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: Акимов К.К.

Преподаватель: Бахарев В.Д.

Оценка:

_____

Дата: 09.12.24

Москва, 2024

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

Вариант 12. Наложить K раз фильтры эрозии и наращивания на матрицу, состоящую из вещественных чисел. На выходе получается 2 результирующие матрицы

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); Дожидается завершения потока

Программа получает на вход два аргумента – количество итераций и максимальное количество потоков. Window_size равняется 3 (окно всегда должно

быть нечетного размера для того, чтобы было однозначно выделить центральный элемент). Iterations отвечает за то, сколько раз будут выполнены фильтры эрозии и наращивания 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 5x5.

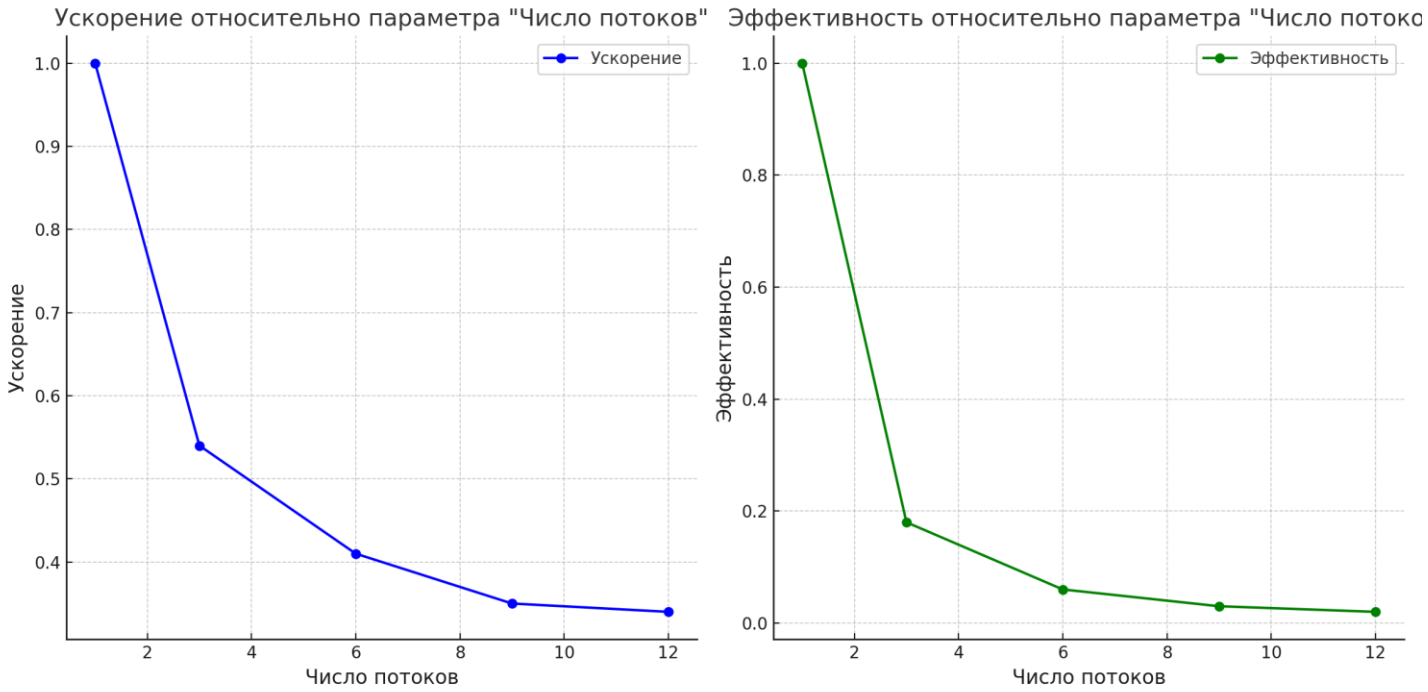
После создается нужное количество потоков для обработки каждого фильтра эрозии и наращивания у матрицы. Изначально созданная матрица копируется в temp_matrix_er и temp_matrix_nar (это две матрицы для наложения на них фильтров эрозии и наращивания соответственно).

