Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт №8 “Компьютерные науки и прикладная математика”

Кафедра №806 “Вычислительная математика и программирование”

**Лабораторная работа №2 по курсу**

**«Операционные системы»**

Группа: М8О-210Б-23

Студент: Морозов Захар Олегович

Преподаватель: Бахарев В.Д. (ФИИТ)

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

Дата: 18.02.2025

Москва, 2025

**Постановка задачи**

**Вариант 15.**

Составить программу на языке Си, обрабатывающую данные в многопоточном режиме. При обработки использовать стандартные средства создания потоков операционной системы (Windows/Unix). Ограничение максимального количества потоков, работающих в один момент времени, должно быть задано ключом запуска вашей программы.

Отсортировать массив целых чисел при помощи параллельного алгоритма быстрой сортировки

**Общий метод и алгоритм решения**

Использованные системные вызовы:

* int pthread\_create(pthread\_t \**thread*, const pthread\_attr\_t \**attr*, void \*(\**start\_routine*)(void\*), void \**arg*);- создание потока.
* int pthread\_mutex\_lock(pthread\_mutex\_t \**mutex*); - блокировка мьютекса
* int pthread\_mutex\_unlock(pthread\_mutex\_t \**mutex*); - разблокировка мьютекса
* int pthread\_cond\_signal(pthread\_cond\_t \*cond) - отправляет сигнал с помощью условной переменной
* int pthread\_cond\_wait(pthread\_cond\_t \*restrict cond, pthread\_mutex\_t \*restrict mutex) –

функция, которая переводит поток в ожидающее состояние, пока не будет получен сигнал с помощью условной переменной.

* void exit(int code) – завершение программы.
* int pthread\_detach(pthread\_t thread) - "Отсоединяет" поток от родительского потока.

Программа предназначена для сортировки массива целых чисел с использованием параллельного алгоритма быстрой сортировки (QuickSort). Для ускорения процесса сортировки применяется многозадачность, где работа распределяется между несколькими потоками. Основные этапы работы программы:

1. Создание данных: Программа создаёт массив указанной длинны и заполняет его случайными числами от 1 до 9999.

2. Параллельная сортировка: Массив разбивается на подмассивы, которые сортируются независимо друг от друга. Для каждого подмассива создается отдельный поток, который выполняет сортировку с использованием алгоритма быстрой сортировки. Размер подмассивов определяется на основе количества доступных ядер процессора и ограничений по оперативной памяти.

3. Использование многозадачности: Для управления потоками и синхронизации их работы используются мьютексы и условные переменные. Главный поток контролирует создание и завершение рабочих потоков, чтобы избежать гонок за ресурсами и обеспечить корректное выполнение сортировки. После завершения сортировки каждого подмассива результаты объединяются в один отсортированный массив.

4. Слияние отсортированных подмассивов: После завершения сортировки всех подмассивов выполняется их слияние в один отсортированный массив.

5. Освобождение ресурсов: После сортировки массива программа выводит время сортировки и освобождает массив.

**Код программы**

**main.c**

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <unistd.h>

#include <time.h>

int \*array;

pthread\_mutex\_t mutex = PTHREAD\_MUTEX\_INITIALIZER;

typedef struct {

    int left;

    int right;

} ThreadData;

void swap(int \*a, int \*b) {

    int temp = \*a;

    \*a = \*b;

    \*b = temp;

}

int partition(int left, int right) {

    int pivot = array[right];

    int i = left - 1;

    for (int j = left; j <= right - 1; j++) {

        if (array[j] < pivot) {

            i++;

            swap(&array[i], &array[j]);

        }

    }

    swap(&array[i + 1], &array[right]);

    return i + 1;

}

void \*quicksort(void \*arg) {

    ThreadData \*data = (ThreadData \*)arg;

    int left = data->left;

    int right = data->right;

    if (left < right) {

        int pi = partition(left, right);

        ThreadData left\_data = {left, pi - 1};

        ThreadData right\_data = {pi + 1, right};

        pthread\_t left\_thread, right\_thread;

        pthread\_mutex\_lock(&mutex);

        pthread\_create(&left\_thread, NULL, quicksort, &left\_data);

        pthread\_create(&right\_thread, NULL, quicksort, &right\_data);

        pthread\_mutex\_unlock(&mutex);

        pthread\_join(left\_thread, NULL);

        pthread\_join(right\_thread, NULL);

    }

    pthread\_exit(NULL);

