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-209БВ-24

Студент: Крюков Д. М.

Преподаватель: Миронов Е.С.

Оценка: _____

Дата: 01.12.25

Москва, 2025

Постановка задачи

Вариант 2.

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

Так же необходимо уметь продемонстрировать количество потоков, используемое вашей программой с помощью стандартных средств операционной системы.

В отчете привести исследование зависимости ускорения и эффективности алгоритма от входных данных и количества потоков. Получившиеся результаты необходимо объяснить.

Задание из варианта: отсортировать массив целых чисел при помощи параллельной сортировки слиянием

Общий метод и алгоритм решения

Использованные системные вызовы:

- pthread_create() – создаёт новый поток выполнения внутри текущего процесса. Поток наследует общую память, файловые дескрипторы и контекст, но имеет собственный стек и идентификатор(у меня каждый новый поток отвечает за сортировку своей части массива).
- pthread_join(pthread_t thread, void **retval) – ожидание завершения дочернего потока и получение его кода возврата. Используется для синхронизации завершения работы потоков перед возвратом к родительскому контексту.
- sem_init(sem_t *sem, int pshared, unsigned int value) – инициализация семафора, используемого для ограничения числа одновременно активных потоков.
Семафор служит средством управления ресурсами — когда поток создаётся, выполняется sem_trywait(), а после завершения — sem_post().
- sem_trywait(sem_t *sem) / sem_post(sem_t *sem) – атомарные операции ожидания и освобождения семафора; предотвращает создание чрезмерного количества потоков и стабилизирует использование системных ресурсов.
- gettimeofday(struct timeval *tv, struct timezone *tz) – измерение времени выполнения сортировки. Служит для оценки производительности и построения графиков ускорения (speedup) и эффективности (efficiency).

Описание метода и логики программы

1. **Инициализация данных**: программа считывает из файла размер массива и его элементы. Максимальное число потоков передаётся через аргумент командной строки.
2. **Параллельная сортировка**: реализован рекурсивный алгоритм merge sort. Если доступен свободный поток (семафор разрешает), то левая половина массива сортируется в отдельном потоке, правая — в текущем. Если потоков недостаточно — сортировка выполняется последовательно.
3. **Синхронизация**: семафор ограничивает число одновременно работающих потоков.
4. **Измерение времени**: фиксируется время начала и конца сортировки.

Код программы

main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#include <sys/time.h>

#define MIN_SIZE 10000

sem_t thread_limit;

typedef struct {
    int* arr;
    int left;
    int right;
} thread_data;

double get_time_ms() {
    struct timeval tv;
    gettimeofday(&tv, NULL);
    return tv.tv_sec * 1000.0 + tv.tv_usec / 1000.0;
}

void merge(int* arr, int left, int mid, int right) {
    int n1 = mid - left + 1;
    int n2 = right - mid;

    int* L = malloc(n1 * sizeof(int));
    int* R = malloc(n2 * sizeof(int));
```

```
    for (int i = 0; i < n1; i++) L[i] = arr[left + i];
    for (int j = 0; j < n2; j++) R[j] = arr[mid + 1 + j];

    int i = 0, j = 0, k = left;

    while (i < n1 && j < n2) {
        arr[k++] = (L[i] <= R[j]) ? L[i++] : R[j++];
    }
    while (i < n1) arr[k++] = L[i++];
    while (j < n2) arr[k++] = R[j++];

    free(L);
    free(R);
}

void merge_sort(int* arr, int left, int right) {
    if (left >= right) return;
    int mid = left + (right - left) / 2;
    merge_sort(arr, left, mid);
    merge_sort(arr, mid + 1, right);
    merge(arr, left, mid, right);
}

void parallel_merge_sort(int* arr, int left, int right);

void* thread_parallel_sort(void* arg) {
    thread_data* data = (thread_data*)arg;

    parallel_merge_sort(data->arr, data->left, data->right);
```

```
    sem_post(&thread_limit);
    free(data);
    return NULL;
}

void parallel_merge_sort(int* arr, int left, int right) {
    if (left >= right) return;

    if (right - left < MIN_SIZE) {
        merge_sort(arr, left, right);
        return;
    }

    int mid = left + (right - left) / 2;

    pthread_t t1 = 0, t2 = 0;
    int created1 = 0, created2 = 0;

    if (sem_trywait(&thread_limit) == 0) {
        thread_data* d = malloc(sizeof(thread_data));
        d->arr = arr;
        d->left = left;
        d->right = mid;
        if (pthread_create(&t1, NULL, thread_parallel_sort, d) == 0) {
            created1 = 1;
        } else {
            free(d);
            sem_post(&thread_limit);
        }
```