Суть метода эрозии (наращивания):

1. Формируется окно вокруг центрального элемента.
2. Из окна берутся все элементы.
3. Находится минимальное (максимальное) из отсортированного списка чисел окна.
4. Центральное значение элемента заменяется на найденное минимальное (максимальное).

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

Число потоков	Время выполнения	Ускорение	Эффективность
1	105	1,00	1,00
3	177	0,54	0,18
6	254	0,41	0,06
9	296	0,35	0,03
12	309	0,34	0,02



Код программы

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



#define ROWS 5
#define COLS 5

double matrix[ROWS][COLS]; // матрица
double temp_matrix_er[ROWS][COLS];
double temp_matrix_nar[ROWS][COLS];


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


int compare(const void* a, const void* b);
double find_min(double* window, int size);
double find_max(double* window, int size);
void* filters(void* args);
void copy_temp_to_matrix();
void generate_matrix();
void print_matrix(double mat[ROWS][COLS]);

sem_t start_sem;
sem_t end_sem;

int stop_threads = 0;

int main(int argc, char* argv[]) {
    if (argc != 3) {
        fprintf(stderr, "Usage: ./a.out <iterations> <max_threads>\n");
        return EXIT_FAILURE;
    }

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


   if (iterations < 1 || max_threads < 1) {
        fprintf(stderr, "Invalid input. Iterations and threads > 0.\n");
        return EXIT_FAILURE;
    }
```

```
 // Ограничиваем количество потоков размером матрицы (не более ROWS чтобы пустых не было)
 if (max_threads > ROWS) {
     max_threads = ROWS;
 }


generate_matrix();
 copy_temp_to_matrix();


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

 // Создание массива для хранения id потоков и их аргументов
 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, filters, &thread_args[i]) != 0) {
         perror("pthread_create failed");
         return EXIT_FAILURE;
    }
 }


clock_t start_time = clock();

 // Основной цикл
 for (int iter = 0; iter < iterations; iter++) {

    for (int i = 0; i < max_threads; i++) {
         sem_post(&start_sem); // увеличиваем значение семафора
     }




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

 }

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

```c
   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("\n");
    printf("Result erosion matrix:\n");
    print_matrix(temp_matrix_er);

    printf("\n");

    printf("Result building up matrix:\n");
    print_matrix(temp_matrix_nar);
    printf("\n");

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

    return EXIT_SUCCESS;
}

int compare(const void* a, const void* b) {
    return (*(double*)a - *(double*)b - 0.00000001 > 0.0);  // Сортируем по возрастанию
}


double find_min(double* window, int size) {
    qsort(window, size, sizeof(double), compare);
   return window[0];  // Возвращаем min
}
double find_max(double* window, int size) {
    qsort(window, size, sizeof(double), compare);
    return window[size - 1];  // Возвращаем max
}


void* filters(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;  // Половина размера окна

    while (1) {
        sem_wait(&start_sem); // ( семафор - 1 )

        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) {
```

```
// краевые эл-ты
                    temp_matrix_er[i][j] = temp_matrix_er[i][j];
                    temp_matrix_nar[i][j] = temp_matrix_nar[i][j];
                } else {
                    // Для остальных элементов вычисляем
                    double 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_er[i][j] = find_min(window, window_size * window_size);
                   temp_matrix_nar[i][j] = find_max(window, window_size * window_size);
                }
            }
        }

        sem_post(&end_sem); // (семафор +1)
    }

    return NULL;
}

void copy_temp_to_matrix() {
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            temp_matrix_er[i][j] = matrix[i][j];
           temp_matrix_nar[i][j] = 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] = 1.0 + (rand() / (double)RAND_MAX) * 99.0;
       }
    }
}


void print_matrix(double mat[ROWS][COLS]) {
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            printf("%.1f ", mat[i][j]);
       }
        printf("\n");
   }
}
```

