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: Матвеев Семен Романович

Преподаватель: Миронов Евгений Сергеевич

Оценка:

Дата: 13.03.25

Москва, 2025

**Постановка задачи**

# Вариант 21.

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.

Было необходимо реализовать подсчет количества простых чисел на отрезке [A, B] наивным алгоритмом и решетом Эратосфена; реализовать сортировку целочисленного массива пузырьковой сортировкой и сортировкой Хоара.

**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:

**static\_linking.cpp**

1. Статическое подключение/линковка библиотек

2. Вывод меню

Программа выводит статическое меню с двумя опциями:

Подсчёт количества простых чисел на интервале [A,B].

Сортировка целочисленного массива.

3. Обработка выбора пользователя

Пользователь вводит число (1 или 2), чтобы выбрать опцию.

4. Опция 1: Подсчёт простых чисел

Если пользователь выбрал опцию 1:

Программа запрашивает два натуральных числа A и B.

Вызывает функцию PrimeCount(A, B), которая возвращает количество простых чисел на интервале [A,B].

Выводит результат.

5. Опция 2: Сортировка массива

Если пользователь выбрал опцию 2:

Программа запрашивает длину массива.

Если длина корректна (не отрицательна), программа запрашивает элементы массива.

Вызывает функцию Sort(arr, size), которая сортирует массив (например, с использованием алгоритма пузырьковой сортировки).

Выводит отсортированный массив.

6. Обработка некорректного ввода

Если пользователь ввёл недопустимый выбор (не 1 и не 2), программа выводит сообщение об ошибке.

# runtime\_linking.cpp

1. Загрузка динамических библиотек

Программа вызывает функцию load\_libs(), которая:

Загружает две динамические библиотеки:

libs/prime\_sieve.so (содержит функцию PrimeCount для подсчёта простых чисел).

libs/hoare\_sort.so (содержит функцию Sort для сортировки массива).

Получает указатели на функции PrimeCount и Sort с помощью dlsym.

Если загрузка библиотек или функций не удалась, программа завершается с ошибкой.

2. Вывод меню

Программа выводит меню с двумя опциями:

Подсчёт количества простых чисел на интервале [A,B].

Сортировка целочисленного массива.

3. Обработка выбора пользователя

Пользователь вводит число (1 или 2), чтобы выбрать опцию.

4. Опция 1: Подсчёт простых чисел

Если пользователь выбрал опцию 1:

Программа запрашивает два натуральных числа A и B.

Вызывает функцию PrimeCount из динамической библиотеки, которая возвращает количество простых чисел на интервале [A,B].

Выводит результат.

5. Опция 2: Сортировка массива

Если пользователь выбрал опцию 2:

Программа запрашивает длину массива.

Если длина корректна (не отрицательна), программа запрашивает элементы массива.

Вызывает функцию Sort из динамической библиотеки, которая сортирует массив (например, с использованием алгоритма Хоара).

Выводит отсортированный массив.

6. Обработка некорректного ввода

Если пользователь ввёл недопустимый выбор (не 1 и не 2), программа выводит сообщение об ошибке.

7. Освобождение ресурсов

После завершения работы программа закрывает загруженные динамические библиотеки с помощью dlclose.

**Код программы**

**static\_linking.cpp**

#include "../include/prime\_naive.cpp"

#include "../include/bubble\_sort.cpp"

#include <iostream>

int main() {

std::cout << "\n[STATIC MENU]\nВыберите опцию:\n1) Подсчет кол-ва простых чисел в интервале [А; Б]\n2) Сортировка целочисленного массива\n";

int choice;

std::cin >> choice;

if (choice == 1) {

std::cout << "Введите натуральные числа A и Б:\n";

int A, B;

std::cin >> A >> B;

std::cout << "Результат: " << PrimeCount(A, B) << std::endl;

} else if (choice == 2) {

std::cout << "Введите длину массива:\n";

int size;

std::cin >> size;

if (size < 0) {

std::cout << "Некорректная длина массива!\n";

}

else {

std::cout << "Введите " << size << " целых чисел, элементы массива:\n";

int\* arr = new int[size];

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

std::cin >> arr[i];

}

Sort(arr, size);

std::cout << "Результат: ";

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

std::cout << arr[i] << " ";

