

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-211Б-23

Студент: Ласточкин М.В.

Преподаватель: Бахарев В.Д.

Оценка:

Дата: 03.12.24

Постановка задачи

Цель работы:

Целью является приобретение практических навыков в:

- Управление потоками в ОС
- Обеспечение синхронизации между потоками

Задание:

Составить программу на языке Си, обрабатывающую данные в многопоточном режиме. При обработке 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. В отчете привести исследование зависимости ускорения и эффективности алгоритма от входных данных и количества потоков. Получившиеся результаты необходимо объяснить.

Вариант 11. Наложить K раз медианный фильтр на матрицу, состоящую из целых чисел. Размер окна задается пользователем

Общий метод и алгоритм решения

Использованные системные вызовы:

- `ssize_t write(STDOUT_FILENO, const char buffer, size_t offset);` – Записывает `offset` байт из буфера в стандартный поток вывода файл. Возвращает количество записанных байт или `-1`.
- `int sem_wait(sem_t *semaphore);` уменьшает значение семафора (`semaphore`), если значение = 0, то вызов блокируется до тех пор, пока нельзя будет выполнить вычитание (пока не произойдет `sem_post`)
- `int sem_post(sem_t *semaphore);` увеличивает значение семафора (`semaphore`) на единицу. Работает в паре с `sem_wait`.
- `int sem_destroy(sem_t *semaphore);` уничтожает семафор, на который указывает `semaphore`.
- `int sem_init(sem_t *semaphore, int (0), unsigned int max_threads);` инициализирует семафор по адресу на который указывает `semaphore`. Второй аргумент отвечает за то, каким им пользоваться. Если значение = 0, то семафор является общим для потоков процесса, иначе он общий для процессов.
- `int pthread_create((pthread_t *thread, const pthread_attr_t *attr, void *(*routine) (void *), void *arg);` Создает поток с начальной функцией и заданными аргументами.
- `int pthread_join(pthread_t threads, void ** value);` Дождется завершения потока

Программа получает на вход три аргумента – размер окна, количество итераций и максимальное количество потоков. `Window_size` определяет размер квадрата (окна), который используется для вычисления медианы вокруг каждой точки матрицы. Это окно всегда должно

быть нечетного размера для того, чтобы было однозначно выделить центральный элемент. Iterations отвечает за то, сколько раз будет выполнен медианный фильтр для матрицы. Max_threads - максимальное количество потоков. Определяет, на сколько частей делится обработка матрицы, чтобы разные части обрабатывались параллельно.

После полученных значений и обработки на то, что они введены корректно, генерируется матрица размера 10x10.

После создается нужное количество потоков для обработки каждого медианного фильтра у матрицы. Для промежуточных вычислений создает временная матрица, из которой впоследствии будут переписаны данные в исходную матрицу, и уже готовая матрица выводится пользователю.

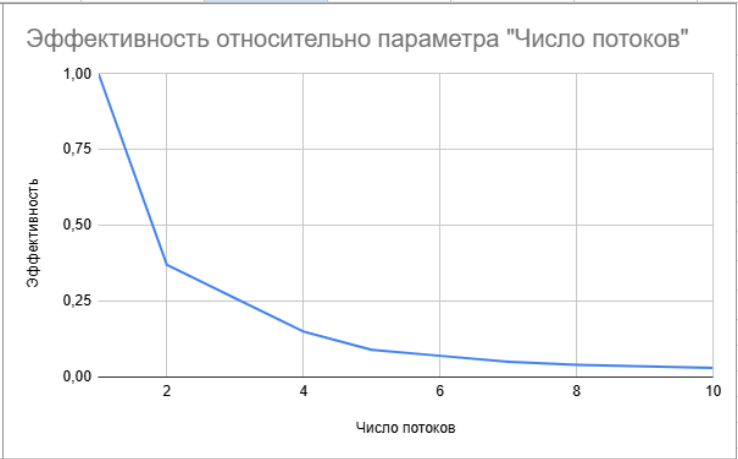
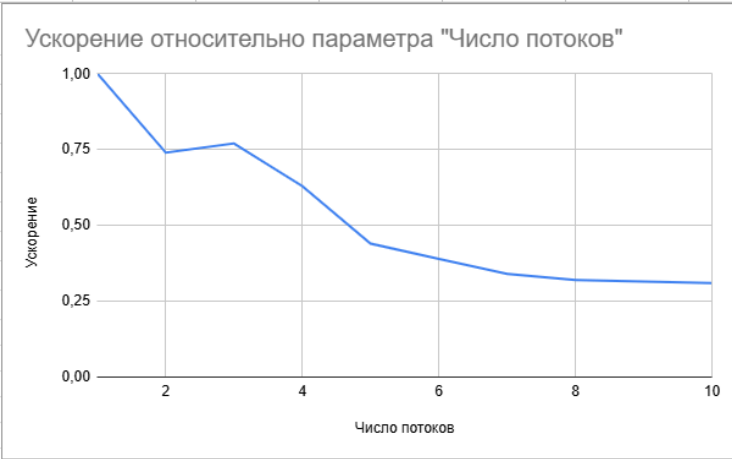
Суть медианного метода:

1. Формируется окно вокруг центрального элемента.
2. Из окна берутся все элементы.
3. Находится медиана (центральное значение) из отсортированного списка чисел окна.
4. Центральное значение элемента заменяется на найденную медиану.

