

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: Лисов Д.С.

Преподаватель: Миронов Е.С.

Оценка: \_\_\_\_\_

Дата: 22.10.25

Москва, 2025

# Постановка задачи

## Вариант 19.

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.

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

Дан массив координат (x, y). Пользователь вводит число кластеров. Проведите кластеризацию методом k-средних

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

## Общий метод и алгоритм решения

Использованные системные вызовы:

- `clock_gettime(CLOCK_MONOTIC, &start);` – получение монотонного времени.
- `pthread_create();` – создание потоков
- `pthread_join();` - ожидание завершения потоков
- `pthread_mutex_lock() / pthread_mutex_unlock();` - работа с мьютексами (блокировка, разблокировка).

Сначала выбираются k центроидов из k точек в списке. Далее для каждой точки находится ближайший центроид. Потом постоянно обновляем центроиды до тех пор, пока они изменяются. Для каждого центроида создаётся отдельный поток. Для наборов из (n / NUM\_THREADS) точек создаются отдельные потоки.

## Код программы

### main.c

```
#include <stdlib.h>
#include <stdio.h>
#include <semaphore.h>
#include <pthread.h>
#include <unistd.h>
#include <math.h>
#include <time.h>
#include <string.h>
```

```
int MAX_THREADS = 4;
```

```
int min(int x, int y) {
    if (x < y) return x;
    else return y;
}
```

```
pthread_t* threads;
pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
pthread_barrier_t barrier;
```

```
typedef struct {
    double x, y;
} Point;
```

```
typedef struct {
    double x, y;
    int count;
} Centroid;
```

```
Point* points;
int num_points;
Centroid* centroids;
```

```
typedef struct {
    int start;
    int end;
    int *changed;
    pthread_mutex_t* changed_mutex;
} Thread_data;
```

```
int* centroid_ids, *new_centroid_ids;
int k, n;
```

```
double distance(Point a, Point b) {
    double dx = a.x - b.x;
    double dy = a.y - b.y;
    return sqrt(dx * dx + dy * dy);
}
```

```
double _distance(Point a, Centroid b) {
    double dx = a.x - b.x;
    double dy = a.y - b.y;
    return sqrt(dx * dx + dy * dy);
}
```

```
void *compute_clusters(void *arg) {
    int i = *((int*)arg);
```

```

    free(arg);
    double sum_x = 0.0, sum_y = 0.0;
    int count = 0;
    for (int j = 0; j < n; ++j) {
        if (centroid_ids[j] == i) {
            sum_x += points[j].x;
            sum_y += points[j].y;
            ++count;
        }
    }

    if (count > 0) {
        centroids[i].x = sum_x / count;
        centroids[i].y = sum_y / count;
    }

    return NULL;
}

void* recalculate_centroids(void *arg) {
    Thread_data* td = (Thread_data*)arg;
    for (int j = td->start; j < td->end; ++j) {
        new_centroid_ids[j] = centroid_ids[j];
        for (int i = 0; i < k; ++i) {
            if (_distance(points[j], centroids[i]) < _distance(points[j],
                centroids[new_centroid_ids[j]])) {
                new_centroid_ids[j] = i;
            }
        }
        if (centroid_ids[j] != new_centroid_ids[j]) {
            pthread_mutex_lock(td->changed_mutex);
            *(td->changed) = 1;
            pthread_mutex_unlock(td->changed_mutex);
        }
    }

    return NULL;
}

int main(int argc, char *argv[]) {
    for (int i = 1; i < argc; i++) {
        if (strcmp(argv[i], "-t") == 0 && i + 1 < argc) {
            MAX_THREADS = atoi(argv[i + 1]);
            if (MAX_THREADS <= 0) {
                fprintf(stderr, "Ошибка: количество потоков должно быть
положительным числом\n");
                return 1;
            }
        }
    }
}

```

```

        }
        i++;
    }
    else {
        fprintf(stderr, "Неизвестный аргумент: %s\n", argv[i]);
        return 1;
    }
}

scanf("%d\n%d", &k, &n);
centroids = (Centroid*)malloc(sizeof(Centroid) * k);
points = (Point*)malloc(sizeof(Point) * n);
centroid_ids = (int*)malloc(sizeof(int) * n);

threads = (pthread_t*)malloc(sizeof(pthread_t) * k);

pthread_mutex_t changed_mutex = PTHREAD_MUTEX_INITIALIZER;

for (int i = 0; i < n; ++i) {
    scanf("%lf %lf", &points[i].x, &points[i].y);
}

struct timespec start, end;
clock_gettime(CLOCK_MONOTONIC, &start);

for (int i = 0; i < k; ++i) {
    centroids[i].x = points[i].x;
    centroids[i].y = points[i].y;
    centroids[i].count = 1;
    centroid_ids[i] = i;
}

for (int i = k; i < n; ++i) {
    int idx = 0;
    for (int j = 0; j < k; ++j) {
        if (distance(points[i], centroids[j]) < _distance(points[i],
centroids[idx])) {
            idx = j;
        }
    }
    centroid_ids[i] = idx;
    centroids[idx].count++;
}

new_centroid_ids = (int*)malloc(sizeof(int) * n);

