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-216БВ-24

Студент: Попов К.А.

Преподаватель: Бахарев В.Д.

Оценка: _____

Дата: 17.12.25

Москва, 2025

Постановка задачи

Вариант 16.

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:

● Если пользователь вводит команду «0», то программа переключает одну реализацию контрактов на другую (необходимо только для программы №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;

● “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.

Контракты и реализации функций

1. Подсчёт количества простых чисел на отрезке [a, b] (a, b – натуральные):

Сигнатура функции: int prime_count(int a, int b);

● Реализация №1: Наивный алгоритм. Проверить делимость текущего числа на все предыдущие числа.

● Реализация №2: Решето Эратосфена

2. Подсчёт наибольшего общего делителя для двух натуральных чисел:

Сигнатура функции: int gcd(int a, int b);

● Реализация №1: Алгоритм Евклида

● Реализация №2: Наивный алгоритм: пытаться разделить числа на все числа, что меньше a и b.

Общий метод и алгоритм решения

Ключевые функции (POSIX):

- void *dlopen(const char *filename, int mode). Открывает разделяемую библиотеку (динамическую библиотеку), указанную в FILENAME, и загружает её в адресное пространство процесса. Возвращает дескриптор (указатель на "handle"), который можно передать в dlsym для поиска символов (функций или переменных).
- void *dlsym(void *restrict handle, const char *restrict name). Находит адрес символа с именем NAME во время выполнения в разделяемом объекте, на который ссылается HANDLE.
- int dlclose(void *handle). Выгружает динамическую библиотеку из памяти, открытую с помощью dlopen, если больше не осталось ссылок на нее. Дескриптор нельзя использовать повторно после вызова dlclose.

Описание работы программы.

В library_1.c размещены первый вариант реализации функций prime_count и gcd, а в library_2.c - другой вариант реализации. Заголовки функций размещены в файле library.h.

Компиляция динамических библиотек:

gcc -o liblibrary_1.so -shared -fPIC ../src/library_1.c

gcc -o liblibrary_2.so -shared -fPIC ../src/library_2.c

-shared компонует динамическую библиотеку вместо исполняемого файла.

-fPIC создаёт PLT и GOT сегменты и заполняет адреса в этих секциях, чтобы вызовы функций шли по этим адресам.

Программа, к которой динамическая библиотека подключается на этапе компиляции, использует обычный #include “include/library.h”.

Компиляция:

gcc -o main_compile ../main_compile_link.c -I./include -L. -llibrary_1

Здесь -I./include значит искать заголовочные файлы в папке ./include. -L. - искать библиотеки в текущей директории. -llibrary_1 - подключить библиотеку liblibrary_1.so.

Программа, которая динамически загружает библиотеку, использует вместо #include функции dlopen, dlsym и dlclode из dlfcn.h.

Компиляция:

gcc -o main_runtime ../main_runtime_link.c -I./include -ldl

Здесь -ldl подключает libdl.so для работы с динамическими библиотеками.

export LD_LIBRARY_PATH=.:$LD_LIBRARY_PATH - временно добавляет текущую директорию в путь для поиска динамических библиотек.

Код программы

library.h:

```
#ifndef LIBRARY_H
#define LIBRARY_H

#define MAX_NUMBER 2000000  // for sieve

int prime_count(int a, int b);

int gcd(int a, int b);

#endif  // LIBRARY_H
```

library_1.c:

```
#include "../include/library.h"

// NOTE: MSVC compiler does not export symbols unless annotated
#ifdef _MSC_VER
#define EXPORT __declspec(dllexport)
#else
#define EXPORT
#endif

EXPORT int prime_count(int a, int b) {
    int res = 0;
    for (int x = a; x <= b; ++x) {
        if (x < 2) {
            continue;
        }
        int is_prime = 1;
        for (int d = 2; d * d <= x; ++d) {
            if (x % d == 0) {
                is_prime = 0;
                break;
            }
        }
        res += is_prime;
    }
    return res;
}

EXPORT int gcd(int a, int b) {
    while (b) {
        int tmp = a;
        a = b;
        b = tmp % b;
    }
    return a;
}
```