}

std::cout << std::endl;

delete[] arr;

}

} else {

std::cout << "Некорретный ввод!\n";

}

}

**runtime\_linking.cpp**

#include <iostream>

#include <cstdlib>

#include <cstddef>

#include <dlfcn.h>

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

int\* (\*SortFunc)(int\*, int);

void\* handle\_prime;

void\* handle\_sort;

void load\_libs();

int main() {

load\_libs();

std::cout << "\n[RUNTIME MENU]\nВыберите опцию:\n1) Подсчет кол-ва простых чисел в интервале [А; Б]\n2) Сортировка целочисленного массива\n";

int choice;

std::cin >> choice;

if (choice == 1) {

std::cout << "Введите натуральные числа A и Б:\n";

int A, B;

std::cin >> A >> B;

std::cout << "Результат: " << (\*PrimeCountFunc)(A, B) << std::endl;

} else if (choice == 2) {

std::cout << "Введите длину массива:\n";

int size;

std::cin >> size;

if (size < 0) {

std::cout << "Некорректная длина массива!\n";

}

else {

std::cout << "Введите " << size << " целых чисел, элементы массива:\n";

int\* arr = new int[size];

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

std::cin >> arr[i];

}

(\*SortFunc)(arr, size);

std::cout << "Результат: ";

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

std::cout << arr[i] << " ";

}

std::cout << std::endl;

delete[] arr;

}

} else {

std::cout << "Некорретный ввод!\n";

}

dlclose(handle\_prime);

dlclose(handle\_sort);

}

void load\_libs() {

handle\_prime = dlopen("libs/prime\_sieve.so", RTLD\_LAZY);

handle\_sort = dlopen("libs/hoare\_sort.so", RTLD\_LAZY);

if (!handle\_prime) {

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

exit(1);

}

if (!handle\_sort) {

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

exit(1);

}

dlerror();

PrimeCountFunc = (int (\*)(int, int))dlsym(handle\_prime, "PrimeCount");

char\* error = dlerror();

if (error) {

fprintf(stderr, "Не удалось найти функцию PrimeCount: %s\n", error);

exit(1);

}

SortFunc = (int\* (\*)(int\*, int))dlsym(handle\_sort, "Sort");

error = dlerror();

if (error) {

fprintf(stderr, "Не удалось найти функцию Sort: %s\n", error);

exit(1);

}

}

**prime\_naive.cpp**

#include <cmath>

#include <iostream>

extern "C" {

bool isPrime(int n) {

if (n <= 1) return false;

if (n == 2) return true;

if (n % 2 == 0) return false;

for (int i = 3; i <= std::sqrt(n); i += 2) {

if (n % i == 0) return false;

}

return true;

}

int PrimeCount(int A, int B) {

if (B < 2) return 0;

if (A < 2) A = 2;

int count = 0;

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

if (isPrime(num)) {

++count;

}

}

return count;

}

}

**prime\_sieve.cpp**

#include <vector>

extern "C" {

int PrimeCount(int A, int B) {

std::vector<bool> is\_prime(B + 1, true);

is\_prime[0] = is\_prime[1] = false;

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

if (is\_prime[i]) {

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

is\_prime[j] = false;

}

}

}

int count = 0;

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

if (is\_prime[i]) {

++count;

}

}

return count;

}

}

**bubble\_sort.cpp**

#include <algorithm>

// Отсортировать целочисленный массив;

// Реализация 1 - Пузырьковая сортировка

extern "C" {

int\* Sort(int\* array, int size) {

for (int i = 0; i < size - 1; ++i) {

for (int j = 0; j < size - i - 1; ++j) {

if (array[j] > array[j + 1]) {

std::swap(array[j], array[j + 1]);

}

}

}

return array;

}

}

**hoare\_sort.cpp**

#include <vector>

// Отсортировать целочисленный массив;

// Реализация 2 - Сортировка Хоара

extern "C" {

int\* Sort(int\* array, int size) {

if (size < 2) return array;

int pivot = array[size / 2];

std::vector<int> left, right;

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

if (i == size / 2) continue;

if (array[i] < pivot) {

left.push\_back(array[i]);

} else {

right.push\_back(array[i]);

}

}

Sort(left.data(), left.size());

