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-213Б-23

Студент: Заитова Е.А

Преподаватель: Бахарев В.Д. (ФИИТ)

Оценка: _____

Дата: 16.11.24

Москва, 2024

Постановка задачи

Вариант 11.

Наложить k раз медианный фильтр на матрицу, состоящую из целых чисел. Размер окна вводится пользователем.

Общий метод и алгоритм решения

Использованные системные вызовы:

- write — для вывода сообщений и матриц на стандартные потоки STDOUT и STDERR.
- sem_init — для инициализации семафора, используемого для синхронизации потоков.
- sem_wait и sem_post — для блокировки и освобождения семафора, что координирует работу потоков.
- pthread_create — для создания потоков.
- pthread_join — для ожидания завершения потоков.
- sem_destroy — для освобождения ресурсов семафора после завершения работы.

Программа применяет медианный фильтр к матрице целых чисел, распределяя обработку строк между несколькими потоками. Алгоритм работает следующим образом: матрица создаётся случайным образом, и для каждого элемента рассчитывается медиана значений из локального окна заданного размера. Потоки синхронизируются с помощью семафоров, чтобы избежать конфликтов при одновременной работе. Каждый поток отвечает за обработку определённого набора строк матрицы, после чего результаты обновляются в основной матрице. После завершения всех итераций программа выводит финальную матрицу, очищенную от "шума" за счёт фильтрации.

Кол-во потоков	Производительность
1	3.201c
5	0.714 c
10	0.432 c

Код программы

main.c

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

#define ROWS 10
#define COLS 10
```

```
#define NUM_THREADS 4

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

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

typedef enum {
    OK,
    INVALID_INPUT,
    SYSTEM_ERROR
} return_code;

sem_t semaphore;

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_philter(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 iterations = thread_args->iterations;
    int offset = window_size / 2;

    for (int iter = 0; iter < iterations; iter++) {
        sem_wait(&semaphore);

        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(&semaphore);
    }
    return NULL;
}


void print_message(const char* message) {
    write(STDOUT_FILENO, message, sizeof(char) * strlen(message));
}


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];
        }
    }
}

int is_number(const char* str) {
    while (*str) {
        if (*str < '0' || *str > '9') return 0;
        str++;
    }
    return INVALID_INPUT;
}

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;
        }
    }
}


void int_to_str(int num, char* buffer) {
    char temp[12];
    int i = 0, j = 0;
```

```
    if (num < 0) {
        buffer[j++] = '-';
        num = -num;
    }

    do {
        temp[i++] = (num % 10) + '0';
        num /= 10;
    } while (num > 0);

    while (i > 0) {
        buffer[j++] = temp[--i];
    }

    buffer[j] = '\0';
}

void print_matrix(int mat[ROWS][COLS]) {
    char buffer[64];
    for (int i = 0; i < ROWS; i++) {
        int offset = 0;
        for (int j = 0; j < COLS; j++) {
            char num_str[12];
            int_to_str(mat[i][j], num_str);
            int len = strlen(num_str);
            if (offset + len + 1 < sizeof(buffer)) {
                memcpy(buffer + offset, num_str, len);
                offset += len;

                buffer[offset++] = ' ';
            }
        }

        buffer[offset - 1] = '\n';

        write(STDOUT_FILENO, buffer, offset);
    }
}

int main(int argc, char* argv[]) {
    if (argc != 4) {
        char msg[] = "Usage: ./a.out <window_size> <iterations> <max_threads>\n";
        write(STDERR_FILENO, msg, sizeof(msg) - 1);
        return INVALID_INPUT;
    }

    int window_size, iterations, max_threads;

    if (!is_number(argv[1]) || !is_number(argv[2]) || !is_number(argv[3])) {
```

```
        char msg[] = "all arguments must be positive integers.\n";
        write(STDERR_FILENO, msg, sizeof(msg) - 1);
        return INVALID_INPUT;
    }
    window_size = atoi(argv[1]);
    iterations = atoi(argv[2]);
    max_threads = atoi(argv[3]);

