все числа, что меньше A и B.
---	--	-----------------------	------------------	--

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:

progone.c 1.

1. Меню команд:

- Программа показывает пользователю меню с доступными командами:
- 1: Вычислить количество простых чисел в диапазоне.
- 2: Вычислить НОД.
- -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.
- Команда 2:
 - Пользователь вводит два числа C и D.
 - Программа вызывает функцию `GCD(C, D)`, которая возвращает НОД.
- Команда -1:
 - Программа завершает работу.

progtwo.c

1. Загрузка библиотеки:

- Программа загружает библиотеку с помощью функции `dlopen`.
- Если загрузка прошла успешно, программа получает указатели на функции `CalculatePrime` и `GCD` 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: Найти НОД.
- -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 два числа A и B.
 - Программа вызывает функцию `GCD(C, D)` 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

```
progone.c

#include <stdio.h>

// Объявление функций из библиотек
extern int CalculatePrime(int, int);
extern int GCF(int, int);

int main() {
    int command;
    while (1) {
        printf("Input program code:\n");
        printf(" 1 -> PrimeCount\n");
        printf(" 2 -> GCF\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 C, D;
        printf("Enter C and D: ");
        scanf("%d %d", &C, &D);
        printf("GCF(%d, %d) = %d\n", C, D, GCF(C, D));
    } else {
        printf("Invalid command\n");
    }
}

return 0;
}

```

progtwo.c

```

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

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

typedef struct {
    tGCF GCF;
    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.GCF = dlsym(result.library, "GCF");

    if (!result.calculatePrime || !result.GCF) { //не возвращают ли NULL
        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 -> GCF\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 C, D;

```

```

        printf("Enter C and D: ");
        scanf("%d %d", &C, &D);
        printf("GCF(%d, %d) = %d\n", C, D, funcLib.GCF(C, D));
        printf("Implementation used: %s\n", lib_index == 0 ?
"./libImpl1.so" : "./libImpl2.so");
    } else {
        printf("Invalid command\n");
    }
}

dlclose(funcLib.library);
return 0;
}

```

gcd_evclid.c

```

#include <stdio.h>
#include <stdlib.h>

int GCF(int a, int b) {
    a = abs(a); // нод для неотрицательных чисел
    b = abs(b);

    while (b != 0) {
        int remainder = a % b; // остаток от деления
        a = b;
        b = remainder;
    }
    return a; // b == 0, НОД = a
}

```

gcd_naive.c

```

#include <stdio.h>
#include <stdlib.h>

int GCF(int a, int b) {
    a = abs(a);
    b = abs(b);

    if (a == 0 && b == 0) return 0;
    if (a == 0) return b;
    if (b == 0) return a;
}

```

```

int min_val = (a < b) ? a : b;
int max_gcd = 1;

for (int i = 1; i <= min_val; i++) {
    if (a % i == 0 && b % i == 0) {
        max_gcd = i;
    }
}

return max_gcd;
}

```

Протокол работы программы

root@LAPTOP-CGCBKBHR:/home/OS/OS_labs/lab4/src# ./progtwo

Введи команду:

0 -> Library switch

1 -> PrimeCount

2 -> GCF

-1 -> Ценок

2

Введи C и D: 4 24

GCF(4, 24) = 4

заюзана библиотека: ./libImpl1.so

Введи команду:

0 -> Library switch

1 -> PrimeCount

2 -> GCF

-1 -> Ценок

-1

Strace:

progone.c

root@LAPTOP-CGCBKBHR:/home/OS/OS_labs/lab4/src# strace -f ./progone

execve("./progone", ["/progone"], 0x7fffd2475ab8 /* 18 vars */) = 0

brk(NULL) = 0x7fffb026000

arch_prctl(0x3001 /* ARCH_??? */, 0x7fffc4a6b430) = -1 EINVAL (Invalid argument)