Ниже приведены данные, показывающие изменения ускорения и эффективности, с разным количеством потоков, для этой реализации.

Число потоков	Время выполнения	Ускорение	Эффективность
1	169	1,00	1,00
2	227	0,74	0,37
3	219	0,77	0,26
4	267	0,63	0,15
5	382	0,44	0,09
6	427	0,39	0,07
7	487	0,34	0,05
8	515	0,32	0,04
10	535	0,31	0,03

Количество раундов	Время выполнения(мс)
100	136
625	256
10000	1743
250000	46857
1000000	220613
100000000	19563182



Код программы

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

#define ROWS 10
#define COLS 10

int matrix[ROWS][COLS];
int temp_matrix[ROWS][COLS];

typedef struct {
    int start_row;
    int end_row;
    int window_size;
} ThreadArgs;

int compare(const void* a, const void* b);

int find_median(int* window, int size);

void* median_filter(void* args);

void copy_temp_to_matrix();

void generate_matrix();

void print_matrix(int mat[ROWS][COLS]);

sem_t start_sem;
sem_t end_sem;

int stop_threads = 0;

int main(int argc, char* argv[]) {
    if (argc != 4) {
        fprintf(stderr, "Usage: ./a.out <window_size> <iterations>
<max_threads>\n");
        return EXIT_FAILURE;
    }

    int window_size = atoi(argv[1]);
    int iterations = atoi(argv[2]);
    int max_threads = atoi(argv[3]);

    if (window_size % 2 == 0 || window_size < 1 || iterations < 1 ||
max_threads < 1) {
```

```

        fprintf(stderr, "Invalid arguments. Window size must be odd and >=
1. Iterations and threads > 0.\n");
        return EXIT_FAILURE;
    }

    if (max_threads > ROWS) {
        max_threads = ROWS;
    }

    generate_matrix();

    printf("Original matrix:\n");
    print_matrix(matrix);

    pthread_t threads[max_threads];
    ThreadArgs thread_args[max_threads];
    int rows_per_thread = ROWS / max_threads;

    sem_init(&start_sem, 0, 0);
    sem_init(&end_sem, 0, 0);

    for (int i = 0; i < max_threads; i++) {
        thread_args[i].start_row = i * rows_per_thread;
        thread_args[i].end_row = (i == max_threads - 1) ? ROWS : (i + 1) *
rows_per_thread;
        thread_args[i].window_size = window_size;

        if (pthread_create(&threads[i], NULL, median_filter,
&thread_args[i]) != 0) {
            perror("pthread_create failed");
            return EXIT_FAILURE;
        }
    }

    clock_t start_time = clock();

    for (int iter = 0; iter < iterations; iter++) {
        for (int i = 0; i < max_threads; i++) {
            sem_post(&start_sem);
        }

        for (int i = 0; i < max_threads; i++) {
            sem_wait(&end_sem);
        }

        copy_temp_to_matrix();
    }

    stop_threads = 1;
    for (int i = 0; i < max_threads; i++) {
        sem_post(&start_sem);
    }

```

```

    for (int i = 0; i < max_threads; i++) {
        pthread_join(threads[i], NULL);
    }

    sem_destroy(&start_sem);
    sem_destroy(&end_sem);

    clock_t end_time = clock();

    printf("Result matrix:\n");
    print_matrix(matrix);

    double time_spent = (double)(end_time - start_time) / CLOCKS_PER_SEC;
    printf("Time taken: %f seconds\n", time_spent);

    return EXIT_SUCCESS;
}

int compare(const void* a, const void* b) {
    return (*(int*)a - *(int*)b);
}

int find_median(int* window, int size) {
    qsort(window, size, sizeof(int), compare);
    return window[size / 2];
}

void* median_filter(void* args) {
    ThreadArgs* thread_args = (ThreadArgs*)args;
    int start_row = thread_args->start_row;
    int end_row = thread_args->end_row;
    int window_size = thread_args->window_size;
    int offset = window_size / 2;

    while (1) {
        sem_wait(&start_sem);

        if (stop_threads) {
            break;
        }

        for (int i = start_row; i < end_row; i++) {
            for (int j = 0; j < COLS; j++) {
                if (i < offset || i >= ROWS - offset || j < offset || j >=
COLS - offset) {
                    temp_matrix[i][j] = matrix[i][j];
                } else {
                    int window[window_size * window_size];
                    int idx = 0;
                    for (int wi = -offset; wi <= offset; wi++) {
                        for (int wj = -offset; wj <= offset; wj++) {
                            window[idx++] = matrix[i + wi][j + wj];
                        }
                    }
                }
            }
        }
    }
}

```