```

```

while (1) {
    int t = 0;

    printf("Создано потоков для пересчета центроидов: %d\n", k);
    for (int i = 0; i < k; ++i) {
        int *arg = (int*)malloc(sizeof(int));
        *arg = i;
        pthread_create(&threads[i], NULL, compute_clusters, arg);
    }
    for (int i = 0; i < k; ++i) {
        pthread_join(threads[i], NULL);
    }

    int num_threads = min(MAX_THREADS, n); // Не больше потоков, чем точек
    printf("Создано потоков для перераспределения точек: %d (ограничение: %d)\n", num_threads, MAX_THREADS);

    pthread_t *assign_threads = (pthread_t*)malloc(num_threads *
sizeof(pthread_t));
    Thread_data *args = (Thread_data*) malloc(num_threads *
sizeof(Thread_data));

    int chunk = n / num_threads;
    for (int th = 0; th < num_threads; ++th) {
        args[th].start = th * chunk;
        args[th].end = (th == num_threads - 1) ? n : (th + 1) * chunk;
        args[th].changed = &t;
        args[th].changed_mutex = &changed_mutex;

        pthread_create(&assign_threads[th], NULL, recalculate_centroids,
&args[th]);
    }

    for (int th = 0; th < num_threads; ++th) {
        pthread_join(assign_threads[th], NULL);
    }

    for (int i = 0; i < n; ++i) {
        centroid_ids[i] = new_centroid_ids[i];
    }
    for (int i = 0; i < k; ++i) {
        centroids[i].count = 0;
    }
    for (int i = 0; i < n; ++i) {
        centroids[centroid_ids[i]].count++;
    }

    free(assign_threads);
    free(args);
}

```

```

        if (!t)
            break;
    }

    clock_gettime(CLOCK_MONOTONIC, &end);
    double time_sec = (end.tv_sec - start.tv_sec) + (end.tv_nsec -
start.tv_nsec) / 1e9;

    for (int i = 0; i < k; ++i) {
        printf("%f %f\n", centroids[i].x, centroids[i].y);
    }

    printf("Время выполнения: %lf секунд\n", time_sec);
    printf("Максимальное количество потоков: %d\n", MAX_THREADS);

    free(centroid_ids);
    free(centroids);
    free(points);
    free(new_centroid_ids);
    free(threads);
    return 0;
}

```

## Протокол работы программы

### Тестирование:

```
$ ./a.out -t 4
```

```
2 4
```

```
1 1
```

```
2 2
```

```
3 3
```

```
4 4
```

```
Создано потоков для пересчета центроидов: 2
```

```
Создано потоков для перераспределения точек: 4 (ограничение: 4)
```

```
1.000000 1.000000
```

```
3.000000 3.000000
```

```
Время выполнения: 0.000934 секунд
```

```
Максимальное количество потоков: 4
```

### Strace:

```
$ strace -f ./a.out -t 4
```

```
execve("./a.out", ["/a.out", "-t", "4"], 0x7ffe273340a8 /* 62 vars */) = 0
```

```
brk(NULL) = 0x55f15bfa9000
```