Протокол работы программы

kirill@DESKTOP-O0B2VHP:/mnt/c/Users/User/OSI/lab2/src$ gcc -o filter_program main2.c -pthread -Wall
kirill@DESKTOP-O0B2VHP:/mnt/c/Users/User/OSI/lab2/src$ ./filter_program 1 3
Original matrix:
23.4 3.2 75.4 41.3 38.5
31.3 8.8 41.2 24.7 53.0
6.9 54.5 69.1 47.7 94.5
6.9 62.1 56.4 98.8 26.9
69.0 8.6 39.0 2.5 13.4

Result erosion matrix:
23.4 3.2 75.4 41.3 38.5
31.3 8.8 41.2 24.7 53.0
6.9 6.9 8.8 24.7 94.5
6.9 6.9 2.5 2.5 26.9
69.0 8.6 39.0 2.5 13.4

Result building up matrix:
23.4 3.2 75.4 41.3 38.5
31.3 8.8 41.2 24.7 53.0
6.9 69.1 98.8 98.8 94.5
6.9 69.1 98.8 98.8 26.9
69.0 8.6 39.0 2.5 13.4

Time: 0.000140 seconds

kirill@DESKTOP-O0B2VHP:/mnt/c/Users/User/OSI/lab2/src$ ./filter_program 3 6
Original matrix:
3.0 27.7 47.6 19.5 60.3
87.2 19.6 23.3 30.3 53.2
68.8 16.4 6.5 10.1 47.2
81.6 7.1 76.2 32.1 38.4
25.2 51.6 34.8 83.1 26.8

Result erosion matrix:
3.0 27.7 47.6 19.5 60.3
87.2 3.0 6.5 6.5 53.2
68.8 6.5 6.5 6.5 47.2
81.6 6.5 6.5 6.5 38.4
25.2 51.6 34.8 83.1 26.8

Result building up matrix:
3.0 27.7 47.6 19.5 60.3
87.2 87.2 47.6 60.3 53.2
68.8 87.2 76.2 76.2 47.2
81.6 81.6 83.1 83.1 38.4
25.2 51.6 34.8 83.1 26.8

Time: 0.000430 seconds

kirill@DESKTOP-O0B2VHP:/mnt/c/Users/User/OSI/lab2/src$ ./filter_program 5 5
Original matrix:
7.7 61.9 68.0 13.4 82.5
14.8 35.1 12.0 12.8 8.5
8.5 94.7 40.7 26.6 11.3
11.6 68.3 35.4 42.3 66.8
77.4 28.3 35.4 90.3 39.5

Result erosion matrix:

7.7 61.9 68.0 13.4 82.5
14.8 7.7 12.0 8.5 8.5
8.5 8.5 12.0 8.5 11.3
11.6 8.5 26.6 11.3 66.8
77.4 28.3 35.4 90.3 39.5

Result building up matrix:
7.7 61.9 68.0 13.4 82.5
14.8 94.7 94.7 82.5 8.5
8.5 94.7 94.7 66.8 11.3
11.6 94.7 94.7 90.3 66.8
77.4 28.3 35.4 90.3 39.5

Time: 0.000672 seconds
kirill@DESKTOP-O0B2VHP:/mnt/c/Users/User/OSI/lab2/src$

kirill@DESKTOP-O0B2VHP:/mnt/c/Users/User/OSI/lab2/src$ strace -f time ./filter_program 1 3

execve("/usr/bin/time", ["time", "./filter_program", "1", "3"], 0x7ffc75ba2e00 /* 28 vars */) = 0

brk(NULL) = 0x55cf48a48000

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f1b26fc4000

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=20231, ...}) = 0

mmap(NULL, 20231, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f1b26fbf000