```

        }
        temp_matrix[i][j] = find_median(window, window_size *
window_size);
    }
}

sem_post(&end_sem);
}

return NULL;
}

void copy_temp_to_matrix() {
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            matrix[i][j] = temp_matrix[i][j];
        }
    }
}

void generate_matrix() {
    srand(time(NULL));
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            matrix[i][j] = rand() % 100 + 1;
        }
    }
}

void print_matrix(int mat[ROWS][COLS]) {
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            printf("%3d ", mat[i][j]);
        }
        printf("\n");
    }
}

```

Протокол работы программы

```
max@DESKTOP-L04A0IM:/mnt/c/Users/lasto/CLionProjects/Osi/laba2$ gcc -g -pthread laba2.c -o laba2
```

```
max@DESKTOP-L04A0IM:/mnt/c/Users/lasto/CLionProjects/Osi/laba2$ time ./laba2 3 1 1
```

Original matrix:

25	11	40	4	10	85	78	64	88	20
56	23	78	92	68	75	5	31	8	68
46	5	39	2	90	81	65	5	95	75
54	19	37	93	23	46	77	52	61	16
23	69	39	52	60	58	79	16	40	86
84	37	42	74	38	31	55	55	87	49
81	40	19	70	32	93	67	60	96	80
28	18	48	66	70	59	75	48	26	66
85	9	2	26	35	92	8	89	46	95
89	78	86	7	47	18	100	66	29	95

Result matrix:

25	11	40	4	10	85	78	64	88	20
56	39	23	40	75	75	65	64	64	68
46	39	37	68	75	68	52	52	52	75
54	39	39	39	58	65	58	61	52	16
23	39	42	42	52	55	55	55	52	86
84	40	42	42	58	58	58	60	60	49
81	40	42	48	66	59	59	60	60	80
28	28	26	35	66	67	67	60	66	66
85	48	26	47	47	59	66	48	66	95
89	78	86	7	47	18	100	66	29	95

Time taken: 0.000120 seconds

real 0m0.006s

user 0m0.001s

sys 0m0.000s

```
max@DESKTOP-L04A0IM:/mnt/c/Users/lasto/CLionProjects/Osi/laba2$ time ./laba2 3 1 10
```

Original matrix:

68	31	22	21	79	3	14	95	49	91
83	94	80	14	68	98	23	99	15	9

```

64 36 54 54 99 85 85 98 46 12
55 65 42 29 85 73 31 50 19 31
92 53 76 72 18 44 21 40 42 87
49 57 22 2 10 72 86 47 21 83
58 76 47 51 56 31 23 38 32 93
20 24 45 95 47 63 38 19 2 79
5 50 36 79 3 97 2 41 43 23
23 100 50 70 3 5 52 25 94 84

```

Result matrix:

```

68 31 22 21 79 3 14 95 49 91
83 64 36 54 68 79 85 49 49 9
0 0 0 0 0 0 0 0 0 0
55 55 54 54 72 73 50 42 42 31
92 55 53 29 44 44 47 40 42 87
49 57 53 47 44 31 40 38 42 83
58 47 47 47 51 47 38 32 38 93
20 45 50 47 56 38 38 32 38 79
5 36 50 47 63 38 38 38 41 23
23 100 50 70 3 5 52 25 94 84

```

Time taken: 0.000593 seconds

real 0m0.007s

user 0m0.003s

sys 0m0.000s

max@DESKTOP-L04A0IM:/mnt/c/Users/lasto/CLionProjects/Osi/laba2\$ time ./laba2 3 5 10

Original matrix:

```

53 12 59 14 35 72 28 16 53 19
54 56 96 19 73 30 99 68 97 53
70 89 9 33 5 40 60 95 73 76
87 25 88 45 91 74 69 70 89 21
89 43 28 36 13 100 66 11 19 14
16 40 54 76 73 58 68 84 4 40
11 90 16 50 87 6 23 55 28 12
27 16 6 54 51 18 5 68 81 75
34 96 15 87 71 39 97 38 22 100
29 84 42 97 34 28 2 8 82 29

```

Result matrix:

53	12	59	14	35	72	28	16	53	19
0	35	35	35	35	35	28	28	19	0
70	35	35	35	35	35	35	28	21	76
0	43	43	43	45	54	54	28	21	0
89	43	43	45	51	54	54	38	21	14
0	43	43	50	51	54	54	38	21	0
11	29	43	50	51	51	51	38	38	12
27	29	50	51	51	51	38	38	38	75
34	34	51	51	51	34	34	38	55	100
29	84	42	97	34	28	2	8	82	29

Time taken: 0.001624 seconds

real 0m0.007s

user 0m0.004s

sys 0m0.000s

max@DESKTOP-L04A0IM:/mnt/c/Users/lasto/CLionProjects/Osi/laba2\$ time ./laba2 5 5 5

Original matrix:

72	64	94	58	55	65	32	100	38	88
6	3	16	15	83	29	67	60	87	86
49	40	51	38	6	79	13	66	45	38
5	69	1	98	26	56	62	9	7	52
97	64	54	64	78	88	92	96	47	79
81	48	70	31	85	27	62	97	93	6
86	49	74	39	99	51	46	12	60	52
63	8	15	68	71	45	55	63	40	54
93	73	1	62	3	85	88	16	33	32
22	18	81	47	56	79	98	1	90	9

Result matrix:

72	64	94	58	55	65	32	100	38	88
6	3	16	15	83	29	67	60	87	86
49	40	51	54	55	56	56	56	45	38
5	69	49	51	52	54	54	52	7	52
0	0	49	51	52	52	52	52	0	0
0	0	51	51	52	52	52	52	0	0
86	49	51	51	52	52	52	52	60	52

```
63  8 51 51 52 52 52 52 40 54
93 73  1 62  3 85 88 16 33 32
22 18 81 47 56 79 98  1 90  9
```

Time taken: 0.000967 seconds

real 0m0.007s

user 0m0.003s

sys 0m0.000s

```
max@DESKTOP-L04A0IM:/mnt/c/Users/lasto/CLionProjects/Osi/laba2$ time ./laba2 6 5 5
```

Invalid arguments. Window size must be odd and ≥ 1 . Iterations and threads > 0 .

real 0m0.007s

user 0m0.001s

sys 0m0.000s

```
max@DESKTOP-L04A0IM:/mnt/c/Users/lasto/CLionProjects/Osi/laba2$ time ./laba2 3 1 2
```

Original matrix:

```
32 51 13 83 66 34  9 49 25 73
72 65 56 49 76 53  8  9 14  1
14 22  6 11 11 25 85 95 44 56
75 28 59 39 62 24 72 22 24 96
95 47 60 50 96 87 55  3 95 20
55  8 41 60 70  3 36 54 49 32
62 75 59 20 13 20 95 36 94 70
83 88 16 95 89 11 81 43 66 76
62 20 35 55 80  5  9 15 10 58
46 23 32 56 42 97 28 88 32 21
```

Result matrix:

```
32 51 13 83 66 34  9 49 25 73
72 32 49 49 49 34 34 25 44  1
14 56 39 49 39 53 25 24 24 56
75 47 39 50 39 62 55 55 44 96
95 55 47 60 60 62 36 49 32 20
55 59 50 59 50 55 36 54 49 32
62 59 59 59 20 36 36 54 54 70
83 62 55 55 20 20 20 43 58 76
```

62 35 35 55 56 42 28 32 43 58

46 23 32 56 42 97 28 88 32 21

Time taken: 0.000171 seconds

real 0m0.004s

user 0m0.001s

sys 0m0.000s

```
max@DESKTOP-L04A0IM:/mnt/c/Users/lasto/CLionProjects/Osi/laba2$ strace -f time ./laba2 3 1 2
execve("/usr/bin/time", ["time", "./laba2", "3", "1", "2"], 0x7fff101a08a8 /* 27 vars */) = 0
brk(NULL)                                = 0x55edc18e1000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffe64c4ac70) = -1 EINVAL (Invalid argument)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f057f338000
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=18383, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 18383, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f057f333000
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\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\0I\17\357\204\3$\f\221\2039x\324\224\323\236S"... , 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) = 0x7f057f10a000
mprotect(0x7f057f132000, 2023424, PROT_NONE) = 0
mmap(0x7f057f132000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f057f132000
mmap(0x7f057f2c7000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7f057f2c7000
mmap(0x7f057f320000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) = 0x7f057f320000
mmap(0x7f057f326000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f057f326000
close(3)                                  = 0
```