}

int main(int argc, char \*argv[]) {

    if (argc != 3) {

        const char msg[] = "Usage: %s <array\_size> <max\_threads>\n";

        write(STDERR\_FILENO, msg, sizeof(msg));

        exit(EXIT\_SUCCESS);

    }

    int size;

    size = atoi(argv[1]);

    int num\_threads = atoi(argv[2]);

    if (size <= 0 || num\_threads <= 0) {

        const char msg[] = "Error: Array size and max threads must be positive integers.\n";

        write(STDERR\_FILENO, msg, sizeof(msg));

        exit(EXIT\_SUCCESS);

    }

    array = (int\*)malloc(size \* sizeof(int));

    if (array == NULL) {

        const char msg[] = "Error: Cannot allocate memory.\n";

        write(STDERR\_FILENO, msg, sizeof(msg));

        exit(EXIT\_SUCCESS);

    }

    clock\_t start\_creation, end\_creation;

    srand(time(NULL));

    for (int i = 0; i < size; i++) {

        array[i] = rand() % 1000;

    }

    ThreadData data = {0, size - 1};

    pthread\_t main\_thread;

    start\_creation = clock();

    pthread\_create(&main\_thread, NULL, quicksort, &data);

    pthread\_join(main\_thread, NULL);

    end\_creation = clock();

    printf("Full time: %f seconds\n", (double)(end\_creation - start\_creation) / CLOCKS\_PER\_SEC);

    free(array);

    return 0;

}

**Протокол работы программы**

**Тестирование:**

****



****

****

**Strace:**

execve("./a.out", ["./a.out", "100", "3"], 0x7ffcd2497b18 /\* 22 vars \*/) = 0

brk(NULL) = 0x556f7ba21000

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f920a70a000

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

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=19871, ...}) = 0

mmap(NULL, 19871, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f920a705000

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\0\220\243\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

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2125328, ...}) = 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, 2170256, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f920a4f3000

mmap(0x7f920a51b000, 1605632, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7f920a51b000

mmap(0x7f920a6a3000, 323584, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1b0000) = 0x7f920a6a3000

mmap(0x7f920a6f2000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1fe000) = 0x7f920a6f2000

mmap(0x7f920a6f8000, 52624, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f920a6f8000

close(3) = 0

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f920a4f0000

arch\_prctl(ARCH\_SET\_FS, 0x7f920a4f0740) = 0

set\_tid\_address(0x7f920a4f0a10) = 23219

set\_robust\_list(0x7f920a4f0a20, 24) = 0

rseq(0x7f920a4f1060, 0x20, 0, 0x53053053) = 0

mprotect(0x7f920a6f2000, 16384, PROT\_READ) = 0

mprotect(0x556f7ac8d000, 4096, PROT\_READ) = 0

mprotect(0x7f920a742000, 8192, PROT\_READ) = 0

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

munmap(0x7f920a705000, 19871) = 0

getrandom("\x9a\x47\xcd\xba\xbb\x44\x60\x3e", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x556f7ba21000

brk(0x556f7ba42000) = 0x556f7ba42000

clock\_gettime(CLOCK\_PROCESS\_CPUTIME\_ID, {tv\_sec=0, tv\_nsec=5505800}) = 0

rt\_sigaction(SIGRT\_1, {sa\_handler=0x7f920a58c520, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_ONSTACK|SA\_RESTART|SA\_SIGINFO, sa\_restorer=0x7f920a538320}, NULL, 8) = 0

rt\_sigprocmask(SIG\_UNBLOCK, [RTMIN RT\_1], NULL, 8) = 0

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f9209cef000

mprotect(0x7f9209cf0000, 8388608, PROT\_READ|PROT\_WRITE) = 0

rt\_sigprocmask(SIG\_BLOCK, ~[], [], 8) = 0

clone3({flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, child\_tid=0x7f920a4ef990, parent\_tid=0x7f920a4ef990, exit\_signal=0, stack=0x7f9209cef000, stack\_size=0x7fff80, tls=0x7f920a4ef6c0}strace: Process 23220 attached

=> {parent\_tid=[23220]}, 88) = 23220

[pid 23220] rseq(0x7f920a4effe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 23219] rt\_sigprocmask(SIG\_SETMASK, [], <unfinished ...>

[pid 23220] <... rseq resumed>) = 0

[pid 23219] <... rt\_sigprocmask resumed>NULL, 8) = 0

[pid 23220] set\_robust\_list(0x7f920a4ef9a0, 24 <unfinished ...>

[pid 23219] futex(0x7f920a4ef990, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 23220, NULL, FUTEX\_BITSET\_MATCH\_ANY <unfinished ...>