    if (window_size < 2) {
        char msg[] = "window size must be >= 2\n";
        write(STDERR_FILENO, msg, sizeof(msg) - 1);
        return INVALID_INPUT;
    }

    if (iterations <= 0 || max_threads <= 0) {
        char msg[] = "mterations and threads count must be positive integers\n";
        write(STDERR_FILENO, msg, sizeof(msg) - 1);
        return INVALID_INPUT;
    }

    if (max_threads > ROWS) {
        char msg[] = "max threads cannot exceed the number of rows in the matrix\n";
        write(STDERR_FILENO, msg, sizeof(msg) - 1);
        return INVALID_INPUT;
    }

    if (sem_init(&semaphore, 0, max_threads) != 0) {
        char msg[] = "couldn't initialize semaphore.\n";
        write(STDERR_FILENO, msg, sizeof(msg) - 1);
        return SYSTEM_ERROR;
    }

    generate_matrix();

    char msg[] = "original matrix:\n";
    write(STDOUT_FILENO, msg, sizeof(msg) - 1);
    print_matrix(matrix);

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

    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;
        thread_args[i].iterations = iterations;
```

```
        if (pthread_create(&threads[i], NULL, median_philter, &thread_args[i]) != 0) {
            char msg[] = "couldn't create thread.\n";
            write(STDERR_FILENO, msg, sizeof(msg) - 1);
            return SYSTEM_ERROR;
        }
    }

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

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

        copy_temp_to_matrix();
    }
    for (int i = 0; i < max_threads; i++) {
        if (pthread_join(threads[i], NULL) != 0) {
            char msg[] = "couldn't join thread\n";
            write(STDERR_FILENO, msg, sizeof(msg) - 1);
            return SYSTEM_ERROR;
        }
    }

    char final_msg[] = "result matrix:\n";
    write(STDOUT_FILENO, final_msg, sizeof(final_msg) - 1);
    print_matrix(matrix);

    sem_destroy(&semaphore);

    return OK;
}
```

Протокол работы программы

Тестирование:

```
(gdb) run 3 2 10
Starting program: /mnt/c/Users/huawei/Desktop/3_cem/labs/os/a.out 3 2 10
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
original matrix:
78 88 13 54 83 19 51 21 72 27
18 72 83 89 96 70 10 35 54 12
14 70 42 17 45 40 31 84 27 10
95 57 50 9 63 33 80 14 54 52
41 24 77 24 13 73 46 75 61 0
39 27 23 81 45 20 73 28 5 52
90 52 61 41 13 24 26 93 38 33
98 31 57 75 8 71 0 54 46 61
7 38 89 82 71 86 2 45 14 7
49 57 12 11 98 25 35 24 19 26
[New Thread 0x7ffff7d89640 (LWP 5010)]
[New Thread 0x7ffff7588640 (LWP 5011)]
[Thread 0x7ffff7d89640 (LWP 5010) exited]
[New Thread 0x7ffff6d87640 (LWP 5012)]
[Thread 0x7ffff7588640 (LWP 5011) exited]
[Thread 0x7ffff6d87640 (LWP 5012) exited]
[New Thread 0x7ffff6586640 (LWP 5013)]
[Thread 0x7ffff6586640 (LWP 5013) exited]
[New Thread 0x7ffff5d85640 (LWP 5014)]
[Thread 0x7ffff5d85640 (LWP 5014) exited]
[New Thread 0x7ffff5584640 (LWP 5015)]
[New Thread 0x7ffff4d83640 (LWP 5016)]
[Thread 0x7ffff5584640 (LWP 5015) exited]
[Thread 0x7ffff4d83640 (LWP 5016) exited]
[New Thread 0x7ffff4582640 (LWP 5017)]
[Thread 0x7ffff4582640 (LWP 5017) exited]
[New Thread 0x7ffff3d81640 (LWP 5018)]
[Thread 0x7ffff3d81640 (LWP 5018) exited]
[New Thread 0x7ffff3580640 (LWP 5019)]
[Thread 0x7ffff3580640 (LWP 5019) exited]
result matrix:
78 88 13 54 83 19 51 21 72 27
18 70 70 54 54 45 35 35 27 12
14 57 57 50 45 45 35 35 35 10
95 50 42 42 33 45 46 54 52 52
41 41 27 45 33 46 46 54 52 0
39 41 41 41 24 26 46 46 38 52
90 52 52 45 41 24 28 38 46 33
98 57 57 61 71 24 45 38 45 61
7 49 57 71 71 35 35 24 26 7
49 57 12 11 98 25 35 24 19 26
[Inferior 1 (process 5006) exited normally]
```

string5

Strace:

$

strace -f ./a.out 3 2 4

execve("./a.out", ["./a.out", "3", "2", "4"], 0x7fff42cefd50 /* 27 vars */) = 0

brk(NULL) = 0x5582dd9a4000