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\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) = 0x7f1b26dad000

mmap(0x7f1b26dd5000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f1b26dd5000

mmap(0x7f1b26f5d000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000) = 0x7f1b26f5d000

mmap(0x7f1b26fac000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x7f1b26fac000

mmap(0x7f1b26fb2000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f1b26fb2000

close(3) = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f1b26daa000

arch_prctl(ARCH_SET_FS, 0x7f1b26daa740) = 0

set_tid_address(0x7f1b26daaa10) = 3401

set_robust_list(0x7f1b26daaa20, 24) = 0

rseq(0x7f1b26dab060, 0x20, 0, 0x53053053) = 0

mprotect(0x7f1b26fac000, 16384, PROT_READ) = 0

mprotect(0x55cf46eef000, 4096, PROT_READ) = 0

mprotect(0x7f1b26ffc000, 8192, PROT_READ) = 0

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

munmap(0x7f1b26fbf000, 20231) = 0

clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLDstrace: Process 3402 attached

, child_tidptr=0x7f1b26daaa10) = 3402

[pid 3401] rt_sigaction(SIGINT, {sa_handler=SIG_IGN, sa_mask=[INT], sa_flags=SA_RESTORER|SA_RESTART, sa_restorer=0x7f1b26df2320}, <unfinished ...>

[pid 3402] set_robust_list(0x7f1b26daaa20, 24 <unfinished ...>

[pid 3401] <... rt_sigaction resumed>{sa_handler=SIG_DFL, sa_mask=[], sa_flags=0}, 8) = 0

[pid 3402] <... set_robust_list resumed>) = 0

[pid 3401] rt_sigaction(SIGQUIT, {sa_handler=SIG_IGN, sa_mask=[QUIT], sa_flags=SA_RESTORER|SA_RESTART, sa_restorer=0x7f1b26df2320}, {sa_handler=SIG_DFL, sa_mask=[], sa_flags=0}, 8) = 0

[pid 3402] execve("./filter_program", ["./filter_program", "1", "3"], 0x7ffd7d944780 /* 28 vars */ <unfinished ...>

[pid 3401] wait4(-1, <unfinished ...>

[pid 3402] <... execve resumed>) = 0

[pid 3402] brk(NULL) = 0x557b5f1fe000

[pid 3402] mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f5a5cdff000

[pid 3402] access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)

[pid 3402] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

[pid 3402] fstat(3, {st_mode=S_IFREG|0644, st_size=20231, ...}) = 0

[pid 3402] mmap(NULL, 20231, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f5a5cdfa000

[pid 3402] close(3) = 0

[pid 3402] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3

[pid 3402] 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

[pid 3402] 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 3402] fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0

[pid 3402] 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 3402] mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f5a5cbe8000

[pid 3402] mmap(0x7f5a5cc10000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f5a5cc10000

[pid 3402] mmap(0x7f5a5cd98000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000) = 0x7f5a5cd98000

[pid 3402] mmap(0x7f5a5cde7000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x7f5a5cde7000

[pid 3402] mmap(0x7f5a5cded000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f5a5cded000

[pid 3402] close(3) = 0

[pid 3402] mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f5a5cbe5000

[pid 3402] arch_prctl(ARCH_SET_FS, 0x7f5a5cbe5740) = 0

[pid 3402] set_tid_address(0x7f5a5cbe5a10) = 3402

[pid 3402] set_robust_list(0x7f5a5cbe5a20, 24) = 0

[pid 3402] rseq(0x7f5a5cbe6060, 0x20, 0, 0x53053053) = 0

[pid 3402] mprotect(0x7f5a5cde7000, 16384, PROT_READ) = 0

[pid 3402] mprotect(0x557b5e81d000, 4096, PROT_READ) = 0

[pid 3402] mprotect(0x7f5a5ce37000, 8192, PROT_READ) = 0

[pid 3402] prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

[pid 3402] munmap(0x7f5a5cdfa000, 20231) = 0

[pid 3402] fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0

[pid 3402] getrandom("\x9a\x53\x6f\xbc\x2c\x4a\x48\x9b", 8, GRND_NONBLOCK) = 8

[pid 3402] brk(NULL) = 0x557b5f1fe000

[pid 3402] brk(0x557b5f21f000) = 0x557b5f21f000

[pid 3402] write(1, "Original matrix:\n", 17Original matrix:

) = 17

[pid 3402] write(1, "49.1 17.4 36.1 60.6 53.9 \n", 2649.1 17.4 36.1 60.6 53.9

) = 26

[pid 3402] write(1, "89.1 54.0 1.5 10.1 80.7 \n", 2589.1 54.0 1.5 10.1 80.7

) = 25

[pid 3402] write(1, "56.4 63.5 25.3 11.0 25.9 \n", 2656.4 63.5 25.3 11.0 25.9

) = 26

[pid 3402] write(1, "46.8 90.0 58.6 34.5 35.9 \n", 2646.8 90.0 58.6 34.5 35.9

) = 26

[pid 3402] write(1, "74.5 64.2 26.8 15.7 57.7 \n", 2674.5 64.2 26.8 15.7 57.7

) = 26

[pid 3402] rt_sigaction(SIGRT_1, {sa_handler=0x7f5a5cc81520, sa_mask=[], sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO, sa_restorer=0x7f5a5cc2d320}, NULL, 8) = 0

[pid 3402] rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0

[pid 3402] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7f5a5c3e4000

[pid 3402] mprotect(0x7f5a5c3e5000, 8388608, PROT_READ|PROT_WRITE) = 0

[pid 3402] rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0

[pid 3402] clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEART

ID, child_tid=0x7f5a5cbe4990, parent_tid=0x7f5a5cbe4990, exit_signal=0, stack=0x7f5a5c3e4000, stack_size=0x7fff80, tls=0x7f5a5cbe46c0}strace: Process 3403 attached

<unfinished ...>

[pid 3403] rseq(0x7f5a5cbe4fe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 3402] <... clone3 resumed> => {parent_tid=[3403]}, 88) = 3403

[pid 3403] <... rseq resumed>) = 0

[pid 3402] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 3403] set_robust_list(0x7f5a5cbe49a0, 24 <unfinished ...>

[pid 3402] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 3403] <... set_robust_list resumed>) = 0

[pid 3402] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0 <unfinished ...>

[pid 3403] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 3402] <... mmap resumed>) = 0x7f5a5bbe3000

[pid 3403] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 3402] mprotect(0x7f5a5bbe4000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

[pid 3403] futex(0x557b5e81e2e0, FUTEX_WAIT_BITSET_PRIVATE|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>

[pid 3402] <... mprotect resumed>) = 0

[pid 3402] rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0

[pid 3402] clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEART

ID, child_tid=0x7f5a5c3e3990, parent_tid=0x7f5a5c3e3990, exit_signal=0, stack=0x7f5a5bbe3000, stack_size=0x7fff80, tls=0x7f5a5c3e36c0}strace: Process 3404 attached

<unfinished ...>

[pid 3404] rseq(0x7f5a5c3e3fe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 3402] <... clone3 resumed> => {parent_tid=[3404]}, 88) = 3404

[pid 3404] <... rseq resumed>) = 0

[pid 3402] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 3404] set_robust_list(0x7f5a5c3e39a0, 24 <unfinished ...>

[pid 3402] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 3404] <... set_robust_list resumed>) = 0

[pid 3402] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0 <unfinished ...>

[pid 3404] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 3402] <... mmap resumed>) = 0x7f5a5b3e2000

[pid 3404] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 3402] mprotect(0x7f5a5b3e3000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

[pid 3404] futex(0x557b5e81e2e0, FUTEX_WAIT_BITSET_PRIVATE|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>

[pid 3402] <... mprotect resumed>) = 0

[pid 3402] rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0

[pid 3402] *clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEART*

ID, child_tid=0x7f5a5bbe2990, parent_tid=0x7f5a5bbe2990, exit_signal=0, stack=0x7f5a5b3e2000, stack_size=0x7fff80, tls=0x7f5a5bbe26c0}strace: Process 3405 attached