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fd273130000

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

openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v3/libImpl1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "./glibc-hwcaps/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/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, "/liblmp1.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\211\1\3\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|0755, st_size=15568, ...}, AT_EMPTY_PATH) = 0
getcwd("/home/OS/OS_labs/lab4/src", 128) = 26
mmap(NULL, 16432, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fd2730e0000
mmap(0x7fd2730e1000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) =
0x7fd2730e1000mmap(0x7fd2730e2000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) =
0x7fd2730e2000
mmap(0x7fd2730e3000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) =
0x7fd2730e3000
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, "/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=16555, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 16555, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fd2730eb000
close(3) = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\211\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\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\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\0\0\0\0\5\0\0\0\0GNU\0\2\0\0\0\300\4\0\0\0\0\3\0\0\0\0\0\0\0"..., 48, 848) = 48
pread64(3, "\4\0\0\0\24\0\0\0\0\3\0\0\0\0GNU\0\302\211\332Pq\2439\235\350\223\322\257\201\326\243f"..., 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\0\0\0"..., 784, 64) = 784
mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fd272eb0000
mprotect(0x7fd272ed8000, 2023424, PROT_NONE) = 0
mmap(0x7fd272ed8000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) =
0x7fd272ed8000
mmap(0x7fd27306d000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) =
0x7fd27306d000
mmap(0x7fd2730c6000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) =
0x7fd2730c6000
mmap(0x7fd2730cc000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) =
0x7fd2730cc000
close(3) = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\211\1\3\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|0644, st_size=940560, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 942344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fd272dc0000
mmap(0x7fd272dce000, 507904, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) =
0x7fd272dce000
mmap(0x7fd272e4a000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8a000) =
0x7fd272e4a000
mmap(0x7fd272ea5000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe4000) =
0x7fd272ea5000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fd272db0000
arch_prctl(ARCH_SET_FS, 0x7fd272db0740) = 0
set_tid_address(0x7fd272db0a10) = 3065
set_robust_list(0x7fd272db0a20, 24) = 0
rseq(0x7fd272db10e0, 0x20, 0, 0x53053053) = -1 ENOSYS (Function not implemented)
mprotect(0x7fd2730c6000, 16384, PROT_READ) = 0
mprotect(0x7fd272ea5000, 4096, PROT_READ) = 0
mprotect(0x7fd2730e3000, 4096, PROT_READ) = 0
mprotect(0x7fd273136000, 4096, PROT_READ) = 0
mprotect(0x7fd273128000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=8192*1024}) = 0
munmap(0x7fd2730eb000, 16555) = 0
newfstatat(1, "", {st_mode=S_IFCHR|0640, st_rdev=makedev(0x4, 0x2), ...}, AT_EMPTY_PATH) = 0
ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0
getrandom("\xa9\x77\x25\x57\x61\xdc\xdd\x9f", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x7fffb026000
brk(0x7fffb047000) = 0x7fffb047000
write(1, "Input program code:\n", 20Input program code:

```

```

) = 20
write(1, " 1 -> PrimeCount\n", 17 1 -> PrimeCount
) = 17
write(1, " 2 -> GCF\n", 10 2 -> GCF
) = 10
write(1, "-1 -> Exit\n", 11-1 -> Exit
) = 11
newfstatat(0, "", {st_mode=S_IFCHR|0640, st_rdev=makedev(0x4, 0x2), ...}, AT_EMPTY_PATH) = 0
ioctl(0, TCGETS, {B38400 opost isig icanon echo ...}) = 0
read(0, 2
"2\n", 4096) = 2
write(1, "Enter C and D: ", 15Enter C and D: ) = 15
read(0, 5 10
"5 10\n", 4096) = 5
write(1, "GCF(5, 10) = 5\n", 15GCF(5, 10) = 5
) = 15
write(1, "Input program code:\n", 20Input program code:
) = 20
write(1, " 1 -> PrimeCount\n", 17 1 -> PrimeCount
) = 17
write(1, " 2 -> GCF\n", 10 2 -> GCF
) = 10
write(1, "-1 -> Exit\n", 11-1 -> Exit
) = 11
read(0, -1
"-1\n", 4096) = 3
lseek(0, -1, SEEK_CUR) = -1 ESPIPE (Illegal seek)
exit_group(0) = ?
+++ exited with 0 +++

```