```
     arch_prctl(0x3001 /* ARCH_??? */, 0x7ffd32735350) = -1 EINVAL (Invalid argument)

     mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f882b164000

     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=17839, ...}, AT_EMPTY_PATH) = 0

     mmap(NULL, 17839, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f882b15f000

     close(3)                                = 0

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

     read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\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) = 0x7f882af36000

     mprotect(0x7f882af5e000, 2023424, PROT_NONE) = 0

     mmap(0x7f882af5e000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x28000) = 0x7f882af5e000

     mmap(0x7f882b0f3000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1bd000) = 0x7f882b0f3000

     mmap(0x7f882b14c000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x215000) = 0x7f882b14c000

     mmap(0x7f882b152000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7f882b152000

     close(3)                                = 0

     mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f882af33000

     arch_prctl(ARCH_SET_FS, 0x7f882af33740) = 0

     set_tid_address(0x7f882af33a10)         = 9817

     set_robust_list(0x7f882af33a20, 24)     = 0

     rseq(0x7f882af340e0, 0x20, 0, 0x53053053) = 0

     mprotect(0x7f882b14c000, 16384, PROT_READ) = 0

     mprotect(0x5582d3e21000, 4096, PROT_READ) = 0

     mprotect(0x7f882b19e000, 8192, PROT_READ) = 0
```

```
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f882b15f000, 17839)           = 0
write(1, "original matrix:\n", 17original matrix:
)      = 17
write(1, "28 50 37 21 41 80 98 28 67 67\n", 3028 50 37 21 41 80 98 28 67 67
) = 30
write(1, "11 34 87 54 40 4 42 3 82 90\n", 2811 34 87 54 40 4 42 3 82 90
) = 28
write(1, "23 25 59 11 12 91 11 45 69 9\n", 2923 25 59 11 12 91 11 45 69 9
) = 29
write(1, "39 49 59 28 71 52 61 69 80 28\n", 3039 49 59 28 71 52 61 69 80 28
) = 30
write(1, "36 43 63 75 98 55 31 92 10 65\n", 3036 43 63 75 98 55 31 92 10 65
) = 30
write(1, "82 34 90 94 97 54 85 8 0 54\n", 2882 34 90 94 97 54 85 8 0 54
) = 28
write(1, "17 91 4 28 19 27 80 32 96 13\n", 2917 91 4 28 19 27 80 32 96 13
) = 29
write(1, "61 84 56 76 59 6 83 42 51 94\n", 2961 84 56 76 59 6 83 42 51 94
) = 29
write(1, "7 85 80 50 79 77 4 16 85 4\n", 277 85 80 50 79 77 4 16 85 4
) = 27
write(1, "71 54 47 27 82 19 54 14 51 2\n", 2971 54 47 27 82 19 54 14 51 2
) = 29
rt_sigaction(SIGRT_1, {sa_handler=0x7f882afc7870, sa_mask=[],
sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO, sa_restorer=0x7f882af78520}, 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) =
0x7f882a732000
mprotect(0x7f882a733000, 8388608, PROT_READ|PROT_WRITE) = 0
getrandom("\x40\x95\x5b\x0e\xe6\x7a\xec\x52", 8, GRND_NONBLOCK) = 8
brk(NULL)                               = 0x5582dd9a4000
brk(0x5582dd9c5000)                     = 0x5582dd9c5000
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CL
ONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f882af32910,
```