Sort(right.data(), right.size());

std::copy(left.begin(), left.end(), array);

array[left.size()] = pivot;

std::copy(right.begin(), right.end(), array + left.size() + 1);

return array;

}

**Протокол работы программы**

# static\_linking.cpp:

# [STATIC MENU]

# Выберите опцию:

# 1) Подсчет кол-ва простых чисел в интервале [А; Б]

# 2) Сортировка целочисленного массива

# 1

# Введите натуральные числа A и Б:

# 5

# 1000

# Результат: 166

**runtime\_linking.cpp:**

[RUNTIME MENU]

Выберите опцию:

1) Подсчет кол-ва простых чисел в интервале [А; Б]

2) Сортировка целочисленного массива

2

Введите длину массива:

5

Введите 5 целых чисел, элементы массива:

-5 64 21 0 62

Результат: -5 0 21 62 64

# Strace:

# **static\_linking.cpp:**

# strace -f ./static

# execve("./static", ["./static"], 0x7ffe25dcc398 /\* 77 vars \*/) = 0

# brk(NULL) = 0x55fdfa747000

# arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffc5568dfb0) = -1 EINVAL (Invalid argument)

# mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f9f29549000

# 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=167639, ...}, AT\_EMPTY\_PATH) = 0

# mmap(NULL, 167639, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f9f29520000

# close(3) = 0

# openat(AT\_FDCWD, "/lib/x86\_64-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>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

# newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=2260296, ...}, AT\_EMPTY\_PATH) = 0

# mmap(NULL, 2275520, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f9f292f4000

# mprotect(0x7f9f2938e000, 1576960, PROT\_NONE) = 0

# mmap(0x7f9f2938e000, 1118208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9a000) = 0x7f9f2938e000

# mmap(0x7f9f2949f000, 454656, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ab000) = 0x7f9f2949f000

# mmap(0x7f9f2950f000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x21a000) = 0x7f9f2950f000

# mmap(0x7f9f2951d000, 10432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f9f2951d000

# 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\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) = 0x7f9f2920d000

# mmap(0x7f9f2921b000, 507904, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe000) = 0x7f9f2921b000

# mmap(0x7f9f29297000, 372736, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x8a000) = 0x7f9f29297000

# mmap(0x7f9f292f2000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe4000) = 0x7f9f292f2000

# 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\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\315A\vq\17\17\tLh2\355\331Y1\0m"..., 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) = 0x7f9f28fe4000

# mprotect(0x7f9f2900c000, 2023424, PROT\_NONE) = 0

# mmap(0x7f9f2900c000, 1658880, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7f9f2900c000

# mmap(0x7f9f291a1000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1bd000) = 0x7f9f291a1000

# mmap(0x7f9f291fa000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x215000) = 0x7f9f291fa000

# mmap(0x7f9f29200000, 52816, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f9f29200000

# close(3) = 0

# openat(AT\_FDCWD, "/lib/x86\_64-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>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

# newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=125488, ...}, AT\_EMPTY\_PATH) = 0

# mmap(NULL, 127720, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f9f28fc4000

# mmap(0x7f9f28fc7000, 94208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f9f28fc7000

# mmap(0x7f9f28fde000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a000) = 0x7f9f28fde000

# mmap(0x7f9f28fe2000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d000) = 0x7f9f28fe2000

# close(3) = 0

# mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f9f28fc2000

# arch\_prctl(ARCH\_SET\_FS, 0x7f9f28fc33c0) = 0

# set\_tid\_address(0x7f9f28fc3690) = 814968

# set\_robust\_list(0x7f9f28fc36a0, 24) = 0

# rseq(0x7f9f28fc3d60, 0x20, 0, 0x53053053) = 0

# mprotect(0x7f9f291fa000, 16384, PROT\_READ) = 0

# mprotect(0x7f9f28fe2000, 4096, PROT\_READ) = 0

# mprotect(0x7f9f292f2000, 4096, PROT\_READ) = 0

# mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f9f28fc0000

# mprotect(0x7f9f2950f000, 45056, PROT\_READ) = 0

# mprotect(0x55fdc02c0000, 4096, PROT\_READ) = 0

# mprotect(0x7f9f29583000, 8192, PROT\_READ) = 0

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

# munmap(0x7f9f29520000, 167639) = 0

# getrandom("\x2d\xd0\xb8\x62\x2b\xc5\x84\xd2", 8, GRND\_NONBLOCK) = 8

# brk(NULL) = 0x55fdfa747000

# brk(0x55fdfa768000) = 0x55fdfa768000

# futex(0x7f9f2951d77c, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

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

# write(1, "\n", 1

# ) = 1

# write(1, "[STATIC MENU]\n", 14[STATIC MENU]

# ) = 14

# write(1, "\320\222\321\213\320\261\320\265\321\200\320\270\321\202\320\265 \320\276\320\277\321\206\320\270\321\216:\n", 29Выберите опцию:

# ) = 29

# write(1, "1) \320\237\320\276\320\264\321\201\321\207\320\265\321\202 \320\272\320\276\320\273-\320\262\320\260 \320\277"..., 871) Подсчет кол-ва простых чисел в интервале [А; Б]

# ) = 87

# write(1, "2) \320\241\320\276\321\200\321\202\320\270\321\200\320\276\320\262\320\272\320\260 \321\206\320\265\320\273\320\276"..., 682) Сортировка целочисленного массива

# ) = 68

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

# read(0, 0x55fdfa7592c0, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

# --- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

# read(0, 0x55fdfa7592c0, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

# --- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

# read(0, 0x55fdfa7592c0, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

# --- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

# read(0, 0x55fdfa7592c0, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

# --- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

# read(0, 0x55fdfa7592c0, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

# --- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

# read(0, 0x55fdfa7592c0, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

# --- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

# read(0, 0x55fdfa7592c0, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

# --- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

# read(0, 0x55fdfa7592c0, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

# --- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

# read(0, 0x55fdfa7592c0, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

# --- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

# read(0, 0x55fdfa7592c0, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

# --- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

# read(0, 0x55fdfa7592c0, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

# --- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

# read(0, 0x55fdfa7592c0, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

# --- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

# read(0, 0x55fdfa7592c0, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

# --- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

# read(0, 1

# "1\n", 1024) = 2

# write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\275\320\260\321\202\321\203\321\200\320\260\320\273\321\214\320"..., 58Введите натуральные числа A и Б:

# ) = 58

# read(0, 1 500

# "1 500\n", 1024) = 6

# write(1, "\320\240\320\265\320\267\321\203\320\273\321\214\321\202\320\260\321\202: 95\n", 23Результат: 95

# ) = 23

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

# exit\_group(0) = ?

# +++ exited with 0 +++

**runtime\_linking.cpp:**

strace -f ./runtime

execve("./runtime", ["./runtime"], 0x7ffd741033a8 /\* 77 vars \*/) = 0

brk(NULL) = 0x55ffd5a4a000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffe75df3700) = -1 EINVAL (Invalid argument)

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fe2fc34c000

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=167639, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 167639, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7fe2fc323000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-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>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=2260296, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 2275520, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe2fc0f7000

mprotect(0x7fe2fc191000, 1576960, PROT\_NONE) = 0

mmap(0x7fe2fc191000, 1118208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9a000) = 0x7fe2fc191000

mmap(0x7fe2fc2a2000, 454656, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ab000) = 0x7fe2fc2a2000

mmap(0x7fe2fc312000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x21a000) = 0x7fe2fc312000

mmap(0x7fe2fc320000, 10432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fe2fc320000

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\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\315A\vq\17\17\tLh2\355\331Y1\0m"..., 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) = 0x7fe2fbece000

mprotect(0x7fe2fbef6000, 2023424, PROT\_NONE) = 0

mmap(0x7fe2fbef6000, 1658880, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7fe2fbef6000

mmap(0x7fe2fc08b000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1bd000) = 0x7fe2fc08b000

mmap(0x7fe2fc0e4000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x215000) = 0x7fe2fc0e4000

mmap(0x7fe2fc0ea000, 52816, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fe2fc0ea000