```
    }

    if (sem_trywait(&thread_limit) == 0) {
        thread_data* d = malloc(sizeof(thread_data));
        d->arr = arr;
        d->left = mid + 1;
        d->right = right;
        if (pthread_create(&t2, NULL, thread_parallel_sort, d) == 0) {
            created2 = 1;
        } else {
            free(d);
            sem_post(&thread_limit);
        }
    }

    if (!created1) parallel_merge_sort(arr, left, mid);
    if (!created2) parallel_merge_sort(arr, mid + 1, right);

    if (created1) pthread_join(t1, NULL);
    if (created2) pthread_join(t2, NULL);

    merge(arr, left, mid, right);
}

int main(int argc, char* argv[]) {
    if (argc < 2) {
        printf("Usage: %s <max_threads>\n", argv[0]);
        return 1;
    }
```

```
int max_threads = atoi(argv[1]);
sem_init(&thread_limit, 0, max_threads);

FILE* f = fopen("input.txt", "r");
if (!f) {
    perror("Failed to open file");
    return 1;
}

int n;
fscanf(f, "%d", &n);

int* arr = malloc(n * sizeof(int));
for (int i = 0; i < n; i++) fscanf(f, "%d", &arr[i]);
fclose(f);

printf("Array size: %d\n", n);
printf("Max threads: %d\n", max_threads);

double start_time = get_time_ms();

sem_wait(&thread_limit);
if (max_threads > 1) {
    parallel_merge_sort(arr, 0, n - 1);
} else {
    merge_sort(arr, 0, n - 1);
}
sem_post(&thread_limit);

double end_time = get_time_ms();
```

```
    printf("Sorting time: %.2f ms\n", end_time - start_time);


    free(arr);

    sem_destroy(&thread_limit);


    return 0;

}
```

Протокол работы программы

Тестирование:

```
root@01d40df164ff:/workspace/lab2/src# ./main 2
Array size: 5000000
Max threads: 2
Sorting time: 517.84 ms
```

Strace:

```
7370  execve("./main", ["./main", "4"], 0x7ffd951a1da0 /* 29 vars */) = 0
7370  brk(NULL)                         = 0x3fc9b000
7370  mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x75beec3d3000
7370  access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
7370  openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
7370  fstat(3, {st_mode=S_IFREG|0644, st_size=29003, ...}) = 0
7370  mmap(NULL, 29003, PROT_READ, MAP_PRIVATE, 3, 0) = 0x75beec3cb000
7370  close(3)                          = 0
7370  openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
7370  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"..., 840, 64) =
840
7370  fstat(3, {st_mode=S_IFREG|0755, st_size=2003408, ...}) = 0
7370  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"..., 840, 64) =
840
7370  mmap(NULL, 2055800, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x75beec1d5000
7370  mmap(0x75beec1fd000, 1462272, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x75beec1fd000
7370  mmap(0x75beec362000, 352256, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x18d000) = 0x75beec362000
7370  mmap(0x75beec3b8000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e2000) = 0x75beec3b8000
7370  mmap(0x75beec3be000, 52856, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x75beec3be000
7370  close(3)                          = 0
7370  mmap(NULL, 12288, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x75beec1d2000
```