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

```
0x7dbfca2f8000
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (Нет такого файла или каталога)
```

```

openat(AT_FDCWD, "/etc/ld.so.cache", 0_RDONLY|0_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=149367, ...}) = 0
mmap(NULL, 149367, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7dbfca2d3000
close(3) = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.so.6", 0_RDONLY|0_CLOEXEC) = 3
832) 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"...,
fstat(3, {st_mode=S_IFREG|0644, st_size=952616, ...}) = 0
mmap(NULL, 950296, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7dbfca1ea000
mmap(0x7dbfca1fa000, 520192, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x10000) = 0x7dbfca1fa000
mmap(0x7dbfca279000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x8f000) = 0x7dbfca279000
mmap(0x7dbfca2d1000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0xe7000) = 0x7dbfca2d1000
close(3) = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", 0_RDONLY|0_CLOEXEC) = 3
0\1\0\0\0\220\243\2\0\0\0\0\0\0\0\0\0\0\0\3\0>\
832) read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\
832) = 832
784, 64) = 784, "\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"...,
fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0
784, 64) = 784, "\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"...,
0x7dbfc9e0000, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
mmap(0x7dbfc9e28000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x28000) = 0x7dbfc9e28000
mmap(0x7dbfc9fb0000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1b0000) = 0x7dbfc9fb0000
mmap(0x7dbfc9fff000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x1fe000) = 0x7dbfc9fff000
mmap(0x7dbfca005000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_ANONYMOUS, -1, 0) = 0x7dbfca005000
close(3) = 0
0x7dbfca1e7000, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
arch_prctl(ARCH_SET_FS, 0x7dbfca1e7740) = 0
set_tid_address(0x7dbfca1e7a10) = 9859
set_robust_list(0x7dbfca1e7a20, 24) = 0
rseq(0x7dbfca1e8060, 0x20, 0, 0x53053053) = 0
mprotect(0x7dbfc9fff000, 16384, PROT_READ) = 0
mprotect(0x7dbfca2d1000, 4096, PROT_READ) = 0
mprotect(0x55f13da4d000, 4096, PROT_READ) = 0
mprotect(0x7dbfca330000, 8192, PROT_READ) = 0
rlim_max=RLIM64_INFINITY, {rlim_cur=8192*1024,
rlimit(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
munmap(0x7dbfca2d3000, 149367) = 0
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
getrandom("\x2d\x8a\x66\x10\x60\x60\x7d\x3d", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x55f15bfa9000
brk(0x55f15bfca000) = 0x55f15bfca000
read(0, 2 4
"2 4\n", 1024) = 4
read(0, 1 1
"1 1\n", 1024) = 4
read(0, 2 2
"2 2\n", 1024) = 4

```



```

read(0, 3 3
"3 3\n", 1024)          = 4
read(0, 4 4
"4 4\n", 1024)          = 4
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, "\320\241\320\276\320\267\320\264\320\260\320\275\320\276\320\277\320\278\320\272\320\276\320\262\320\264...", 80) = 80
Потоков для пересчета центроидов: 2
) = 80
rt_sigaction(SIGRT_1, {sa_handler=0x7dbfc9e99530, sa_mask=[],
sa_flags=SA_RESTORER|SA_UNSTACK|SA_RESTART|SA_SIGINFO, sa_restorer=0x7dbfc9e45330},
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) =
0x7dbfc95ff000
mprotect(0x7dbfc9600000, 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=0x7dbfc9dff990, parent_tid=0x7dbfc9dff990, exit_signal=0,
stack=0x7dbfc95ff000, stack_size=0x7fff80, tls=0x7dbfc9dff6c0}, strace: Process 9861
attached
=> {parent_tid=[9861]}, 88) = 9861
[pid 9861] rseq(0x7dbfc9dffe0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 9859] rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
[pid 9861] <... rseq resumed>) = 0
[pid 9859] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,
-1, 0) = 0x7dbfc8dff000
[pid 9861] set_robust_list(0x7dbfc9dff9a0, 24 <unfinished ...>
[pid 9859] mprotect(0x7dbfc8dff000, 8388608, PROT_READ|PROT_WRITE
<unfinished ...>
[pid 9861] <... set_robust_list resumed>) = 0
[pid 9859] <... mprotect resumed>) = 0
[pid 9859] rt_sigprocmask(SIG_BLOCK, ~[], <unfinished ...>
[pid 9861] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9859] <... rt_sigprocmask resumed>[], 8) = 0
[pid 9859] clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|
CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x7dbfc95fe990, parent_tid=0x7dbfc95fe990, exit_signal=0,
stack=0x7dbfc8dff000, stack_size=0x7fff80, tls=0x7dbfc95fe6c0}, <unfinished ...>
[pid 9861] <... rt_sigprocmask resumed>NULL, 8) = 0
strace: Process 9862 attached
[pid 9861] mmap(NULL, 134217728, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0
<unfinished ...>
[pid 9859] <... clone3 resumed> => {parent_tid=[9862]}, 88) = 9862
[pid 9862] rseq(0x7dbfc95fefe0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 9861] <... mmap resumed>) = 0x7dbfc0c00000
[pid 9859] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9862] <... rseq resumed>) = 0
[pid 9859] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9861] munmap(0x7dbfc0c00000, 54525952 <unfinished ...>
[pid 9862] set_robust_list(0x7dbfc95fe9a0, 24 <unfinished ...>
[pid 9859] futax(0x7dbfc9dff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9861,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
[pid 9862] <... set_robust_list resumed>) = 0
[pid 9861] <... munmap resumed>) = 0
[pid 9862] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9861] munmap(0x7dbfc8000000, 12582912 <unfinished ...>
[pid 9862] <... rt_sigprocmask resumed>NULL, 8) = 0