library_2.c:

```
#include "../include/library.h"

// NOTE: MSVC compiler does not export symbols unless annotated
#ifdef _MSC_VER
#define EXPORT __declspec(dllexport)
#else
#define EXPORT
#endif

EXPORT int prime_count(int a, int b) {
    int res = 0;
    static int sieve[MAX_NUMBER] = {0};
    static int is_init = 0;
    if (!is_init) {
        sieve[0] = sieve[1] = 1;
        for (int i = 2; i < MAX_NUMBER; ++i) {
            if (sieve[i] == 0 && i <= MAX_NUMBER / i) {
                for (int j = i * i; j < MAX_NUMBER; j += i) {
                    sieve[j] = 1;
                }
            }
        }
    }
    for (int x = a; x <= b; ++x) {
        res += (int)(sieve[x] == 0);
    }
    return res;
}

EXPORT int gcd(int a, int b) {
    if (b < a) {
        int tmp = a;
        a = b;
        b = tmp;
    }
    for (int i = a; i > 1; --i) {
        if (a % i == 0 && b % i == 0) {
            return i;
        }
    }
    return 1;
}
```

main_compile_link.c:

```
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <unistd.h>

#include "include/library.h"
```

```
#define BUFSIZE 1024

int readLine(char *buffer, const ssize_t size) {
    ssize_t bytes_read = read(STDIN_FILENO, buffer, size);
    if (bytes_read < 0) {
        return 1;
    }
    if (bytes_read == 0 || buffer[0] == '\n') {
        return 2;
    }
    if (bytes_read == size && buffer[size - 1] != '\n') {
        return 3;
    }
    if (buffer[bytes_read - 1] == '\n') {
        buffer[bytes_read - 1] = '\0';
    } else {
        buffer[bytes_read] = '\0';
    }
    return 0;
}

void print_invalid_input_msg() {
    const char msg[] = "error: invalid input\n";
    write(STDERR_FILENO, msg, sizeof(msg));
}

int main() {
    char user_input[BUFSIZE];
    int running = 1;

    do {
        int err = readLine(user_input, BUFSIZE);
        switch (err) {
            case 1: {
                const char msg[] = "error: failed to read stdin\n";
                write(STDERR_FILENO, msg, sizeof(msg));
            } exit(EXIT_FAILURE);
            case 2: {
                running = 0;
            } break;
            case 3: {
                const char msg[] = "error: buffer overflow\n";
                write(STDERR_FILENO, msg, sizeof(msg));
            } exit(EXIT_FAILURE);
        }

        if (!running) {
            break;
        }

        char *arg = strtok(user_input, " \t\n");
        int command = atoi(arg);
        switch (command) {
            case 1:
```

```
                case 2: {
                    int a = atoi(strtok(NULL, " \t\n"));
                    int b = atoi(strtok(NULL, " \t\n"));
                    int res;
                    if (command == 1) {
                        res = prime_count(a, b);
                    } else {
                        res = gcd(a, b);
                    }
                    char result_msg[BUFSIZE];
                    int msg_size = snprintf(result_msg, BUFSIZE, "%d\n",
res);
                    write(STDOUT_FILENO, result_msg, msg_size);
                } break;
                default: {
                    const char msg[] = "error: invalid input\n";
                    write(STDERR_FILENO, msg, sizeof(msg));
                } exit(EXIT_FAILURE);
            }
        } while (running);

        return 0;
    }
```