```

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f057f107000
arch_prctl(ARCH_SET_FS, 0x7f057f107740) = 0
set_tid_address(0x7f057f107a10) = 2768
set_robust_list(0x7f057f107a20, 24) = 0
rseq(0x7f057f1080e0, 0x20, 0, 0x53053053) = 0
mprotect(0x7f057f320000, 16384, PROT_READ) = 0
mprotect(0x55eda30bf000, 4096, PROT_READ) = 0
mprotect(0x7f057f372000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f057f333000, 18383) = 0
clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, strace: Process
2769 attached
, child_tidptr=0x7f057f107a10) = 2769
[pid 2769] set_robust_list(0x7f057f107a20, 24 <unfinished ...>
[pid 2768] rt_sigaction(SIGINT, {sa_handler=SIG_IGN, sa_mask=[INT],
sa_flags=SA_RESTORER|SA_RESTART, sa_restorer=0x7f057f14c520}, <unfinished ...>
[pid 2769] <... set_robust_list resumed>) = 0
[pid 2768] <... rt_sigaction resumed>{sa_handler=SIG_DFL, sa_mask=[], sa_flags=0}, 8) = 0
[pid 2768] rt_sigaction(SIGQUIT, {sa_handler=SIG_IGN, sa_mask=[QUIT],
sa_flags=SA_RESTORER|SA_RESTART, sa_restorer=0x7f057f14c520}, <unfinished ...>
[pid 2769] execve("./laba2", [ "./laba2", "3", "1", "2"], 0x7ffe64c4ae68 /* 27 vars */
<unfinished ...>
[pid 2768] <... rt_sigaction resumed>{sa_handler=SIG_DFL, sa_mask=[], sa_flags=0}, 8) = 0
[pid 2768] wait4(-1, <unfinished ...>
[pid 2769] <... execve resumed>) = 0
[pid 2769] brk(NULL) = 0x5648628a3000
[pid 2769] arch_prctl(0x3001 /* ARCH_??? */, 0x7fff10273bd0) = -1 EINVAL (Invalid argument)
[pid 2769] mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f8ab5281000
[pid 2769] access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
[pid 2769] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
[pid 2769] newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=18383, ...}, AT_EMPTY_PATH) = 0
[pid 2769] mmap(NULL, 18383, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f8ab527c000
[pid 2769] close(3) = 0
[pid 2769] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
[pid 2769] 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"...,
832) = 832
[pid 2769] 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
[pid 2769] 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"...,

```

48, 848) = 48

[pid 2769] pread64(3, "
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0I\17\357\204\3\$\f\221\2039x\324\224\323\236S"... , 68, 896) = 68

[pid 2769] newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2220400, ...}, AT_EMPTY_PATH) = 0

[pid 2769] 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

[pid 2769] mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f8ab5053000

[pid 2769] mprotect(0x7f8ab507b000, 2023424, PROT_NONE) = 0

[pid 2769] mmap(0x7f8ab507b000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f8ab507b000

[pid 2769] mmap(0x7f8ab5210000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7f8ab5210000

[pid 2769] mmap(0x7f8ab5269000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) = 0x7f8ab5269000

[pid 2769] mmap(0x7f8ab526f000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f8ab526f000

[pid 2769] close(3) = 0

[pid 2769] mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f8ab5050000

[pid 2769] arch_prctl(ARCH_SET_FS, 0x7f8ab5050740) = 0

[pid 2769] set_tid_address(0x7f8ab5050a10) = 2769

[pid 2769] set_robust_list(0x7f8ab5050a20, 24) = 0

[pid 2769] rseq(0x7f8ab50510e0, 0x20, 0, 0x53053053) = 0

[pid 2769] mprotect(0x7f8ab5269000, 16384, PROT_READ) = 0

[pid 2769] mprotect(0x564837626000, 4096, PROT_READ) = 0

[pid 2769] mprotect(0x7f8ab52bb000, 8192, PROT_READ) = 0

[pid 2769] prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

[pid 2769] munmap(0x7f8ab527c000, 18383) = 0

[pid 2769] newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH) = 0

[pid 2769] getrandom("\xf2\x69\x90\x26\x9d\xe3\x36\x4c", 8, GRND_NONBLOCK) = 8

[pid 2769] brk(NULL) = 0x5648628a3000

[pid 2769] brk(0x5648628c4000) = 0x5648628c4000

[pid 2769] write(1, "Original matrix:\n", 17Original matrix:
) = 17

[pid 2769] write(1, " 54 75 73 73 47 85 24 69 "... , 41 54 75 73 73 47 85 24
69 42 32

) = 41

[pid 2769] write(1, " 31 73 42 68 28 91 47 33 "... , 41 31 73 42 68 28 91 47
33 41 15


```

) = 41

[pid 2769] write(1, " 25 40 21 70 38 82 87 22 "..., 41 25 40 21 70 38 82 87
22 80 89

) = 41

[pid 2769] write(1, " 76 33 16 48 5 62 84 81 "..., 41 76 33 16 48 5 62 84
81 82 26