```

```

[pid 9862] mmap(NULL, 134217728, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0
<unfinished ...>
[pid 9861] <... munmap resumed>) = 0
[pid 9862] <... mmap resumed>) = 0x7dbfbc000000
[pid 9861] mprotect(0x7dbfc4000000, 135168, PROT_READ|PROT_WRITE
<unfinished ...>
[pid 9862] munmap(0x7dbfc0000000, 67108864 <unfinished ...>
[pid 9861] <... mprotect resumed>) = 0
[pid 9862] <... munmap resumed>) = 0
[pid 9861] mprotect(0x7dbfbc000000, 135168, PROT_READ|PROT_WRITE
<unfinished ...>
[pid 9861] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>
[pid 9862] <... mprotect resumed>) = 0
[pid 9861] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9861] madvise(0x7dbfc95ff000, 8368128, MADV_DONTNEED <unfinished ...>
[pid 9862] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>
[pid 9861] <... madvise resumed>) = 0
[pid 9862] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9862] madvise(0x7dbfc8dfe000, 8368128, MADV_DONTNEED <unfinished ...>
[pid 9861] exit(0 <unfinished ...>
[pid 9862] <... madvise resumed>) = 0
[pid 9861] <... exit resumed>) = ?
[pid 9862] exit(0 <unfinished ...>
[pid 9861] +++ exited with 0 +++
[pid 9859] <... futex resumed>) = 0
[pid 9862] <... exit resumed>) = ?
[pid 9859] futex(0x7dbfc95fe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9862,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
[pid 9862] +++ exited with 0 +++
[pid 9859] futex resumed>) = -1 EAGAIN (Ресурс временно
недоступен)
[pid 9862] write(1, "\320\276\320\276\320\276\320\267\320\264\320\260\320\275\320\276\
320\277\320\276\320\276\320\276\320\272\320\276\320\267\320\264...", 114Создано
потоков для перераспределения точек: 4 (ограничение: 4)
) = 114
[pid 9862] rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
[pid 9863] clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|
CLONE_NEWNS|CLONE_NEWPID|CLONE_NEWUTS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x7dbfc95fe990, parent_tid=0x7dbfc95fe990, child_signal=0,
stack=0x7dbfc8dfe000, stack_size=0x7fff80, tls=0x7dbfc95fe6c0}, strace: Process 9863
attached
=> {parent_tid=[9863]}, 88) = 9863
[pid 9863] rseq(0x7dbfc95fefe0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 9859] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9863] <... rseq resumed>) = 0
[pid 9859] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9863] set_robust_list(0x7dbfc95fe9a0, 24 <unfinished ...>
[pid 9859] rt_sigprocmask(SIG_BLOCK, ~[], <unfinished ...>
[pid 9863] <... set_robust_list resumed>) = 0
[pid 9859] <... rt_sigprocmask resumed>[], 8) = 0
[pid 9863] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9859] clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|
CLONE_NEWNS|CLONE_NEWPID|CLONE_NEWUTS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x7dbfc95fe990, parent_tid=0x7dbfc95fe990, child_signal=0,
stack=0x7dbfc95ff000, stack_size=0x7fff80, tls=0x7dbfc9dfe6c0}, <unfinished ...>
[pid 9863] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9863] rt_sigprocmask(SIG_BLOCK, ~[RT_1], strace: Process 9864 attached
NULL, 8) = 0