main_runtime_link.c:

```
    #include <stddef.h>
    #include <stdlib.h>
    #include <string.h>
    #include <stdio.h>

    #include <unistd.h> // read, write
    #include <dlfcn.h> // dlopen, dlsym, dlclose, RTLD_*

    #define BUFSIZE 1024
    const char *LIB_NAMES[2] = { "./liblibrary_1.so", "./liblibrary_2.so" };

    typedef int (*prime_count_func)(int, int);
    typedef int (*gcd_func)(int, int);

    int readLine(char *buffer, const ssize_t size) {
        ssize_t bytes_read = read(STDIN_FILENO, buffer, size);
        if (bytes_read < 0) {
            return 1;
        }
        if (bytes_read == 0 || buffer[0] == '\n') {
            return 2;
        }
        if (bytes_read == size && buffer[size - 1] != '\n') {
            return 3;
        }
        if (buffer[bytes_read - 1] == '\n') {
            buffer[bytes_read - 1] = '\0';
        } else {
            buffer[bytes_read] = '\0';
```

```
        }
        return 0;
    }

    void print_invalid_input_msg() {
        const char msg[] = "error: invalid input\n";
        write(STDERR_FILENO, msg, sizeof(msg));
    }

    static prime_count_func prime_count;
    static gcd_func gcd;

    static int func_impl_stub(int a, int b) {
        (void)a;
        (void)b;
        return 0;
    }

    int connect_library(const char *lib_name, void **library) {
        if (!lib_name || !library) {
            return -1;
        }
        if (*library) {
            dlclose(*library);
        }
        *library = NULL;
        *library = dlopen(lib_name, RTLD_LOCAL | RTLD_NOW);
        if (!*library) {
            return -1;
        }
        prime_count = dlsym(*library, "prime_count");
        if (prime_count == NULL) {
            const char msg[] = "warning: failed to find prime_count function
implementation\n";
            write(STDERR_FILENO, msg, sizeof(msg));
            prime_count = func_impl_stub;
        }
        gcd = dlsym(*library, "gcd");
        if (gcd == NULL) {
            const char msg[] = "warning: failed to find gcd function
implementation\n";
            write(STDERR_FILENO, msg, sizeof(msg));
            gcd = func_impl_stub;
        }
        return 0;
    }

    int connect_next_lib(void **library) {
        static int lib_number = 1;
        lib_number = lib_number == 0 ? 1 : 0;
        return connect_library(LIB_NAMES[lib_number], library);
    }

    int main() {
```

```
        void *library = NULL;
        if (connect_next_lib(&library)) {
            const char msg[] = "error: library failed to load\n";
            write(STDERR_FILENO, msg, sizeof(msg));
            exit(EXIT_FAILURE);
        }

        char user_input[BUFSIZE];
        int running = 1;
        do {
            int err = readLine(user_input, BUFSIZE);
            switch (err) {
                case 1: {
                    const char msg[] = "error: failed to read stdin\n";
                    write(STDERR_FILENO, msg, sizeof(msg));
                } exit(EXIT_FAILURE);
                case 2: {
                    running = 0;
                } break;
                case 3: {
                    const char msg[] = "error: buffer overflow\n";
                    write(STDERR_FILENO, msg, sizeof(msg));
                } exit(EXIT_FAILURE);
            }
            if (!running) {
                break;
            }

            char *arg = strtok(user_input, " \t\n");
            int command = atoi(arg);
            switch (command) {
                case 0: {
                    if (connect_next_lib(&library)) {
                        const char msg[] = "error: library failed to load\n";
                        write(STDERR_FILENO, msg, sizeof(msg));
                        exit(EXIT_FAILURE);
                    }
                } break;
                case 1:
                case 2: {
                    int a = atoi(strtok(NULL, " \t\n"));
                    int b = atoi(strtok(NULL, " \t\n"));
                    int res;
                    if (command == 1) {
                        res = prime_count(a, b);
                    } else {
                        res = gcd(a, b);
                    }
                    char result_msg[BUFSIZE];
                    int msg_size = snprintf(result_msg, BUFSIZE, "%d\n",
res);
                    write(STDOUT_FILENO, result_msg, msg_size);
                } break;
                default: {
```

```
                const char msg[] = "error: invalid input\n";
                write(STDERR_FILENO, msg, sizeof(msg));
            } exit(EXIT_FAILURE);
        }
    } while (running);

    if (library) {
        dlclose(library);
    }

    return 0;
}
```

Протокол работы программы

Тестирование (связывание библиотеки во время исполнения):

```
root@4bcd85e1faed:/workspaces/OS_labs/lab4/build# strace -
o ../tests/strace_out.txt -f ./main_runtime
1 1 10
4
2 15 30
15
0
1 1 10
4
2 15 30
15
0
1 1 10
4
2 15 30
15
```