) = 41

[pid 2769] write(1, " 12 64 50 5 31 29 47 29 "..., 41 12 64 50 5 31 29 47
29 62 39

) = 41

[pid 2769] write(1, " 44 38 78 64 8 67 98 46 "..., 41 44 38 78 64 8 67 98
46 88 29

) = 41

[pid 2769] write(1, " 86 63 14 1 63 70 15 46 "..., 41 86 63 14 1 63 70 15
46 50 96

) = 41

[pid 2769] write(1, " 23 13 60 24 17 42 53 63 "..., 41 23 13 60 24 17 42 53
63 71 66

) = 41

[pid 2769] write(1, " 53 66 3 30 29 62 48 78 "..., 41 53 66 3 30 29 62 48
78 59 36

) = 41

[pid 2769] write(1, " 7 45 50 72 97 12 41 11 "..., 41 7 45 50 72 97 12 41
11 10 91

) = 41

[pid 2769] rt_sigaction(SIGRT_1, {sa_handler=0x7f8ab50e4870, sa_mask=[],
sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO, sa_restorer=0x7f8ab5095520}, NULL, 8)
= 0

[pid 2769] rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0

[pid 2769] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) =
0x7f8ab484f000

[pid 2769] mprotect(0x7f8ab4850000, 8388608, PROT_READ|PROT_WRITE) = 0

[pid 2769] rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0

[pid 2769]
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_S
ETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f8ab504f910,
parent_tid=0x7f8ab504f910, exit_signal=0, stack=0x
7f8ab484f000, stack_size=0x7fff00, tls=0x7f8ab504f640}strace: Process 2770 attached

<unfinished ...>

[pid 2770] rseq(0x7f8ab504ffe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 2769] <... clone3 resumed> => {parent_tid=[2770]}, 88) = 2770

[pid 2770] <... rseq resumed> = 0

[pid 2769] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

```

```

[pid 2770] set_robust_list(0x7f8ab504f920, 24 <unfinished ...>
[pid 2769] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 2770] <... set_robust_list resumed>) = 0
[pid 2769] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0
<unfinished ...>
[pid 2770] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 2769] <... mmap resumed>) = 0x7f8ab404e000
[pid 2770] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 2769] mprotect(0x7f8ab404f000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>
[pid 2770] futex(0x564837627380, FUTEX_WAIT_BITSET_PRIVATE|FUTEX_CLOCK_REALTIME, 0, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
[pid 2769] <... mprotect resumed>) = 0
[pid 2769] rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
[pid 2769]
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_S
ETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f8ab484e910,
parent_tid=0x7f8ab484e910, exit_signal=0, stack=0x
7f8ab404e000, stack_size=0x7fff00, tls=0x7f8ab484e640}strace: Process 2771 attached
=> {parent_tid=[2771]}, 88) = 2771
[pid 2771] rseq(0x7f8ab484efe0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 2769] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 2771] <... rseq resumed>) = 0
[pid 2769] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 2771] set_robust_list(0x7f8ab484e920, 24 <unfinished ...>
[pid 2769] clock_gettime(CLOCK_PROCESS_CPUTIME_ID, <unfinished ...>
[pid 2771] <... set_robust_list resumed>) = 0
[pid 2769] <... clock_gettime resumed>{tv_sec=0, tv_nsec=2541100}) = 0
[pid 2771] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 2769] futex(0x564837627380, FUTEX_WAKE_PRIVATE, 1 <unfinished ...>
[pid 2771] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 2771] futex(0x564837627380, FUTEX_WAIT_BITSET_PRIVATE|FUTEX_CLOCK_REALTIME, 0, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
[pid 2770] <... futex resumed>) = 0
[pid 2769] <... futex resumed>) = 1
[pid 2770] futex(0x564837627380, FUTEX_WAIT_BITSET_PRIVATE|FUTEX_CLOCK_REALTIME, 0, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
[pid 2769] futex(0x564837627380, FUTEX_WAKE_PRIVATE, 1 <unfinished ...>
[pid 2771] <... futex resumed>) = 0
[pid 2770] <... futex resumed>) = -1 EAGAIN (Resource temporarily unavailable)
[pid 2769] <... futex resumed>) = 1

```

```

[pid 2770] futex(0x564837627380, FUTEX_WAIT_BITSET_PRIVATE|FUTEX_CLOCK_REALTIME, 0, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>

[pid 2771] futex(0x564837627380, FUTEX_WAIT_BITSET_PRIVATE|FUTEX_CLOCK_REALTIME, 0, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>

[pid 2769] futex(0x564837627380, FUTEX_WAKE_PRIVATE, 1 <unfinished ...>

[pid 2770] <... futex resumed>)          = 0
[pid 2769] <... futex resumed>)          = 1