```
        7370  arch_prctl(ARCH_SET_FS, 0x75beec1d2740) = 0
        7370  set_tid_address(0x75beec1d2a10)   = 7370
        7370  set_robust_list(0x75beec1d2a20, 24) = 0
        7370  rseq(0x75beec1d2680, 0x20, 0, 0x53053053) = 0
        7370  mprotect(0x75beec3b8000, 16384, PROT_READ) = 0
        7370  mprotect(0x403000, 4096, PROT_READ) = 0
        7370  mprotect(0x75beec409000, 8192, PROT_READ) = 0
        7370  prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
        7370  munmap(0x75beec3cb000, 29003)     = 0
        7370  getrandom("\x0f\xae\x62\x25\x1f\x78\x4b\x1b", 8, GRND_NONBLOCK) = 8
        7370  brk(NULL)                         = 0x3fc9b000
        7370  brk(0x3fcbc000)                   = 0x3fcbc000
        7370  openat(AT_FDCWD, "input.txt", O_RDONLY) = 3
        7370  fstat(3, {st_mode=S_IFREG|0777, st_size=34444438, ...}) = 0
        7370  mmap(NULL, 20000768, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x75beeaebf000
        7370  close(3)                          = 0
        7370  fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
        7370  write(1, "Array size: 5000000\n", 20) = 20
        7370  write(1, "Max threads: 4\n", 15)  = 15
        7370  rt_sigaction(SIGRT_1, {sa_handler=0x75beec2654b0, sa_mask=[],
sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO,
sa_restorer=0x75beec214df0}, NULL, 8) = 0
        7370  rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
        7370  mmap(NULL, 8392704, PROT_NONE,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x75beea6be000
        7370  mprotect(0x75beea6bf000, 8388608, PROT_READ|PROT_WRITE) = 0
        7370  rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
        7370
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLO
NE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x75beeaebe990, parent_tid=0x75beeaebe990, exit_signal=0, stack=0x75beea6be000,
stack_size=0x7fff80, tls=0x75beeaebe6c0}, 88) = -1 ENOSYS (Function not implemented)
        7370  clone(child_stack=0x75beeaebdf70,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE
_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tid=[7400], tls=0x75beeaebe6c0, child_tidptr=0x75beeaebe990) = 7400
        7400  rseq(0x75beeaebe600, 0x20, 0, 0x53053053 <unfinished ...>
        7370  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
        7400  <... rseq resumed>)              = 0
        7370  <... rt_sigprocmask resumed>NULL, 8) = 0
        7400  set_robust_list(0x75beeaebe9a0, 24 <unfinished ...>
        7370  mmap(NULL, 8392704, PROT_NONE,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0 <unfinished ...>
        7400  <... set_robust_list resumed>)    = 0
        7370  <... mmap resumed>)               = 0x75bee9ebd000
        7400  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
        7370  mprotect(0x75bee9ebe000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>
        7400  <... rt_sigprocmask resumed>NULL, 8) = 0
        7370  <... mprotect resumed>)           = 0
        7400  mmap(NULL, 134217728, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0
<unfinished ...>
        7370  rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>
        7400  <... mmap resumed>)               = 0x75bee1ebd000
        7370  <... rt_sigprocmask resumed>[], 8) = 0
```

