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-210Б-23

Студент: Гончаров Роман Витальевич

Преподаватель: Бахарев В.Д. Оценка:

Дата: 11.03.2025

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 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); Дожидается 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 впоследствие 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.

# 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**

roma3366@Revision-PC:~/LABS-OSI/Lab2$ ./prog 3 1 1

Original matrix:

3 55 73 26 26 91 93 63 72 17

75 3 12 59 41 29 74 77 78 10

46 32 38 15 50 79 5 98 56 56

68 10 62 40 87 87 82 79 1 53

47 27 7 11 37 47 39 10 75 68

71 21 99 8 87 48 87 43 97 94

50 64 55 11 55 93 49 36 71 1

88 69 79 94 79 15 93 69 76 67

36 46 39 87 6 25 86 92 19 83

37 68 98 43 30 53 35 78 88 57

Result matrix:

3 55 73 26 26 91 93 63 72 17

75 38 32 38 41 50 77 74 63 10

46 38 32 41 50 74 79 77 56 56

68 38 27 38 47 50 79 56 56 53

47 47 21 40 47 82 48 75 68 68

71 50 21 37 47 49 47 49 68 94

50 69 64 79 55 79 49 71 69 1

88 55 64 55 55 55 69 71 69 67

36 68 69 79 43 35 69 78 76 83

37 68 98 43 30 53 35 78 88 57

Time taken: 0.000047 seconds

roma3366@Revision-PC:~/LABS-OSI/Lab2$ ./prog 3 2 1

Original matrix:

51 90 5 20 9 89 84 87 49 37

30 90 35 32 32 33 45 24 93 8

69 19 61 99 44 10 49 5 33 72

10 83 14 14 54 74 55 90 13 55

26 94 44 60 25 27 93 22 50 37

29 70 7 41 69 50 51 69 6 83

93 15 17 58 81 22 83 35 11 95

41 36 41 84 48 65 63 92 86 64

28 67 34 34 59 54 35 9 22 40

43 66 7 11 75 87 33 58 73 43

Result matrix:

51 90 5 20 9 89 84 87 49 37

30 35 35 32 33 45 45 45 45 8

69 35 35 35 44 45 45 45 45 72

10 35 41 44 44 49 49 49 49 55

26 29 44 44 49 50 50 50 50 37

29 36 41 44 50 54 54 51 51 83

93 36 41 48 51 58 54 50 63 95

41 36 41 48 58 59 59 58 58 64

28 36 36 41 59 59 58 58 58 40

43 66 7 11 75 87 33 58 73 43

Time taken: 0.000254 seconds

roma3366@Revision-PC:~/LABS-OSI/Lab2$ ./prog 4 1 2

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

roma3366@Revision-PC:~/LABS-OSI/Lab2$ ./prog 1 2 3

Original matrix:

21 97 35 64 1 77 24 4 44 29

1 12 67 12 49 89 49 33 42 50

95 8 16 63 100 69 15 45 63 93

87 35 41 21 50 42 49 73 97 92

54 49 56 20 60 4 61 60 88 2

9 34 9 76 96 60 44 62 4 58

6 42 92 47 62 93 88 62 66 36

54 71 84 9 90 95 64 2 54 3

55 15 89 15 90 36 74 86 98 29

43 3 70 87 49 83 79 88 45 96

Result matrix:

21 97 35 64 1 77 24 4 44 29

1 12 67 12 49 89 49 33 42 50

95 8 16 63 100 69 15 45 63 93

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

6 42 92 47 62 93 88 62 66 36

54 71 84 9 90 95 64 2 54 3

55 15 89 15 90 36 74 86 98 29

43 3 70 87 49 83 79 88 45 96

Time taken: 0.000229 seconds

roma3366@Revision-PC:~/LABS-OSI/Lab2$ ./prog 2 1 3

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

roma3366@Revision-PC:~/LABS-OSI/Lab2$ ./prog 3 1 1

Original matrix:

76 20 40 90 81 21 59 21 88 61

12 82 94 52 9 59 74 27 54 98

52 10 16 27 39 90 2 61 95 38

74 22 57 66 11 37 86 22 9 73

82 20 6 75 71 15 33 44 93 38

93 96 99 60 22 38 49 24 98 95

13 23 69 69 88 79 57 25 52 65

50 34 84 55 60 54 21 45 50 13

82 94 61 81 6 82 70 54 57 19

1 69 41 69 89 81 99 45 5 51

Result matrix:

76 20 40 90 81 21 59 21 88 61

12 40 40 40 52 59 59 59 61 98

52 52 52 39 39 39 59 54 54 38

74 22 22 39 39 37 37 44 44 73

82 74 60 60 38 37 37 44 44 38

93 69 69 69 69 49 38 49 52 95

13 69 69 69 60 54 45 49 50 65

50 61 69 69 69 60 54 52 50 13

82 61 69 61 69 70 54 50 45 19

1 69 41 69 89 81 99 45 5 51

Time taken: 0.000240 seconds

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|SIGCHLDstrace: 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]  <unfinished | | execve("./laba2", ["./laba2", "3", "1", "2"], 0x7ffe64c4ae68 /\* 27 vars  ...> | \*/ |
| [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 832) | 2769]  = 832 | 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"..., | |
| [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\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  = 0 | 2769] | prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) |
| [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,**  **FUTEX\_BITSET\_MATCH\_ANY <unfinished ...>** | **0,** | **NULL,** |
| **[pid 2771] futex(0x564837627380, FUTEX\_WAIT\_BITSET\_PRIVATE|FUTEX\_CLOCK\_REALTIME,**  **FUTEX\_BITSET\_MATCH\_ANY <unfinished ...>** | **0,** | **NULL,** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **[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 +++ | | |  |  | |

# 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.