[pid 2770] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>
[pid 2769] futex(0x564837627380, FUTEX_WAKE_PRIVATE, 1 <unfinished ...>

[pid 2770] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 2771] <... futex resumed>)          = 0
[pid 2769] <... futex resumed>)          = 1

[pid 2770] madvise(0x7f8ab484f000, 8368128, MADV_DONTNEED <unfinished ...>
[pid 2771] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>

[pid 2770] <... madvise resumed>)          = 0

[pid 2769] futex(0x7f8ab504f910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 2770, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>

[pid 2771] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 2770] exit(0 <unfinished ...>

[pid 2771] madvise(0x7f8ab404e000, 8368128, MADV_DONTNEED) = 0
[pid 2770] <... exit resumed>)            = ?
[pid 2771] exit(0 <unfinished ...>

[pid 2769] <... futex resumed>)          = 0
[pid 2770] +++ exited with 0 +++
[pid 2771] <... exit resumed>)            = ?

[pid 2769] futex(0x7f8ab484e910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 2771, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>

[pid 2771] +++ exited with 0 +++
[pid 2769] <... futex resumed>)          = -1 EAGAIN (Resource temporarily unavailable)
[pid 2769] clock_gettime(CLOCK_PROCESS_CPUTIME_ID, {tv_sec=0, tv_nsec=3099700}) = 0
[pid 2769] write(1, "Result matrix:\n", 15Result matrix:

) = 15
[pid 2769] write(1, " 0 0 0 0 0 0 0 0 "..., 41 0 0 0 0 0 0 0
0 0 0

) = 41
[pid 2769] write(1, " 0 0 0 0 0 0 0 0 "..., 41 0 0 0 0 0 0 0
0 0 0

) = 41
[pid 2769] write(1, " 0 0 0 0 0 0 0 0 "..., 41 0 0 0 0 0 0 0
0 0 0

```

```

) = 41

[pid 2769] write(1, " 0 0 0 0 0 0 0 0 "..., 41 0 0 0 0 0 0 0
0 0 0

) = 41

[pid 2769] write(1, " 0 0 0 0 0 0 0 0 "..., 41 0 0 0 0 0 0 0
0 0 0

) = 41

[pid 2769] write(1, " 44 50 50 31 31 47 46 47 "..., 41 44 50 50 31 31 47 46
47 46 29

) = 41

[pid 2769] write(1, " 86 44 38 24 42 53 53 53 "..., 41 86 44 38 24 42 53 53
53 63 96

) = 41

[pid 2769] write(1, " 23 53 24 24 30 48 53 53 "..., 41 23 53 24 24 30 48 53
53 63 66

) = 41

[pid 2769] write(1, " 53 45 45 30 30 42 48 53 "..., 41 53 45 45 30 30 42 48
53 63 36

) = 41

[pid 2769] write(1, " 7 45 50 72 97 12 41 11 "..., 41 7 45 50 72 97 12 41
11 10 91

) = 41

[pid 2769] write(1, "Time taken: 0.000558 seconds\n", 29Time taken: 0.000558 seconds

) = 29

[pid 2769] exit_group(0) = ?

[pid 2769] +++ exited with 0 +++

<... wait4 resumed>[{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, {ru_utime={tv_sec=0,
tv_usec=3572}, ru_stime={tv_sec=0, tv_usec=0}, ...}) = 2769

--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=2769, si_uid=1000, si_status=0,
si_utime=0, si_stime=0} ---

rt_sigaction(SIGINT, {sa_handler=SIG_DFL, sa_mask=[INT], sa_flags=SA_RESTORER|SA_RESTART,
sa_restorer=0x7f057f14c520}, {sa_handler=SIG_IGN, sa_mask=[INT],
sa_flags=SA_RESTORER|SA_RESTART, sa_restorer=0x7f057f14c520}, 8) = 0

rt_sigaction(SIGQUIT, {sa_handler=SIG_DFL, sa_mask=[QUIT], sa_flags=SA_RESTORER|SA_RESTART,
sa_restorer=0x7f057f14c520}, {sa_handler=SIG_IGN, sa_mask=[QUIT],
sa_flags=SA_RESTORER|SA_RESTART, sa_restorer=0x7f057f14c520}, 8) = 0