```
7400  munmap(0x75bee1ebd000, 34877440 <unfinished ...>
7370  clone(child_stack=0x75beea6bcf70,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE
_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID
<unfinished ...>
7400  <... munmap resumed>)            = 0
7370  <... clone resumed>, parent_tid=[7401], tls=0x75beea6bd6c0,
child_tidptr=0x75beea6bd990) = 7401
7400  munmap(0x75bee8000000, 32231424 <unfinished ...>
7370  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
7401  rseq(0x75beea6bd600, 0x20, 0, 0x53053053 <unfinished ...>
7370  <... rt_sigprocmask resumed>NULL, 8) = 0
7400  <... munmap resumed>)            = 0
7370  futex(0x75beeaebe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 7400,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
7401  <... rseq resumed>)              = 0
7400  mprotect(0x75bee4000000, 135168, PROT_READ|PROT_WRITE <unfinished ...>
7401  set_robust_list(0x75beea6bd9a0, 24 <unfinished ...>
7400  <... mprotect resumed>)          = 0
7401  <... set_robust_list resumed>)   = 0
7400  mmap(NULL, 8392704, PROT_NONE,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0 <unfinished ...>
7401  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
7400  <... mmap resumed>)              = 0x75bee96bc000
7401  <... rt_sigprocmask resumed>NULL, 8) = 0
7400  mprotect(0x75bee96bd000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>
7401  mmap(NULL, 134217728, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0
<unfinished ...>
7400  <... mprotect resumed>)          = 0
7401  <... mmap resumed>)              = 0x75bedc000000
7400  rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>
7401  munmap(0x75bee0000000, 67108864 <unfinished ...>
7400  <... rt_sigprocmask resumed>[], 8) = 0
7401  <... munmap resumed>)            = 0
7400  clone(child_stack=0x75bee9ebbf70,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE
_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID
<unfinished ...>
7401  mprotect(0x75bedc000000, 135168, PROT_READ|PROT_WRITE) = 0
7402  rseq(0x75bee9ebc600, 0x20, 0, 0x53053053 <unfinished ...>
7400  <... clone resumed>, parent_tid=[7402], tls=0x75bee9ebc6c0,
child_tidptr=0x75bee9ebc990) = 7402
7402  <... rseq resumed>)              = 0
7400  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
7402  set_robust_list(0x75bee9ebc9a0, 24 <unfinished ...>
7400  <... rt_sigprocmask resumed>NULL, 8) = 0
7402  <... set_robust_list resumed>)   = 0
7402  rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
7402  mmap(0x75bee0000000, 67108864, PROT_NONE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x75bee0000000
7402  mprotect(0x75bee0000000, 135168, PROT_READ|PROT_WRITE) = 0
7401  mprotect(0x75bedc021000, 28672, PROT_READ|PROT_WRITE) = 0
7400  mprotect(0x75bee4021000, 28672, PROT_READ|PROT_WRITE <unfinished ...>
7401  openat(AT_FDCWD, "/proc/sys/vm/overcommit_memory", O_RDONLY|O_CLOEXEC
<unfinished ...>
7400  <... mprotect resumed>)          = 0
```

```
7401  <... openat resumed>)            = 3
7401  close(3 <unfinished ...>
7402  mprotect(0x75bee0021000, 28672, PROT_READ|PROT_WRITE <unfinished ...>
7400  madvise(0x75bee4022000, 24576, MADV_DONTNEED <unfinished ...>
7402  <... mprotect resumed>)          = 0
7401  <... close resumed>)             = 0
7400  <... madvise resumed>)           = 0
7402  madvise(0x75bee0022000, 24576, MADV_DONTNEED <unfinished ...>
7401  madvise(0x75bedc022000, 24576, MADV_DONTNEED <unfinished ...>
7402  <... madvise resumed>)           = 0
7401  <... madvise resumed>)           = 0
7400  madvise(0x75bee4022000, 24576, MADV_DONTNEED) = 0
7402  madvise(0x75bee0022000, 24576, MADV_DONTNEED <unfinished ...>
7401  madvise(0x75bedc022000, 24576, MADV_DONTNEED <unfinished ...>
7400  mmap(NULL, 159744, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
7402  <... madvise resumed>)           = 0
7401  <... madvise resumed>)           = 0
7402  mmap(NULL, 159744, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
7400  <... mmap resumed>)              = 0x75bee9695000
7402  <... mmap resumed>)              = 0x75bee966e000
7401  mmap(NULL, 159744, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
7402  mmap(NULL, 159744, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
7400  mmap(NULL, 159744, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
7402  <... mmap resumed>)              = 0x75bee9620000
7401  <... mmap resumed>)              = 0x75bee9647000
7400  <... mmap resumed>)              = 0x75bee95f9000
7401  mmap(NULL, 159744, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x75bee95d2000
7402  munmap(0x75bee966e000, 159744)    = 0
7402  munmap(0x75bee9620000, 159744 <unfinished ...>
7400  munmap(0x75bee9695000, 159744 <unfinished ...>
7402  <... munmap resumed>)            = 0
7401  munmap(0x75bee9647000, 159744 <unfinished ...>
7400  <... munmap resumed>)            = 0
7401  <... munmap resumed>)            = 0
7400  munmap(0x75bee95f9000, 159744 <unfinished ...>
7401  munmap(0x75bee95d2000, 159744 <unfinished ...>
7400  <... munmap resumed>)            = 0
7401  <... munmap resumed>)            = 0
7402  mprotect(0x75bee0028000, 155648, PROT_READ|PROT_WRITE) = 0
7400  mprotect(0x75bee4028000, 155648, PROT_READ|PROT_WRITE <unfinished ...>
7401  mprotect(0x75bedc028000, 155648, PROT_READ|PROT_WRITE <unfinished ...>
7400  <... mprotect resumed>)          = 0
7401  <... mprotect resumed>)          = 0
7402  mmap(NULL, 315392, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x75bee966f000
7401  mmap(NULL, 315392, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
7400  mmap(NULL, 315392, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
7401  <... mmap resumed>)              = 0x75bee9622000
```