strace_out.txt:

```
42058 execve("./main_runtime", ["./main_runtime"], 0x7ffe53714ce8 /* 28
vars */) = 0
42058 brk(NULL)                         = 0x311fc000
42058 mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS,
-1, 0) = 0x70d17c7d9000
42058 access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or
directory)
42058 openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v3/libc.so.6",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
42058 openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v2/libc.so.6",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
42058 openat(AT_FDCWD, "./tls/x86_64/x86_64/libc.so.6",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
42058 openat(AT_FDCWD, "./tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -
1 ENOENT (No such file or directory)
42058 openat(AT_FDCWD, "./tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -
1 ENOENT (No such file or directory)
```

```
    42058 openat(AT_FDCWD, "./tls/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
    42058 openat(AT_FDCWD, "./x86_64/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC)
= -1 ENOENT (No such file or directory)
    42058 openat(AT_FDCWD, "./x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
    42058 openat(AT_FDCWD, "./x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
    42058 openat(AT_FDCWD, "./libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
    42058 openat(AT_FDCWD, "glibc-hwcaps/x86-64-v3/libc.so.6",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
    42058 openat(AT_FDCWD, "glibc-hwcaps/x86-64-v2/libc.so.6",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
    42058 openat(AT_FDCWD, "tls/x86_64/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC)
= -1 ENOENT (No such file or directory)
    42058 openat(AT_FDCWD, "tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
    42058 openat(AT_FDCWD, "tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
    42058 openat(AT_FDCWD, "tls/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
    42058 openat(AT_FDCWD, "x86_64/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) =
-1 ENOENT (No such file or directory)
    42058 openat(AT_FDCWD, "x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
    42058 openat(AT_FDCWD, "x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
    42058 openat(AT_FDCWD, "libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
    42058 openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
    42058 newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=33091, ...},
AT_EMPTY_PATH) = 0
    42058 mmap(NULL, 33091, PROT_READ, MAP_PRIVATE, 3, 0) = 0x70d17c7d0000
    42058 close(3)                                  = 0
    42058 openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6",
O_RDONLY|O_CLOEXEC) = 3
    42058 read(3,
"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\20t\2\0\0\0\0\0"..., 832) =
832
    42058 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
    42058 newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=1926232, ...},
AT_EMPTY_PATH) = 0
    42058 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
    42058 mmap(NULL, 1974096, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x70d17c5ee000
    42058 mmap(0x70d17c614000, 1400832, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x26000) = 0x70d17c614000
    42058 mmap(0x70d17c76a000, 339968, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x17c000) = 0x70d17c76a000
```

```
    42058 mmap(0x70d17c7bd000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1cf000) = 0x70d17c7bd000
    42058 mmap(0x70d17c7c3000, 53072, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x70d17c7c3000
    42058 close(3)                          = 0
    42058 mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS,
-1, 0) = 0x70d17c5eb000
    42058 arch_prctl(ARCH_SET_FS, 0x70d17c5eb740) = 0
    42058 set_tid_address(0x70d17c5eba10)   = 42058
    42058 set_robust_list(0x70d17c5eba20, 24) = 0