```

root@LAPTOP-CGCBKBBHR:/home/OS/OS_labs/lab4/src# strace -f ./progtwo
execve("./progtwo", ["/progtwo"], 0x7ffcca27d98 /* 18 vars */) = 0
brk(NULL)
    = 0x7ffdb2f9000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffe213c250) = -1 EINVAL (Invalid argument)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f6b5e7d0000
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=16555, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 16555, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f6b5e7d6000
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\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\0"...
, 784, 64) = 784
pread64(3, "\4\0\0\0\0\0\0\0\5\0\0\0\0GNU\0\2\0\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0\0"...
, 48, 848) = 48
pread64(3, "\4\0\0\0\0\24\0\0\0\3\0\0\0GNU\0\302\211\332Pq\2439\235\350\223\322\257\201\326\243\
f"...
, 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\0"...
, 784, 64) = 784
mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f6b5e560000
mprotect(0x7f6b5e588000, 2023424, PROT_NONE) = 0
mmap(0x7f6b5e588000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) =
0x7f6b5e588000
mmap(0x7f6b5e71d000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7f6b5e71d000
mmap(0x7f6b5e776000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) =
0x7f6b5e776000
mmap(0x7f6b5e77c000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) =
0x7f6b5e77c000
close(3)
    = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f6b5e550000
arch_prctl(ARCH_SET_FS, 0x7f6b5e550740) = 0
set_tid_address(0x7f6b5e550a10)
    = 3105
set_robust_list(0x7f6b5e550a20, 24)
    = 0
rseq(0x7f6b5e5510e0, 0x20, 0, 0x53053053) = -1 ENOSYS (Function not implemented)
mprotect(0x7f6b5e776000, 16384, PROT_READ) = 0
mprotect(0x7f6b5e7de000, 4096, PROT_READ) = 0
mprotect(0x7f6b5e7c8000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=8192*1024}) = 0
munmap(0x7f6b5e7d6000, 16555)
    = 0
getrandom("\xd3\x95\x61\xec\x14\x26\x1b\x3d", 8, GRND_NONBLOCK) = 8
brk(NULL)
    = 0x7ffdb2f9000
brk(0x7ffdb31a000)
    = 0x7ffdb31a000
openat(AT_FDCWD, "/lib/ld-linux-x86_64.so.2", 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|0755, st_size=15568, ...}, AT_EMPTY_PATH) = 0
getcwd("/home/OS/OS_labs/lab4/src", 128) = 26
mmap(NULL, 16432, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f6b5e540000
mmap(0x7f6b5e541000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) =
0x7f6b5e541000
mmap(0x7f6b5e542000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f6b5e542000
mmap(0x7f6b5e543000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) =
0x7f6b5e543000
close(3) = 0
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=16555, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 16555, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f6b5e7d6000
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"..., 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) = 0x7f6b5e450000
mmap(0x7f6b5e45e000, 507904, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) =
0x7f6b5e45e000
mmap(0x7f6b5e4da000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8a000) = 0x7f6b5e4da000
mmap(0x7f6b5e535000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe4000) =
0x7f6b5e535000
close(3) = 0
mprotect(0x7f6b5e535000, 4096, PROT_READ) = 0
mprotect(0x7f6b5e543000, 4096, PROT_READ) = 0
munmap(0x7f6b5e7d6000, 16555) = 0
newfstatat(1, "", {st_mode=S_IFCHR|0640, st_rdev=makedev(0x4, 0x2), ...}, AT_EMPTY_PATH) = 0
ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0
write(1, "Input program code:\n", 20Input program code:
) = 20
write(1, " 0 -> Library switch\n", 21 0 -> Library switch
) = 21
write(1, " 1 -> PrimeCount\n", 17 1 -> PrimeCount
) = 17
write(1, " 2 -> GCF\n", 10 2 -> GCF
) = 10
write(1, "-1 -> Exit\n", 11-1 -> Exit
) = 11
newfstatat(0, "", {st_mode=S_IFCHR|0640, st_rdev=makedev(0x4, 0x2), ...}, AT_EMPTY_PATH) = 0
ioctl(0, TCGETS, {B38400 opost isig icanon echo ...}) = 0
read(0, 1
"1\n", 4096) = 2
write(1, "Enter A and B: ", 15Enter A and B: ) = 15
read(0, 5 10
"5 10\n", 4096) = 5
write(1, "PrimeCount(5, 10) = 2\n", 22PrimeCount(5, 10) = 2
) = 22
write(1, "Implementation used: ./libImpl1."..., 35Implementation used: ./libImpl1.so
) = 35
write(1, "Input program code:\n", 20Input program code:
) = 20
write(1, " 0 -> Library switch\n", 21 0 -> Library switch
) = 21
write(1, " 1 -> PrimeCount\n", 17 1 -> PrimeCount
) = 17
write(1, " 2 -> GCF\n", 10 2 -> GCF
) = 10
write(1, "-1 -> Exit\n", 11-1 -> Exit
) = 11
read(0, 0
"0\n", 4096) = 2
munmap(0x7f6b5e540000, 16432) = 0
munmap(0x7f6b5e450000, 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|0755, st_size=15592, ...}, AT_EMPTY_PATH) = 0
getcwd("/home/OS/OS_labs/lab4/src", 128) = 26
mmap(NULL, 16432, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f6b5e540000
mmap(0x7f6b5e541000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) =
0x7f6b5e541000
mmap(0x7f6b5e542000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f6b5e542000

```