<unfinished ...>

[pid 3405] rseq(0x7f5a5bbe2fe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 3402] <... clone3 resumed> => {parent_tid=[3405]}, 88) = 3405

[pid 3405] <... rseq resumed>) = 0

[pid 3402] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 3405] set_robust_list(0x7f5a5bbe29a0, 24 <unfinished ...>

[pid 3402] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 3405] <... set_robust_list resumed>) = 0

[pid 3402] clock_gettime(CLOCK_PROCESS_CPUTIME_ID, <unfinished ...>

[pid 3405] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 3402] <... clock_gettime resumed>{tv_sec=0, tv_nsec=6624800}) = 0

[pid 3405] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 3402] futex(0x557b5e81e2e0, FUTEX_WAKE_PRIVATE, 1 <unfinished ...>

[pid 3405] futex(0x557b5e81e2e0, FUTEX_WAIT_BITSET_PRIVATE|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>

[pid 3403] <... futex resumed>) = 0

[pid 3402] <... futex resumed>) = 1

[pid 3403] futex(0x557b5e81e2e0, FUTEX_WAIT_BITSET_PRIVATE|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY) = -1 EAGAIN (Resource temporarily unavailable)

[pid 3402] futex(0x557b5e81e2e0, FUTEX_WAKE_PRIVATE, 1 <unfinished ...>

[pid 3403] futex(0x557b5e81e2e0, FUTEX_WAIT_BITSET_PRIVATE|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>

[pid 3402] <... futex resumed>) = 1

[pid 3404] <... futex resumed>) = 0

[pid 3402] futex(0x557b5e81e2e0, FUTEX_WAKE_PRIVATE, 1 <unfinished ...>

[pid 3404] futex(0x557b5e81e2e0, FUTEX_WAIT_BITSET_PRIVATE|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>

[pid 3405] <... futex resumed>) = 0

[pid 3402] <... futex resumed>) = 1

[pid 3405] futex(0x557b5e81e2e0, FUTEX_WAIT_BITSET_PRIVATE|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>

[pid 3402] futex(0x557b5e81e2e0, FUTEX_WAKE_PRIVATE, 1 <unfinished ...>

[pid 3405] <... futex resumed>) = -1 EAGAIN (Resource temporarily unavailable)

[pid 3403] <... futex resumed>) = 0

[pid 3402] <... futex resumed>) = 1

[pid 3405] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>

[pid 3403] futex(0x557b5e81e2e0, FUTEX_WAIT_BITSET_PRIVATE|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>

[pid 3402] futex(0x557b5e81e2e0, FUTEX_WAKE_PRIVATE, 1 <unfinished ...>

[pid 3403] <... futex resumed>) = -1 EAGAIN (Resource temporarily unavailable)

[pid 3405] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 3403] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>

[pid 3402] <... futex resumed>) = 1

[pid 3405] madvise(0x7f5a5b3e2000, 8368128, MADV_DONTNEED <unfinished ...>

[pid 3404] <... futex resumed>) = 0

[pid 3403] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 3405] <... madvise resumed>) = 0

[pid 3404] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>

[pid 3402] futex(0x557b5e81e2e0, FUTEX_WAKE_PRIVATE, 1 <unfinished ...>

[pid 3405] exit(0 <unfinished ...>

[pid 3404] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 3403] madvise(0x7f5a5c3e4000, 8368128, MADV_DONTNEED <unfinished ...>

[pid 3405] <... exit resumed>) = ?

[pid 3404] madvise(0x7f5a5bbe3000, 8368128, MADV_DONTNEED <unfinished ...>

[pid 3402] <... futex resumed>) = 0

[pid 3405] +++ exited with 0 +++

[pid 3404] <... madvise resumed>) = 0

[pid 3403] <... madvise resumed>) = 0

[pid 3402] futex(0x7f5a5cbe4990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 3403, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>

[pid 3404] exit(0 <unfinished ...>

[pid 3403] exit(0 <unfinished ...>

[pid 3404] <... exit resumed>) = ?