```
parent_tid=0x7f882af32910, exit_signal=0, stack=0x7f882a732000, stack_size=0x7fff00,
tls=0x7f882af32640}strace: Process 9819 attached

      => {parent_tid=[9819]}, 88) = 9819

     [pid  9819] rseq(0x7f882af32fe0, 0x20, 0, 0x53053053 <unfinished ...>

     [pid  9817] rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     [pid  9819] <... rseq resumed>)         = 0

     [pid  9817] <... rt_sigprocmask resumed>NULL, 8) = 0

     [pid  9819] set_robust_list(0x7f882af32920, 24 <unfinished ...>

     [pid  9817] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0
<unfinished ...>

     [pid  9819] <... set_robust_list resumed>) = 0

     [pid  9817] <... mmap resumed>)         = 0x7f8829f31000

     [pid  9819] rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     [pid  9817] mprotect(0x7f8829f32000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

     [pid  9819] <... rt_sigprocmask resumed>NULL, 8) = 0

     [pid  9817] <... mprotect resumed>)     = 0

     [pid  9817] rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>

     [pid  9819] rt_sigprocmask(SIG_BLOCK, ~[RT_1],  <unfinished ...>

     [pid  9817] <... rt_sigprocmask resumed>[], 8) = 0

     [pid  9819] <... rt_sigprocmask resumed>NULL, 8) = 0

     [pid  9817]
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=0x7f882a731910,
parent_tid=0x7f882a731910, exit_signal=0, stack=0x7f8829f31000, stack_size=0x7fff00,
tls=0x7f882a731640} <unfinished ...>

     [pid  9819] madvise(0x7f882a732000, 8368128, MADV_DONTNEED) = 0

     strace: Process 9820 attached

     [pid  9819] exit(0 <unfinished ...>

     [pid  9817] <... clone3 resumed> => {parent_tid=[9820]}, 88) = 9820

     [pid  9820] rseq(0x7f882a731fe0, 0x20, 0, 0x53053053 <unfinished ...>

     [pid  9817] rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     [pid  9819] <... exit resumed>)         = ?

     [pid  9817] <... rt_sigprocmask resumed>NULL, 8) = 0

     [pid  9820] <... rseq resumed>)         = 0

     [pid  9817] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0
<unfinished ...>

     [pid  9820] set_robust_list(0x7f882a731920, 24 <unfinished ...>

     [pid  9817] <... mmap resumed>)         = 0x7f8829730000

     [pid  9820] <... set_robust_list resumed>) = 0
```

```
     [pid  9819] +++ exited with 0 +++

     [pid  9817] mprotect(0x7f8829731000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

     [pid  9820] rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     [pid  9817] <... mprotect resumed>)     = 0

     [pid  9820] <... rt_sigprocmask resumed>NULL, 8) = 0

     [pid  9817] rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>

     [pid  9820] rt_sigprocmask(SIG_BLOCK, ~[RT_1],  <unfinished ...>

     [pid  9817] <... rt_sigprocmask resumed>[], 8) = 0

     [pid  9820] <... rt_sigprocmask resumed>NULL, 8) = 0

     [pid  9817]
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=0x7f8829f30910,
parent_tid=0x7f8829f30910, exit_signal=0, stack=0x7f8829730000, stack_size=0x7fff00,
tls=0x7f8829f30640} <unfinished ...>

     [pid  9820] madvise(0x7f8829f31000, 8368128, MADV_DONTNEED) = 0

     strace: Process 9821 attached

     [pid  9817] <... clone3 resumed> => {parent_tid=[9821]}, 88) = 9821

     [pid  9820] exit(0 <unfinished ...>

     [pid  9817] rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     [pid  9821] rseq(0x7f8829f30fe0, 0x20, 0, 0x53053053 <unfinished ...>

     [pid  9817] <... rt_sigprocmask resumed>NULL, 8) = 0

     [pid  9820] <... exit resumed>)         = ?