write(2, "0.00", 40.00) = 4

write(2, "u", 1u) = 1

write(2, "s", 1s) = 1

write(2, "e", 1e) = 1

write(2, "r", 1r) = 1

write(2, " ", 1 ) = 1

write(2, "0.00", 40.00) = 4

```

write(2, "s", 1s)	= 1
write(2, "y", 1y)	= 1
write(2, "s", 1s)	= 1
write(2, "t", 1t)	= 1
write(2, "e", 1e)	= 1
write(2, "m", 1m)	= 1
write(2, " ", 1)	= 1
write(2, "0:00.01", 70:00.01)	= 7
write(2, "e", 1e)	= 1
write(2, "l", 1l)	= 1
write(2, "a", 1a)	= 1
write(2, "p", 1p)	= 1
write(2, "s", 1s)	= 1
write(2, "e", 1e)	= 1
write(2, "d", 1d)	= 1
write(2, " ", 1)	= 1
write(2, "20%", 320%)	= 3
write(2, "C", 1C)	= 1
write(2, "P", 1P)	= 1
write(2, "U", 1U)	= 1
write(2, " ", 1)	= 1
write(2, "(", 1()	= 1
write(2, "0", 10)	= 1
write(2, "a", 1a)	= 1
write(2, "v", 1v)	= 1
write(2, "g", 1g)	= 1
write(2, "t", 1t)	= 1
write(2, "e", 1e)	= 1
write(2, "x", 1x)	= 1
write(2, "t", 1t)	= 1
write(2, "+", 1+)	= 1
write(2, "0", 10)	= 1
write(2, "a", 1a)	= 1
write(2, "v", 1v)	= 1
write(2, "g", 1g)	= 1
write(2, "d", 1d)	= 1
write(2, "a", 1a)	= 1

write(2, "t", 1t)	= 1
write(2, "a", 1a)	= 1
write(2, " ", 1)	= 1
write(2, "1908", 41908)	= 4
write(2, "m", 1m)	= 1
write(2, "a", 1a)	= 1
write(2, "x", 1x)	= 1
write(2, "r", 1r)	= 1
write(2, "e", 1e)	= 1
write(2, "s", 1s)	= 1
write(2, "i", 1i)	= 1
write(2, "d", 1d)	= 1
write(2, "e", 1e)	= 1
write(2, "n", 1n)	= 1
write(2, "t", 1t)	= 1
write(2, ")", 1))	= 1
write(2, "k", 1k)	= 1
write(2, "\n", 1	
)	= 1
write(2, "48", 248)	= 2
write(2, "i", 1i)	= 1
write(2, "n", 1n)	= 1
write(2, "p", 1p)	= 1
write(2, "u", 1u)	= 1
write(2, "t", 1t)	= 1
write(2, "s", 1s)	= 1
write(2, "+", 1+)	= 1
write(2, "0", 10)	= 1
write(2, "o", 1o)	= 1
write(2, "u", 1u)	= 1
write(2, "t", 1t)	= 1
write(2, "p", 1p)	= 1
write(2, "u", 1u)	= 1
write(2, "t", 1t)	= 1
write(2, "s", 1s)	= 1
write(2, " ", 1)	= 1
write(2, "(", 1(= 1

```

write(2, "1", 11)           = 1
write(2, "m", 1m)           = 1
write(2, "a", 1a)           = 1
write(2, "j", 1j)           = 1
write(2, "o", 1o)           = 1
write(2, "r", 1r)           = 1
write(2, "+", 1+)           = 1
write(2, "78", 278)         = 2
write(2, "m", 1m)           = 1
write(2, "i", 1i)           = 1
write(2, "n", 1n)           = 1
write(2, "o", 1o)           = 1
write(2, "r", 1r)           = 1
write(2, ")", 1))           = 1
write(2, "p", 1p)           = 1
write(2, "a", 1a)           = 1
write(2, "g", 1g)           = 1
write(2, "e", 1e)           = 1
write(2, "f", 1f)           = 1
write(2, "a", 1a)           = 1
write(2, "u", 1u)           = 1
write(2, "l", 1l)           = 1
write(2, "t", 1t)           = 1
write(2, "s", 1s)           = 1
write(2, " ", 1 )           = 1
write(2, "0", 10)           = 1
write(2, "s", 1s)           = 1
write(2, "w", 1w)           = 1
write(2, "a", 1a)           = 1
write(2, "p", 1p)           = 1
write(2, "s", 1s)           = 1
write(2, "\n", 1
)                             = 1

```

```

exit_group(0)                = ?

```

```

+++ exited with 0 +++

```

```

max@DESKTOP-L04A0IM:/mnt/c/Users/lasto/CLionProjects/Osi/lab2$

```

Вывод

В ходе написания данной лабораторной работы я научился создавать программы, работающие с несколькими потоками, а также синхронизировать их между собой. В результате тестирования программы, я проанализировал каким образом количество потоков влияет на эффективность и ускорение работы программы. Оказалось, что большое количество потоков даёт хорошее ускорение на больших количествах входных данных, но эффективность использования ресурсов находится на приемлемом уровне только на небольшом количестве потоков, не превышающем количества логических ядер процессора. Лабораторная работа была довольно интересна, так как я впервые работал с многопоточностью и синхронизацией на СИ.