[pid 3403] <... exit resumed>) = ?

[pid 3404] +++ exited with 0 +++

[pid 3403] +++ exited with 0 +++

[pid 3402] <... futex resumed>) = 0

[pid 3402] clock_gettime(CLOCK_PROCESS_CPUTIME_ID, {tv_sec=0, tv_nsec=7347200}) = 0

[pid 3402] write(1, "\n", 1

) = 1

[pid 3402] write(1, "Result erosion matrix:\n", 23Result erosion matrix:

) = 23

[pid 3402] write(1, "49.1 17.4 36.1 60.6 53.9 \n", 2649.1 17.4 36.1 60.6 53.9

) = 26

[pid 3402] write(1, "89.1 1.5 1.5 1.5 80.7 \n", 2389.1 1.5 1.5 1.5 80.7

) = 23

[pid 3402] write(1, "56.4 1.5 1.5 1.5 25.9 \n", 2356.4 1.5 1.5 1.5 25.9

) = 23

[pid 3402] write(1, "46.8 25.3 11.0 11.0 35.9 \n", 2646.8 25.3 11.0 11.0 35.9

) = 26

[pid 3402] write(1, "74.5 64.2 26.8 15.7 57.7 \n", 2674.5 64.2 26.8 15.7 57.7

) = 26

[pid 3402] write(1, "\n", 1

) = 1

[pid 3402] write(1, "Result building up matrix:\n", 27Result building up matrix:

) = 27

[pid 3402] write(1, "49.1 17.4 36.1 60.6 53.9 \n", 2649.1 17.4 36.1 60.6 53.9

) = 26

[pid 3402] write(1, "89.1 89.1 63.5 80.7 80.7 \n", 2689.1 89.1 63.5 80.7 80.7

) = 26

[pid 3402] write(1, "56.4 90.0 90.0 80.7 25.9 \n", 2656.4 90.0 90.0 80.7 25.9

) = 26

[pid 3402] write(1, "46.8 90.0 90.0 58.6 35.9 \n", 2646.8 90.0 90.0 58.6 35.9

) = 26

[pid 3402] write(1, "74.5 64.2 26.8 15.7 57.7 \n", 2674.5 64.2 26.8 15.7 57.7

) = 26

[pid 3402] write(1, "\n", 1

) = 1

[pid 3402] write(1, "Time: 0.000723 seconds\n", 23Time: 0.000723 seconds

) = 23

[pid 3402] exit_group(0) = ?

[pid 3402] +++ exited with 0 +++

<... wait4 resumed>[{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, {ru_utime={tv_sec=0, tv_usec=0}, ru_stime={tv_sec=0, tv_usec=9082}, ...}) = 3402

--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=3402, si_uid=1000, si_status=0, si_utime=0, si_stime=2 /* 0.02 s */} ---

rt_sigaction(SIGINT, {sa_handler=SIG_DFL, sa_mask=[INT], sa_flags=SA_RESTORER|SA_RESTART,

sa_restorer=0x7f1b26df2320}, {sa_handler=SIG_IGN, sa_mask=[INT], sa_flags=SA_RESTORER|SA_RESTART, sa_restorer=0x7f1b26df2320}, 8) = 0

rt_sigaction(SIGQUIT, {sa_handler=SIG_DFL, sa_mask=[QUIT], sa_flags=SA_RESTORER|SA_RESTART, sa_restorer=0x7f1b26df2320}, {sa_handler=SIG_IGN, sa_mask=[QUIT], sa_flags=SA_RESTORER|SA_RESTART, sa_restorer=0x7f1b26df2320}, 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, "0:00.03", 70:00.03) = 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, "28%", 328%) = 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, "1900", 41900) = 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, "40", 240) = 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, "82", 282) = 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 +++

kirill@DESKTOP-O0B2VHP:/mnt/c/Users/User/OSI/lab2/src$

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.