[pid 23220] <... set\_robust\_list resumed>) = 0

[pid 23220] rt\_sigprocmask(SIG\_SETMASK, [], NULL, 8) = 0

[pid 23220] mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f92094ee000

[pid 23220] mprotect(0x7f92094ef000, 8388608, PROT\_READ|PROT\_WRITE) = 0

[pid 23220] mmap(NULL, 134217728, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f92014ee000

[pid 23220] munmap(0x7f92014ee000, 45162496) = 0

[pid 23220] munmap(0x7f9208000000, 21946368) = 0

[pid 23220] mprotect(0x7f9204000000, 135168, PROT\_READ|PROT\_WRITE) = 0

[pid 23220] rt\_sigprocmask(SIG\_BLOCK, ~[], [], 8) = 0

[pid 23220] clone3({flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, child\_tid=0x7f9209cee990, parent\_tid=0x7f9209cee990, exit\_signal=0, stack=0x7f92094ee000, stack\_size=0x7fff80, tls=0x7f9209cee6c0}strace: Process 23221 attached

<unfinished ...>

[pid 23221] rseq(0x7f9209ceefe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 23220] <... clone3 resumed> => {parent\_tid=[23221]}, 88) = 23221

[pid 23221] <... rseq resumed>) = 0

[pid 23220] rt\_sigprocmask(SIG\_SETMASK, [], <unfinished ...>

[pid 23221] set\_robust\_list(0x7f9209cee9a0, 24 <unfinished ...>

[pid 23220] <... rt\_sigprocmask resumed>NULL, 8) = 0

[pid 23221] <... set\_robust\_list resumed>) = 0

[pid 23220] mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0 <unfinished ...>

[pid 23221] rt\_sigprocmask(SIG\_SETMASK, [], <unfinished ...>

[pid 23220] <... mmap resumed>) = 0x7f9208ced000

[pid 23221] <... rt\_sigprocmask resumed>NULL, 8) = 0

[pid 23220] mprotect(0x7f9208cee000, 8388608, PROT\_READ|PROT\_WRITE <unfinished ...>

[pid 23221] futex(0x556f7ac8e060, FUTEX\_WAIT\_PRIVATE, 2, NULL <unfinished ...>

[pid 23220] <... mprotect resumed>) = 0

[pid 23220] rt\_sigprocmask(SIG\_BLOCK, ~[], [], 8) = 0

[pid 23343] <... mprotect resumed>) = 0

[pid 23341] rt\_sigprocmask(SIG\_BLOCK, ~[RT\_1], NULL, 8) = 0

[pid 23343] rt\_sigprocmask(SIG\_BLOCK, ~[], <unfinished ...>

[pid 23341] madvise(0x7f9200ffa000, 8368128, MADV\_DONTNEED <unfinished ...>

[pid 23343] <... rt\_sigprocmask resumed>[], 8) = 0

[pid 23341] <... madvise resumed>) = 0

[pid 23327] <... futex resumed>) = 0

[pid 23292] futex(0x7f920a7450f8, FUTEX\_WAKE\_PRIVATE, 1 <unfinished ...>

[pid 23327] munmap(0x7f91ceffe000, 8392704 <unfinished ...>

[pid 23331] <... futex resumed>) = 0

[pid 23292] <... futex resumed>) = 1

[pid 23331] futex(0x7f920a7450f8, FUTEX\_WAIT\_PRIVATE, 2, NULL <unfinished ...>

[pid 23327] <... munmap resumed>) = 0

[pid 23292] rt\_sigprocmask(SIG\_BLOCK, ~[RT\_1], <unfinished ...>

[pid 23327] futex(0x7f920a7450f8, FUTEX\_WAKE\_PRIVATE, 1 <unfinished ...>

[pid 23292] <... rt\_sigprocmask resumed>NULL, 8) = 0

[pid 23331] <... futex resumed>) = 0

[pid 23327] <... futex resumed>) = 1

[pid 23331] munmap(0x7f91da7fd000, 8392704 <unfinished ...>

[pid 23292] madvise(0x7f91bcffa000, 8368128, MADV\_DONTNEED <unfinished ...>

[pid 23331] <... munmap resumed>) = 0

[pid 23327] futex(0x7f920a7450f8, FUTEX\_WAIT\_PRIVATE, 2, NULL <unfinished ...>

[pid 23292] <... madvise resumed>) = 0

[pid 23331] futex(0x7f920a7450f8, FUTEX\_WAKE\_PRIVATE, 1 <unfinished ...>

[pid 23327] <... futex resumed>) = -1 EAGAIN (Resource temporarily unavailable)

[pid 23292] exit(0 <unfinished ...>

[pid 23331] <... futex resumed>) = 0

[pid 23327] munmap(0x7f91ccffa000, 8392704 <unfinished ...>

[pid 23331] futex(0x7f920a7450f8, FUTEX\_WAIT\_PRIVATE, 2, NULL <unfinished ...>

[pid 23292] <... exit resumed>) = ?