```

mmap(0x7f6b5e543000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) =
0x7f6b5e543000
close(3) = 0
mprotect(0x7f6b5e543000, 4096, PROT_READ) = 0
write(1, "Library switched successfully!\n", 31Library switched successfully!
) = 31
write(1, "Current lib: ./libImpl2.so\n", 27Current lib: ./libImpl2.so
) = 27
write(1, "Input program code:\n", 20Input program code:
) = 20
write(1, " 0 -> Library switch\n", 21 0 -> Library switch
) = 21
write(1, " 1 -> PrimeCount\n", 17 1 -> PrimeCount
) = 17
write(1, " 2 -> GCF\n", 10 2 -> GCF
) = 10
write(1, "-1 -> Exit\n", 11-1 -> Exit
) = 11
read(0, 2
"2\n", 4096) = 2
write(1, "Enter C and D: ", 15Enter C and D: ) = 15
read(0, 2 4
"2 4\n", 4096) = 4
write(1, "GCF(2, 4) = 2\n", 14GCF(2, 4) = 2
) = 14
write(1, "Implementation used: ./libImpl2."..., 35Implementation used: ./libImpl2.so
) = 35
write(1, "Input program code:\n", 20Input program code:
) = 20
write(1, " 0 -> Library switch\n", 21 0 -> Library switch
) = 21
write(1, " 1 -> PrimeCount\n", 17 1 -> PrimeCount
) = 17
write(1, " 2 -> GCF\n", 10 2 -> GCF
) = 10
write(1, "-1 -> Exit\n", 11-1 -> Exit
) = 11
read(0, -1
"-1\n", 4096) = 3
munmap(0x7f6b5e540000, 16432) = 0
lseek(0, -1, SEEK_CUR) = -1 ESPIPE (Illegal seek)
exit_group(0) = ?
+++ exited with 0 +++
root@LAPTOP-CGCBKBHR:/home/OS/OS_labs/lab4/src#

```

Вывод

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