    42058 rseq(0x70d17c5ec060, 0x20, 0, 0x53053053) = 0
    42058 mprotect(0x70d17c7bd000, 16384, PROT_READ) = 0
    42058 mprotect(0x403000, 4096, PROT_READ) = 0
    42058 mprotect(0x70d17c80c000, 8192, PROT_READ) = 0
    42058 prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
    42058 munmap(0x70d17c7d0000, 33091)     = 0
    42058 getrandom("\x46\x0e\xd1\x54\xc7\x1e\x25\x11", 8, GRND_NONBLOCK) =
8
    42058 brk(NULL)                         = 0x311fc000
    42058 brk(0x3121d000)                   = 0x3121d000
    42058 openat(AT_FDCWD, "./liblibrary_1.so", O_RDONLY|O_CLOEXEC) = 3
    42058 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
    42058 newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=15192, ...},
AT_EMPTY_PATH) = 0
    42058 getcwd("/workspaces/OS_labs/lab4/build", 128) = 31
    42058 mmap(NULL, 16400, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x70d17c7d4000
    42058 mmap(0x70d17c7d5000, 4096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x70d17c7d5000
    42058 mmap(0x70d17c7d6000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x70d17c7d6000
    42058 mmap(0x70d17c7d7000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x70d17c7d7000
    42058 close(3)                          = 0
    42058 mprotect(0x70d17c7d7000, 4096, PROT_READ) = 0
    42058 read(0, "1 1 10\n", 1024)         = 7
    42058 write(1, "4\n", 2)                = 2
    42058 read(0, "2 15 30\n", 1024)        = 8
    42058 write(1, "15\n", 3)               = 3
    42058 read(0, "0\n", 1024)              = 2
    42058 munmap(0x70d17c7d4000, 16400)     = 0
    42058 openat(AT_FDCWD, "./liblibrary_2.so", O_RDONLY|O_CLOEXEC) = 3
    42058 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
    42058 newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=15256, ...},
AT_EMPTY_PATH) = 0
    42058 getcwd("/workspaces/OS_labs/lab4/build", 128) = 31
    42058 mmap(NULL, 8016480, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x70d17be45000
```

```
    42058 mmap(0x70d17be46000, 4096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x70d17be46000
    42058 mmap(0x70d17be47000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x70d17be47000
    42058 mmap(0x70d17be48000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x70d17be48000
    42058 mmap(0x70d17be4a000, 7996000, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x70d17be4a000
    42058 close(3)                          = 0
    42058 mprotect(0x70d17be48000, 4096, PROT_READ) = 0
    42058 read(0, "1 1 10\n", 1024)         = 7
    42058 write(1, "4\n", 2)                = 2
    42058 read(0, "2 15 30\n", 1024)        = 8
    42058 write(1, "15\n", 3)               = 3
    42058 read(0, "0\n", 1024)              = 2
    42058 munmap(0x70d17be45000, 8016480)   = 0
    42058 openat(AT_FDCWD, "./liblibrary_1.so", O_RDONLY|O_CLOEXEC) = 3
    42058 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
    42058 newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=15192, ...},
AT_EMPTY_PATH) = 0
    42058 getcwd("/workspaces/OS_labs/lab4/build", 128) = 31
    42058 mmap(NULL, 16400, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x70d17c7d4000
    42058 mmap(0x70d17c7d5000, 4096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x70d17c7d5000
    42058 mmap(0x70d17c7d6000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x70d17c7d6000
    42058 mmap(0x70d17c7d7000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x70d17c7d7000
    42058 close(3)                          = 0
    42058 mprotect(0x70d17c7d7000, 4096, PROT_READ) = 0
    42058 read(0, "1 1 10\n", 1024)         = 7
    42058 write(1, "4\n", 2)                = 2
    42058 read(0, "2 15 30\n", 1024)        = 8
    42058 write(1, "15\n", 3)               = 3
    42058 read(0, "\n", 1024)               = 1
    42058 munmap(0x70d17c7d4000, 16400)     = 0
    42058 exit_group(0)                     = ?
    42058 +++ exited with 0 +++
```