```
7400  <... mmap resumed>)             = 0x75bee95d5000
7402  munmap(0x75bee966f000, 315392)    = 0
7400  munmap(0x75bee95d5000, 315392 <unfinished ...>
7401  munmap(0x75bee9622000, 315392 <unfinished ...>
7400  <... munmap resumed>)             = 0
7401  <... munmap resumed>)             = 0
7402  mprotect(0x75bee004e000, 311296, PROT_READ|PROT_WRITE) = 0
7400  mprotect(0x75bee404e000, 315392, PROT_READ|PROT_WRITE <unfinished ...>
7401  mprotect(0x75bedc04e000, 311296, PROT_READ|PROT_WRITE <unfinished ...>
7400  <... mprotect resumed>)           = 0
7401  <... mprotect resumed>)           = 0
7402  mmap(NULL, 626688, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x75bee9623000
7400  mmap(NULL, 626688, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
7401  mmap(NULL, 626688, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
7400  <... mmap resumed>)             = 0x75bee958a000
7401  <... mmap resumed>)             = 0x75bee94f1000
7402  munmap(0x75bee9623000, 626688)    = 0
7401  munmap(0x75bee94f1000, 626688 <unfinished ...>
7400  munmap(0x75bee958a000, 626688 <unfinished ...>
7401  <... munmap resumed>)             = 0
7400  <... munmap resumed>)             = 0
7402  mprotect(0x75bee009a000, 626688, PROT_READ|PROT_WRITE) = 0
7401  mprotect(0x75bedc09a000, 626688, PROT_READ|PROT_WRITE) = 0
7400  mprotect(0x75bee409b000, 622592, PROT_READ|PROT_WRITE) = 0
7402  mmap(NULL, 1253376, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x75bee958a000
7401  mmap(NULL, 1253376, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x75bee9458000
7400  mmap(NULL, 1253376, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x75bee9326000
7402  munmap(0x75bee958a000, 1253376)   = 0
7401  munmap(0x75bee9458000, 1253376)   = 0
7400  munmap(0x75bee9326000, 1253376)   = 0
7402  mprotect(0x75bee0133000, 1249280, PROT_READ|PROT_WRITE) = 0
7401  mprotect(0x75bedc133000, 1249280, PROT_READ|PROT_WRITE) = 0
7400  mprotect(0x75bee4133000, 1249280, PROT_READ|PROT_WRITE) = 0
7402  mmap(NULL, 2502656, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x75bee9459000
7401  mmap(NULL, 2502656, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x75bee91f6000
7400  mmap(NULL, 2502656, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x75bee8f93000
7402  munmap(0x75bee9459000, 2502656)   = 0
7401  munmap(0x75bee91f6000, 2502656 <unfinished ...>
7402  rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
7402  madvise(0x75bee96bc000, 8368128, MADV_DONTNEED <unfinished ...>
7401  <... munmap resumed>)             = 0
7402  <... madvise resumed>)            = 0
7401  mmap(NULL, 8392704, PROT_NONE,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0 <unfinished ...>
7402  exit(0 <unfinished ...>
7401  <... mmap resumed>)             = 0x75bee8792000
7400  munmap(0x75bee8f93000, 2502656 <unfinished ...>
```