[pid 23327] <... munmap resumed>) = 0

[pid 23292] +++ exited with 0 +++

[pid 23279] <... futex resumed>) = 0

[pid 23327] futex(0x7f920a7450f8, FUTEX\_WAKE\_PRIVATE, 1 <unfinished ...>

[pid 23279] munmap(0x7f91d8ffa000, 8392704) = 0

[pid 23331] <... futex resumed>) = 0

[pid 23327] <... futex resumed>) = 1

[pid 23331] munmap(0x7f92037ff000, 8392704 <unfinished ...>

[pid 23279] rt\_sigprocmask(SIG\_BLOCK, ~[RT\_1], <unfinished ...>

[pid 23331] <... munmap resumed>) = 0

[pid 23327] rt\_sigprocmask(SIG\_BLOCK, ~[RT\_1], <unfinished ...>

[pid 23279] <... rt\_sigprocmask resumed>NULL, 8) = 0

[pid 23331] futex(0x7f920a7450f8, FUTEX\_WAKE\_PRIVATE, 1 <unfinished ...>

[pid 23327] <... rt\_sigprocmask resumed>NULL, 8) = 0

[pid 23331] <... futex resumed>) = 0

[pid 23327] madvise(0x7f91b67fd000, 8368128, MADV\_DONTNEED <unfinished ...>

[pid 23279] madvise(0x7f91beffe000, 8368128, MADV\_DONTNEED <unfinished ...>

[pid 23331] rt\_sigprocmask(SIG\_BLOCK, ~[RT\_1], <unfinished ...>

[pid 23327] <... madvise resumed>) = 0

[pid 23279] <... madvise resumed>) = 0

[pid 23331] <... rt\_sigprocmask resumed>NULL, 8) = 0

[pid 23327] exit(0 <unfinished ...>

[pid 23279] exit(0 <unfinished ...>

[pid 23331] madvise(0x7f91b77ff000, 8368128, MADV\_DONTNEED <unfinished ...>

[pid 23327] <... exit resumed>) = ?

[pid 23279] <... exit resumed>) = ?

[pid 23331] <... madvise resumed>) = 0

[pid 23327] +++ exited with 0 +++

[pid 23331] exit(0 <unfinished ...>

[pid 23279] +++ exited with 0 +++

[pid 23263] <... futex resumed>) = 0

[pid 23331] <... exit resumed>) = ?

[pid 23281] <... futex resumed>) = 0

[pid 23263] munmap(0x7f91cf7ff000, 8392704 <unfinished ...>

[pid 23331] +++ exited with 0 +++

[pid 23324] <... futex resumed>) = 0

[pid 23263] <... munmap resumed>) = 0

[pid 23324] futex(0x7f920a7450f8, FUTEX\_WAIT\_PRIVATE, 2, NULL <unfinished ...>

[pid 23281] futex(0x7f920a7450f8, FUTEX\_WAIT\_PRIVATE, 2, NULL <unfinished ...>

[pid 23263] futex(0x7f920a7450f8, FUTEX\_WAKE\_PRIVATE, 1 <unfinished ...>

[pid 23324] <... futex resumed>) = -1 EAGAIN (Resource temporarily unavailable)

[pid 23281] <... futex resumed>) = -1 EAGAIN (Resource temporarily unavailable)

[pid 23324] munmap(0x7f92007f9000, 8392704 <unfinished ...>

[pid 23281] futex(0x7f920a7450f8, FUTEX\_WAIT\_PRIVATE, 2, NULL <unfinished ...>

[pid 23263] <... futex resumed>) = 0

[pid 23324] <... munmap resumed>) = 0

[pid 23263] futex(0x7f9201ffb990, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 23280, NULL, FUTEX\_BITSET\_MATCH\_ANY <unfinished ...>