Тестирование (связывание библиотеки во время компиляции):

```
    root@4bcd85e1faed:/workspaces/OS_labs/lab4/build# strace -
o ../tests/strace_out.txt -f ./main_compile
    1 1 10
    4
    2 15 30
    15
```

strace_out.txt:

```
    41661 execve("./main_compile", ["./main_compile"], 0x7ffd20e19fa8 /* 28 vars */)
= 0
    41661 brk(NULL)                         = 0x34002000
```

```
     41661 mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0)
= 0x74a3fb963000
     41661 access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
     41661 openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v3/liblibrary_1.so",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
     41661 openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v2/liblibrary_1.so",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
     41661 openat(AT_FDCWD, "./tls/x86_64/x86_64/liblibrary_1.so",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
     41661 openat(AT_FDCWD, "./tls/x86_64/liblibrary_1.so", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
     41661 openat(AT_FDCWD, "./tls/x86_64/liblibrary_1.so", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
     41661 openat(AT_FDCWD, "./tls/liblibrary_1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
     41661 openat(AT_FDCWD, "./x86_64/x86_64/liblibrary_1.so", O_RDONLY|O_CLOEXEC) =
-1 ENOENT (No such file or directory)
     41661 openat(AT_FDCWD, "./x86_64/liblibrary_1.so", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
     41661 openat(AT_FDCWD, "./x86_64/liblibrary_1.so", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
     41661 openat(AT_FDCWD, "./liblibrary_1.so", O_RDONLY|O_CLOEXEC) = 3
     41661 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
     41661 newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=15192, ...},
AT_EMPTY_PATH) = 0
     41661 getcwd("/workspaces/OS_labs/lab4/build", 128) = 31
     41661 mmap(NULL, 16400, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x74a3fb95e000
     41661 mmap(0x74a3fb95f000, 4096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x74a3fb95f000
     41661 mmap(0x74a3fb960000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x2000) = 0x74a3fb960000
     41661 mmap(0x74a3fb961000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x74a3fb961000
     41661 close(3)                          = 0
     41661 openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v3/libc.so.6", O_RDONLY|O_CLOEXEC)
= -1 ENOENT (No such file or directory)
     41661 openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v2/libc.so.6", O_RDONLY|O_CLOEXEC)
= -1 ENOENT (No such file or directory)
     41661 openat(AT_FDCWD, "./tls/x86_64/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -
1 ENOENT (No such file or directory)
     41661 openat(AT_FDCWD, "./tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
     41661 openat(AT_FDCWD, "./tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
     41661 openat(AT_FDCWD, "./tls/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
     41661 openat(AT_FDCWD, "./x86_64/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
     41661 openat(AT_FDCWD, "./x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
     41661 openat(AT_FDCWD, "./x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
     41661 openat(AT_FDCWD, "./libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
     41661 openat(AT_FDCWD, "glibc-hwcaps/x86-64-v3/libc.so.6", O_RDONLY|O_CLOEXEC)
= -1 ENOENT (No such file or directory)
```

```
     41661 openat(AT_FDCWD, "glibc-hwcaps/x86-64-v2/libc.so.6", O_RDONLY|O_CLOEXEC)
= -1 ENOENT (No such file or directory)
     41661 openat(AT_FDCWD, "tls/x86_64/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
     41661 openat(AT_FDCWD, "tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
     41661 openat(AT_FDCWD, "tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
     41661 openat(AT_FDCWD, "tls/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
     41661 openat(AT_FDCWD, "x86_64/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
     41661 openat(AT_FDCWD, "x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
     41661 openat(AT_FDCWD, "x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
     41661 openat(AT_FDCWD, "libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
     41661 openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
     41661 newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=33091, ...},
AT_EMPTY_PATH) = 0
     41661 mmap(NULL, 33091, PROT_READ, MAP_PRIVATE, 3, 0) = 0x74a3fb955000
     41661 close(3)                          = 0
     41661 openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) =
3
     41661 read(3,
"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\20t\2\0\0\0\0\0"..., 832) = 832
     41661 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
     41661 newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=1926232, ...},
AT_EMPTY_PATH) = 0
     41661 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
     41661 mmap(NULL, 1974096, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x74a3fb773000
     41661 mmap(0x74a3fb799000, 1400832, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x26000) = 0x74a3fb799000
     41661 mmap(0x74a3fb8ef000, 339968, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x17c000) = 0x74a3fb8ef000
     41661 mmap(0x74a3fb942000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1cf000) = 0x74a3fb942000
     41661 mmap(0x74a3fb948000, 53072, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x74a3fb948000
     41661 close(3)                          = 0
     41661 mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0)
= 0x74a3fb770000
     41661 arch_prctl(ARCH_SET_FS, 0x74a3fb770740) = 0
     41661 set_tid_address(0x74a3fb770a10)   = 41661
     41661 set_robust_list(0x74a3fb770a20, 24) = 0
     41661 rseq(0x74a3fb771060, 0x20, 0, 0x53053053) = 0
     41661 mprotect(0x74a3fb942000, 16384, PROT_READ) = 0
     41661 mprotect(0x74a3fb961000, 4096, PROT_READ) = 0
     41661 mprotect(0x403000, 4096, PROT_READ) = 0
     41661 mprotect(0x74a3fb996000, 8192, PROT_READ) = 0
     41661 prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
     41661 munmap(0x74a3fb955000, 33091)     = 0
     41661 read(0, "1 1 10\n", 1024)         = 7
     41661 write(1, "4\n", 2)                = 2
```

```
41661 read(0, "2 15 30\n", 1024)        = 8
41661 write(1, "15\n", 3)                = 3
41661 read(0, "\n", 1024)               = 1
41661 exit_group(0)                     = ?
41661 +++ exited with 0 +++
```

Вывод

Приобрел практические навыки в создании динамических библиотек в ОС Linux и программ, которые используют эти библиотеки. Было изучено два способа подключения динамических библиотек: на этапе компиляции и во время исполнения.