```
7402  <... exit resumed>)              = ?
7401  mprotect(0x75bee8793000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>
7402  +++ exited with 0 +++
7401  <... mprotect resumed>)          = 0
7400  <... munmap resumed>)            = 0
7401  rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>
7400  mmap(NULL, 5001216, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
7401  <... rt_sigprocmask resumed>[], 8) = 0
7400  <... mmap resumed>)              = 0x75bee91f7000
7401  clone(child_stack=0x75bee8f91f70,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE
_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID
<unfinished ...>
7400  mmap(NULL, 5001216, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
7404  rseq(0x75bee8f92600, 0x20, 0, 0x53053053 <unfinished ...>
7401  <... clone resumed>, parent_tid=[7404], tls=0x75bee8f926c0,
child_tidptr=0x75bee8f92990) = 7404
7400  <... mmap resumed>)              = 0x75bee82cd000
7404  <... rseq resumed>)              = 0
7401  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
7404  set_robust_list(0x75bee8f929a0, 24 <unfinished ...>
7401  <... rt_sigprocmask resumed>NULL, 8) = 0
7404  <... set_robust_list resumed>)    = 0
7404  rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
7400  munmap(0x75bee91f7000, 5001216)   = 0
7400  munmap(0x75bee82cd000, 5001216)   = 0
7400  rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
7400  madvise(0x75beea6be000, 8368128, MADV_DONTNEED) = 0
7400  exit(0)                          = ?
7370  <... futex resumed>)             = 0
7400  +++ exited with 0 +++
7370  futex(0x75beea6bd990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 7401,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
7401  rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
7401  clone(child_stack=0x75bee9ebbf70,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE
_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID
<unfinished ...>
7405  rseq(0x75bee9ebc600, 0x20, 0, 0x53053053 <unfinished ...>
7401  <... clone resumed>, parent_tid=[7405], tls=0x75bee9ebc6c0,
child_tidptr=0x75bee9ebc990) = 7405
7405  <... rseq resumed>)              = 0
7401  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
7405  set_robust_list(0x75bee9ebc9a0, 24 <unfinished ...>
7401  <... rt_sigprocmask resumed>NULL, 8) = 0
7405  <... set_robust_list resumed>)    = 0
7405  rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
7401  futex(0x75bee9ebc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 7405,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
7405  rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
7405  madvise(0x75bee96bc000, 8368128, MADV_DONTNEED) = 0
7405  exit(0)                          = ?
7405  +++ exited with 0 +++
7401  <... futex resumed>)             = 0
```

```
7404  rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
7404  clone(child_stack=0x75beeaebdf70,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE
_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tid=[7406], tls=0x75beeaebe6c0, child_tidptr=0x75beeaebe990) = 7406
7406  rseq(0x75beeaebe600, 0x20, 0, 0x53053053 <unfinished ...>
7404  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
7406  <... rseq resumed>)               = 0
7404  <... rt_sigprocmask resumed>NULL, 8) = 0
7406  set_robust_list(0x75beeaebe9a0, 24) = 0
7406  rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
7404  futex(0x75beeaebe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 7406,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
7406  rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
7406  madvise(0x75beea6be000, 8368128, MADV_DONTNEED) = 0
7406  exit(0 <unfinished ...>
7401  rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>
7406  <... exit resumed>)               = ?
7401  <... rt_sigprocmask resumed>[], 8) = 0
7406  +++ exited with 0 +++
7404  <... futex resumed>)              = 0
7401  clone(child_stack=0x75bee9ebbf70,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE
_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tid=[7407], tls=0x75bee9ebc6c0, child_tidptr=0x75bee9ebc990) = 7407
7407  rseq(0x75bee9ebc600, 0x20, 0, 0x53053053 <unfinished ...>
7401  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
7407  <... rseq resumed>)               = 0
7401  <... rt_sigprocmask resumed>NULL, 8) = 0
7407  set_robust_list(0x75bee9ebc9a0, 24) = 0
7407  rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
7401  futex(0x75bee9ebc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 7407,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
7407  rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
7407  madvise(0x75bee96bc000, 8368128, MADV_DONTNEED) = 0
7404  rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>
7407  exit(0 <unfinished ...>
7404  <... rt_sigprocmask resumed>[], 8) = 0
7407  <... exit resumed>)               = ?
7404  clone(child_stack=0x75beeaebdf70,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE
_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID
<unfinished ...>
7407  +++ exited with 0 +++
7401  <... futex resumed>)              = 0
7408  rseq(0x75beeaebe600, 0x20, 0, 0x53053053 <unfinished ...>
7404  <... clone resumed>, parent_tid=[7408], tls=0x75beeaebe6c0,
child_tidptr=0x75beeaebe990) = 7408
7408  <... rseq resumed>)               = 0
7404  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
7408  set_robust_list(0x75beeaebe9a0, 24 <unfinished ...>
7404  <... rt_sigprocmask resumed>NULL, 8) = 0
7408  <... set_robust_list resumed>)    = 0
7408  rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
7404  futex(0x75beeaebe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 7408,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
```