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\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) = 0x7fe2fbde7000

mmap(0x7fe2fbdf5000, 507904, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe000) = 0x7fe2fbdf5000

mmap(0x7fe2fbe71000, 372736, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x8a000) = 0x7fe2fbe71000

mmap(0x7fe2fbecc000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe4000) = 0x7fe2fbecc000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-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>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=125488, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 127720, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe2fbdc7000

mmap(0x7fe2fbdca000, 94208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7fe2fbdca000

mmap(0x7fe2fbde1000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a000) = 0x7fe2fbde1000

mmap(0x7fe2fbde5000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d000) = 0x7fe2fbde5000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fe2fbdc5000

arch\_prctl(ARCH\_SET\_FS, 0x7fe2fbdc63c0) = 0

set\_tid\_address(0x7fe2fbdc6690) = 816025

set\_robust\_list(0x7fe2fbdc66a0, 24) = 0

rseq(0x7fe2fbdc6d60, 0x20, 0, 0x53053053) = 0

mprotect(0x7fe2fc0e4000, 16384, PROT\_READ) = 0

mprotect(0x7fe2fbde5000, 4096, PROT\_READ) = 0

mprotect(0x7fe2fbecc000, 4096, PROT\_READ) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fe2fbdc3000

mprotect(0x7fe2fc312000, 45056, PROT\_READ) = 0

mprotect(0x55ffbc619000, 4096, PROT\_READ) = 0

mprotect(0x7fe2fc38c000, 8192, PROT\_READ) = 0

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

munmap(0x7fe2fc323000, 167639) = 0

getrandom("\x32\x9d\xb9\x9e\xae\x17\xd4\x68", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x55ffd5a4a000

brk(0x55ffd5a6b000) = 0x55ffd5a6b000

futex(0x7fe2fc32077c, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

openat(AT\_FDCWD, "libs/prime\_sieve.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|0775, st\_size=28376, ...}, AT\_EMPTY\_PATH) = 0

getcwd("/home/user/Desktop/all/main/projects/c\_projects/MAI\_OS\_Labs\_3sem/lab4", 128) = 70

mmap(NULL, 25048, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe2fc345000

mmap(0x7fe2fc347000, 8192, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7fe2fc347000

mmap(0x7fe2fc349000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7fe2fc349000

mmap(0x7fe2fc34a000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7fe2fc34a000

close(3) = 0

mprotect(0x7fe2fc34a000, 4096, PROT\_READ) = 0

openat(AT\_FDCWD, "libs/hoare\_sort.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|0775, st\_size=33752, ...}, AT\_EMPTY\_PATH) = 0

getcwd("/home/user/Desktop/all/main/projects/c\_projects/MAI\_OS\_Labs\_3sem/lab4", 128) = 70

mmap(NULL, 29240, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe2fc33d000

mmap(0x7fe2fc340000, 8192, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7fe2fc340000

mmap(0x7fe2fc342000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x5000) = 0x7fe2fc342000

mmap(0x7fe2fc343000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x5000) = 0x7fe2fc343000

close(3) = 0

mprotect(0x7fe2fc343000, 4096, PROT\_READ) = 0

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

write(1, "\n", 1

) = 1

write(1, "[RUNTIME MENU]\n", 15[RUNTIME MENU]

) = 15

write(1, "\320\222\321\213\320\261\320\265\321\200\320\270\321\202\320\265 \320\276\320\277\321\206\320\270\321\216:\n", 29Выберите опцию:

) = 29

write(1, "1) \320\237\320\276\320\264\321\201\321\207\320\265\321\202 \320\272\320\276\320\273-\320\262\320\260 \320\277"..., 871) Подсчет кол-ва простых чисел в интервале [А; Б]

) = 87

write(1, "2) \320\241\320\276\321\200\321\202\320\270\321\200\320\276\320\262\320\272\320\260 \321\206\320\265\320\273\320\276"..., 682) Сортировка целочисленного массива

) = 68

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

read(0, 1

"1\n", 1024) = 2

write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\275\320\260\321\202\321\203\321\200\320\260\320\273\321\214\320"..., 58Введите натуральные числа A и Б:

) = 58

read(0, 1

"1\n", 1024) = 2

read(0, 500

"500\n", 1024) = 4

write(1, "\320\240\320\265\320\267\321\203\320\273\321\214\321\202\320\260\321\202: 95\n", 23Результат: 95

) = 23

munmap(0x7fe2fc345000, 25048) = 0

munmap(0x7fe2fc33d000, 29240) = 0

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

exit\_group(0) = ?

+++ exited with 0 +++

**Вывод**

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