[pid 23324] futex(0x7f920a7450f8, FUTEX\_WAKE\_PRIVATE, 1) = 1

[pid 23281] <... futex resumed>) = 0

[pid 23324] munmap(0x7f91bcffa000, 8392704 <unfinished ...>

[pid 23281] futex(0x7f920a7450f8, FUTEX\_WAIT\_PRIVATE, 2, NULL <unfinished ...>

[pid 23324] <... munmap resumed>) = 0

[pid 23324] futex(0x7f920a7450f8, FUTEX\_WAKE\_PRIVATE, 1) = 1

[pid 23281] <... futex resumed>) = 0

[pid 23324] rt\_sigprocmask(SIG\_BLOCK, ~[RT\_1], <unfinished ...>

[pid 23281] munmap(0x7f91d87f9000, 8392704 <unfinished ...>

[pid 23324] <... rt\_sigprocmask resumed>NULL, 8) = 0

[pid 23281] <... munmap resumed>) = 0

[pid 23324] madvise(0x7f92094ee000, 8368128, MADV\_DONTNEED <unfinished ...>

[pid 23281] futex(0x7f920a7450f8, FUTEX\_WAKE\_PRIVATE, 1 <unfinished ...>

[pid 23324] <... madvise resumed>) = 0

[pid 23281] <... futex resumed>) = 0

[pid 23324] exit(0 <unfinished ...>

[pid 23281] munmap(0x7f91beffe000, 8392704 <unfinished ...>

[pid 23324] <... exit resumed>) = ?

[pid 23281] <... munmap resumed>) = 0

[pid 23324] +++ exited with 0 +++

[pid 23264] <... futex resumed>) = 0

[pid 23281] rt\_sigprocmask(SIG\_BLOCK, ~[RT\_1], <unfinished ...>

[pid 23264] munmap(0x7f91b77ff000, 8392704 <unfinished ...>

[pid 23281] <... rt\_sigprocmask resumed>NULL, 8) = 0

[pid 23264] <... munmap resumed>) = 0

[pid 23281] madvise(0x7f91db7ff000, 8368128, MADV\_DONTNEED <unfinished ...>

[pid 23264] rt\_sigprocmask(SIG\_BLOCK, ~[RT\_1], <unfinished ...>

[pid 23281] <... madvise resumed>) = 0

[pid 23264] <... rt\_sigprocmask resumed>NULL, 8) = 0

[pid 23281] exit(0 <unfinished ...>

[pid 23264] madvise(0x7f91cd7fb000, 8368128, MADV\_DONTNEED <unfinished ...>

[pid 23281] <... exit resumed>) = ?

[pid 23264] <... madvise resumed>) = 0

[pid 23281] +++ exited with 0 +++

[pid 23280] <... futex resumed>) = 0

[pid 23264] exit(0 <unfinished ...>

[pid 23280] munmap(0x7f92084ec000, 8392704 <unfinished ...>

[pid 23264] <... exit resumed>) = ?

[pid 23280] <... munmap resumed>) = 0

[pid 23264] +++ exited with 0 +++

[pid 23280] munmap(0x7f91b67fd000, 8392704) = 0

[pid 23280] rt\_sigprocmask(SIG\_BLOCK, ~[RT\_1], NULL, 8) = 0

[pid 23280] madvise(0x7f92017fb000, 8368128, MADV\_DONTNEED) = 0

[pid 23280] exit(0) = ?

[pid 23280] +++ exited with 0 +++

[pid 23263] <... futex resumed>) = 0

[pid 23263] munmap(0x7f91bf7ff000, 8392704) = 0

[pid 23263] rt\_sigprocmask(SIG\_BLOCK, ~[RT\_1], NULL, 8) = 0

[pid 23263] madvise(0x7f91cc7f9000, 8368128, MADV\_DONTNEED) = 0

[pid 23263] exit(0) = ?

[pid 23263] +++ exited with 0 +++

[pid 23250] <... futex resumed>) = 0

[pid 23250] munmap(0x7f92094ee000, 8392704) = 0

[pid 23250] munmap(0x7f91db7ff000, 8392704) = 0

[pid 23250] rt\_sigprocmask(SIG\_BLOCK, ~[RT\_1], NULL, 8) = 0

[pid 23250] madvise(0x7f91d97fb000, 8368128, MADV\_DONTNEED) = 0

[pid 23250] exit(0) = ?