```
7408  rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
7408  madvise(0x75beea6be000, 8368128, MADV_DONTNEED) = 0
7408  exit(0)                           = ?
7404  <... futex resumed>)              = 0
7408  +++ exited with 0 +++
7401  rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
7401  clone(child_stack=0x75beeaebdf70,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE
_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tid=[7409], tls=0x75beeaebe6c0, child_tidptr=0x75beeaebe990) = 7409
7409  rseq(0x75beeaebe600, 0x20, 0, 0x53053053 <unfinished ...>
7401  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
7409  <... rseq resumed>)               = 0
7401  <... rt_sigprocmask resumed>NULL, 8) = 0
7409  set_robust_list(0x75beeaebe9a0, 24) = 0
7409  rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
7404  rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
7404  madvise(0x75bee8792000, 8368128, MADV_DONTNEED) = 0
7404  exit(0)                           = ?
7404  +++ exited with 0 +++
7401  rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
7401  clone(child_stack=0x75bee9ebbf70,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE
_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID
<unfinished ...>
7410  rseq(0x75bee9ebc600, 0x20, 0, 0x53053053 <unfinished ...>
7401  <... clone resumed>, parent_tid=[7410], tls=0x75bee9ebc6c0,
child_tidptr=0x75bee9ebc990) = 7410
7410  <... rseq resumed>)               = 0
7401  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
7410  set_robust_list(0x75bee9ebc9a0, 24 <unfinished ...>
7401  <... rt_sigprocmask resumed>NULL, 8) = 0
7410  <... set_robust_list resumed>)    = 0
7410  rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
7401  futex(0x75bee9ebc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 7410,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
7410  rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
7410  madvise(0x75bee96bc000, 8368128, MADV_DONTNEED) = 0
7410  exit(0 <unfinished ...>
7409  rt_sigprocmask(SIG_BLOCK, ~[RT_1],  <unfinished ...>
7410  <... exit resumed>)               = ?
7409  <... rt_sigprocmask resumed>NULL, 8) = 0
7410  +++ exited with 0 +++
7401  <... futex resumed>)              = 0
7409  madvise(0x75beea6be000, 8368128, MADV_DONTNEED) = 0
7409  exit(0)                           = ?
7401  futex(0x75beeaebe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 7409,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
7409  +++ exited with 0 +++
7401  <... futex resumed>)              = -1 EAGAIN (Resource temporarily unavailable)
7401  mprotect(0x75bedc264000, 2502656, PROT_READ|PROT_WRITE) = 0
7401  mprotect(0x75bedc4c7000, 4997120, PROT_READ|PROT_WRITE) = 0
7401  rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
7401  madvise(0x75bee9ebd000, 8368128, MADV_DONTNEED) = 0
7401  exit(0)                           = ?
7370  <... futex resumed>)              = 0
```

7401 +++ exited with 0 +++
7370 mmap(NULL, 10002432, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x75bedb676000
7370 mmap(NULL, 10002432, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x75bedacec000
7370 munmap(0x75bedb676000, 10002432) = 0
7370 munmap(0x75bedacec000, 10002432) = 0
7370 write(1, "Sorting time: 477.49 ms\n", 24) = 24
7370 munmap(0x75beeaebf000, 20000768) = 0
7370 exit_group(0) = ?
7370 +++ exited with 0 +++

Для N = 5 000 000

Число потоков	**Время исполнения(мс)**	**Ускорение**	**Эффективность**
1	840,44	1,00	1,00
2	510,53	1,65	0,82
4	395,91	2,12	0,53
8	312,91	2,69	0,34

Вывод

- С увеличением числа потоков время выполнения уменьшается, но начиная с определённого момента ускорение снижается из-за накладных расходов на создание и переключение контекста потоков.
- Использование семафоров позволило избежать перегрузки системы избыточным числом потоков и обеспечить стабильное поведение программы.
- Для мелких или средних массивов накладные расходы на управление потоками могут превышать выгоду от параллельных вычислений.