     [pid  9817] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0
<unfinished ...>

     [pid  9821] <... rseq resumed>)         = 0

     [pid  9817] <... mmap resumed>)         = 0x7f8828f2f000

     [pid  9820] +++ exited with 0 +++

     [pid  9817] mprotect(0x7f8828f30000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

     [pid  9821] set_robust_list(0x7f8829f30920, 24 <unfinished ...>

     [pid  9817] <... mprotect resumed>)     = 0

     [pid  9821] <... set_robust_list resumed>) = 0

     [pid  9817] rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>

     [pid  9821] rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0

     [pid  9817] <... rt_sigprocmask resumed>[], 8) = 0

     [pid  9821] rt_sigprocmask(SIG_BLOCK, ~[RT_1],  <unfinished ...>

     [pid  9817]
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=0x7f882972f910,
parent_tid=0x7f882972f910, exit_signal=0, stack=0x7f8828f2f000, stack_size=0x7fff00,
tls=0x7f882972f640} <unfinished ...>
```

```
[pid  9821] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid  9821] madvise(0x7f8829730000, 8368128, MADV_DONTNEEDstrace: Process 9822 attached
) = 0
[pid  9817] <... clone3 resumed> => {parent_tid=[9822]}, 88) = 9822
[pid  9822] rseq(0x7f882972ffe0, 0x20, 0, 0x53053053 <unfinished ...>
[pid  9817] rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
[pid  9821] exit(0 <unfinished ...>
[pid  9817] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid  9822] <... rseq resumed>)         = 0
[pid  9817] futex(0x7f8829f30910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9821, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
[pid  9821] <... exit resumed>)         = ?
[pid  9822] set_robust_list(0x7f882972f920, 24 <unfinished ...>
[pid  9817] <... futex resumed>)        = 0
[pid  9822] <... set_robust_list resumed>) = 0
[pid  9821] +++ exited with 0 +++
[pid  9817] futex(0x7f882972f910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9822, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
[pid  9822] rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
[pid  9822] rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
[pid  9822] madvise(0x7f8828f2f000, 8368128, MADV_DONTNEED) = 0
[pid  9822] exit(0)                     = ?
[pid  9822] +++ exited with 0 +++
<... futex resumed>)                    = 0
write(1, "result matrix:\n", 15result matrix:
)        = 15
write(1, "28 50 37 21 41 80 98 28 67 67\n", 3028 50 37 21 41 80 98 28 67 67
) = 30
write(1, "11 34 37 40 40 41 42 45 67 90\n", 3011 34 37 40 40 41 42 45 67 90
) = 30
write(1, "23 39 49 54 40 42 45 61 69 9\n", 2923 39 49 54 40 42 45 61 69 9
) = 29
write(1, "39 43 49 59 55 55 55 61 65 28\n", 3039 43 49 59 55 55 55 61 65 28
) = 30
write(1, "36 49 59 75 71 61 55 61 54 65\n", 3036 49 59 75 71 61 55 61 54 65
) = 30
```

```
write(1, "82 43 63 75 55 55 54 32 32 54\n", 3082 43 63 75 55 55 54 32 32 54
) = 30
write(1, "0 0 0 0 0 0 0 0 0 0\n", 200 0 0 0 0 0 0 0 0 0
)   = 20
write(1, "0 0 0 0 0 0 0 0 0 0\n", 200 0 0 0 0 0 0 0 0 0
)   = 20
write(1, "0 0 0 0 0 0 0 0 0 0\n", 200 0 0 0 0 0 0 0 0 0
)   = 20
write(1, "0 0 0 0 0 0 0 0 0 0\n", 200 0 0 0 0 0 0 0 0 0
)   = 20
exit_group(0)                           = ?
+++ exited with 0 +++
```

Вывод

В данной работе был реализован алгоритм применения медианного фильтра к матрице целых чисел с использованием многопоточности. Программа позволяет задавать параметры фильтра (размер окна, количество итераций, число потоков) через аргументы командной строки, обеспечивая гибкость использования. Для синхронизации потоков и защиты общих данных были использованы семафоры. Результаты показали, что увеличение количества потоков ускоряет обработку, однако производительность может снижаться из-за накладных расходов на синхронизацию и управление потоками. Работа демонстрирует эффективное использование системных вызовов и многопоточности для обработки данных.