[pid 23250] +++ exited with 0 +++

[pid 23237] <... futex resumed>) = 0

[pid 23237] munmap(0x7f91cdffc000, 8392704) = 0

[pid 23237] rt\_sigprocmask(SIG\_BLOCK, ~[RT\_1], NULL, 8) = 0

[pid 23237] madvise(0x7f91ecffa000, 8368128, MADV\_DONTNEED) = 0

[pid 23237] exit(0) = ?

[pid 23237] +++ exited with 0 +++

[pid 23236] <... futex resumed>) = 0

[pid 23236] munmap(0x7f92017fb000, 8392704) = 0

[pid 23236] munmap(0x7f91cc7f9000, 8392704) = 0

[pid 23236] rt\_sigprocmask(SIG\_BLOCK, ~[RT\_1], NULL, 8) = 0

[pid 23236] madvise(0x7f91ed7fb000, 8368128, MADV\_DONTNEED) = 0

[pid 23236] exit(0) = ?

[pid 23236] +++ exited with 0 +++

[pid 23234] <... futex resumed>) = 0

[pid 23234] munmap(0x7f91cd7fb000, 8392704) = 0

[pid 23234] rt\_sigprocmask(SIG\_BLOCK, ~[RT\_1], NULL, 8) = 0

[pid 23234] madvise(0x7f91ee7fd000, 8368128, MADV\_DONTNEED) = 0

[pid 23234] exit(0) = ?

[pid 23234] +++ exited with 0 +++

[pid 23227] <... futex resumed>) = 0

[pid 23227] munmap(0x7f91d97fb000, 8392704) = 0

[pid 23227] rt\_sigprocmask(SIG\_BLOCK, ~[RT\_1], NULL, 8) = 0

[pid 23227] madvise(0x7f92027fd000, 8368128, MADV\_DONTNEED) = 0

[pid 23227] exit(0) = ?

[pid 23227] +++ exited with 0 +++

[pid 23222] <... futex resumed>) = 0

[pid 23222] munmap(0x7f91ecffa000, 8392704) = 0

[pid 23222] munmap(0x7f91edffc000, 8392704) = 0

[pid 23222] rt\_sigprocmask(SIG\_BLOCK, ~[RT\_1], NULL, 8) = 0

[pid 23222] madvise(0x7f9208ced000, 8368128, MADV\_DONTNEED) = 0

[pid 23222] exit(0) = ?

[pid 23222] +++ exited with 0 +++

[pid 23220] <... futex resumed>) = 0

[pid 23220] munmap(0x7f91ed7fb000, 8392704) = 0

[pid 23220] rt\_sigprocmask(SIG\_BLOCK, ~[RT\_1], NULL, 8) = 0

[pid 23220] madvise(0x7f9209cef000, 8368128, MADV\_DONTNEED) = 0

[pid 23220] exit(0) = ?

[pid 23219] <... futex resumed>) = 0

[pid 23220] +++ exited with 0 +++

munmap(0x7f91ee7fd000, 8392704) = 0

clock\_gettime(CLOCK\_PROCESS\_CPUTIME\_ID, {tv\_sec=0, tv\_nsec=159768000}) = 0

fstat(1, {st\_mode=S\_IFCHR|0600, st\_rdev=makedev(0x88, 0), ...}) = 0

write(1, "Full time: 0.154263 seconds\n", 28Full time: 0.154263 seconds

) = 28

exit\_group(0) = ?

+++ exited with 0 +++

|  |  |  |  |
| --- | --- | --- | --- |
| Число потоков | Время выполнения, мс | Ускорение | Эффективность |
| 1 | 1263 | 1 | 1 |
| 2 | 1252 | 1, 008 | 0.504 |
| 3 | 1250 | 1,01 | 0.336 |
| 4 | 1241 | 1,017 | 0.254 |

**Объяснение.**

Увеличение количества потоков позволяет сократить время выполнения программы за счёт распределения вычислений между несколькими параллельными задачами. Однако с каждым новым добавленным потоком эффективность прироста производительности снижается. Это происходит из-за дополнительной нагрузки на процессор, который способен выполнять ограниченное количество операций одновременно. Когда количество потоков превышает число доступных ядер процессора, дальнейший прирост эффективности становится незначительным, так как потоки начинают конкурировать за ресурсы и выполняются фактически последовательно, а не параллельно.

**Вывод**

В процессе выполнения лабораторной работы я освоил базовые навыки работы с потоками на языке C. Кроме того, я познакомился с основными принципами параллельного программирования и применил их на практике. Во время работы мне пришлось преодолеть определённые трудности, так как написание параллельно выполняемого кода впервые оказалось достаточно сложной задачей.