```

```

[pid 9859] <... clone3 resumed> => {parent_tid=[9864]}, 88) = 9864
[pid 9864] rseq(0x7dbfc9dffe0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 9859] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9863] madvise(0x7dbfc8dfe000, 8368128, MADV_DONTNEED <unfinished ...>
[pid 9859] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9864] <... rseq resumed>) = 0
[pid 9859] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,
-1, 0 <unfinished ...>
[pid 9863] <... madvise resumed>) = 0
[pid 9864] set_robust_list(0x7dbfc9dff9a0, 24 <unfinished ...>
[pid 9859] <... mmap resumed>) = 0x7dbfc85fd000
[pid 9864] <... set_robust_list resumed>) = 0
[pid 9863] exit(0 <unfinished ...>
[pid 9859] mprotect(0x7dbfc85fe000, 8388608, PROT_READ|PROT_WRITE
<unfinished ...>
[pid 9864] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9859] <... mprotect resumed>) = 0
[pid 9863] <... exit resumed>) = ?
[pid 9864] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9859] rt_sigprocmask(SIG_BLOCK, ~[], <unfinished ...>
[pid 9863] +++ exited with 0 +++
[pid 9859] <... rt_sigprocmask resumed>[], 8) = 0
[pid 9864] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>
[pid 9859] clone3(<flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|
CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x7dbfc85fd000, parent_tid=0x7dbfc85fd000, exit_signal=0,
stack=0x7dbfc85fd000, stack_size=0x7fff80, tls=0x7dbfc8dfe000> <unfinished ...>
[pid 9864] <... rt_sigprocmask resumed>NULL, 8) = 0
strace: Process 9865 attached
[pid 9859] <... clone3 resumed> => {parent_tid=[9865]}, 88) = 9865
[pid 9865] rseq(0x7dbfc8dfdfe0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 9859] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9865] <... rseq resumed>) = 0
[pid 9864] madvise(0x7dbfc95ff000, 8368128, MADV_DONTNEED <unfinished ...>
[pid 9859] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9865] set_robust_list(0x7dbfc8dfd9a0, 24 <unfinished ...>
[pid 9859] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,
-1, 0 <unfinished ...>
[pid 9864] <... madvise resumed>) = 0
[pid 9859] <... mmap resumed>) = 0x7dbfc37ff000
[pid 9865] <... set_robust_list resumed>) = 0
[pid 9859] mprotect(0x7dbfc3800000, 8388608, PROT_READ|PROT_WRITE
<unfinished ...>
[pid 9864] exit(0 <unfinished ...>
[pid 9865] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9859] <... mprotect resumed>) = 0
[pid 9864] <... exit resumed>) = ?
[pid 9865] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9859] rt_sigprocmask(SIG_BLOCK, ~[], <unfinished ...>
[pid 9864] +++ exited with 0 +++
[pid 9859] <... rt_sigprocmask resumed>[], 8) = 0
[pid 9865] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>
[pid 9859] clone3(<flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|
CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,

```

```

child_tid=0x7dbfc3fff990, parent_tid=0x7dbfc3fff990, exit_signal=0
stack=0x7dbfc37ff000, stack_size=0x7fff80, tls=0x7dbfc3fff6c0} <unfinished ...>
[pid 9865] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9865] madvise(0x7dbfc85fd000, 8368128, MADV_DONTNEEDstrace: Process 9866
attached
) = 0
[pid 9859] <... clone3 resumed> => {parent_tid=[9866]}, 88) = 9866
[pid 9866] rseq(0x7dbfc3ffffe0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 9865] exit(0 <unfinished ...>
[pid 9859] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9866] <... rseq resumed>) = 0
[pid 9865] <... exit resumed>) = ?
[pid 9859] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9866] set_robust_list(0x7dbfc3fff9a0, 24 <unfinished ...>
[pid 9859] futex(0x7dbfc3fff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9866,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
[pid 9865] +++ exited with 0 +++
[pid 9866] <... set_robust_list resumed>) = 0
[pid 9866] rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
[pid 9866] rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
[pid 9866] madvise(0x7dbfc37ff000, 8368128, MADV_DONTNEED) = 0
[pid 9866] exit(0) = ?
[pid 9866] +++ exited with 0 +++
<... futex resumed>) = 0
write(1, "1.000000 1.000000\n", 181.000000 1.000000
) = 18
write(1, "3.000000 3.000000\n", 183.000000 3.000000
) = 18
write(1, "320\262\321\213\320\277\320\276\320\273\320\275\320\265\320\274\321\217\
, 55Время выполнения: 0.007517 секунд
) = 55
write(1, "320\262\321\213\320\277\320\276\320\273\320\275\320\265\320\274\321\217\
, 64Максимальное количество потоков: 4
) = 64
lseek(0, -1, SEEK_CUR) = -1 ESPIPE (Недопустимая операция
смещения)
exit_group(0) = ?
+++ exited with 0 +++

```

Потоков	Время	Ускорение	Эффективность
1	0.012190	1	1
2	0.011650	1,05	0,525
4	0.006543	1,86	0,465
8	0.005613	2,17	0,271

## **Вывод**

**Лабораторная работа показалась достаточно интересной и важной, поскольку научился работать с многопоточностью. Столкнулся с проблемами на ограничения количества потоков в своей ОС Linux.**