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: М80-206Б-22

Студент: Ларин И. А.

Преподаватель: Миронов Е.С.

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: 31.10.23

Москва, 2023

**Постановка задачи**

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

**Общие сведения о программе**  
Использованные системные вызовы:

* **pthread\_create**(&threadID, NULL, pfunction, &pdata)) - создает новый поток с номером threadID, запускает потоковую функцию pfunction и передает в нее данные pdata.
* **pthread\_join**(threadID, NULL) - ждет завершение текущего потока.

Программе задаём общее число потоков ( или поумолчанию в двупоточном). Задаём размер матрицы, размер окна, количество фильтров.

Программа делит матрицу на строки и и распределяет потокам строки поровну, однако один поток может обрабатывать на одну строку больше. Программа работает с двумя матрицами, чтобы потоки не пересекались при выполнении своей задачи: в случае одного медианного фильтра, результаты записываются во 2-ую матрицу, в случае двух медианных фильтров, начальной матрицей для потоков становится матрица 2, а в первую матрицу записываем ответ.

**Исходный код**

Lab.h

#include <vector>

#include <iostream>

#include <string>

#include <pthread.h>

#include <ctime>

using namespace std;

using TMatrix = vector<vector<int>>;

struct thread\_args{

    int hgh;

    int wth;

    TMatrix \*f\_mat;

    TMatrix \*s\_mat;

    int obl;

    int l\_row;

    int upr\_row;

};

void median\_filter(int &hgh, int &wth, TMatrix &a, TMatrix &b, int &obl, int &l\_row, int &upr\_row);

void Work(int &n, int &m, TMatrix &mat1, TMatrix &mat2, int &k, int &win\_s, int &threadc, TMatrix &res);

lab.cpp

#include "lab.h"

#include <ctime>

void median\_filter(int &hgh, int &wth, TMatrix &a, TMatrix &b, int &obl, int &l\_row, int &upr\_row){

    int rad = (obl - 1) / 2;

    for (int y = l\_row; y < upr\_row; y++){

        int top;

        if ((y-rad) > 0)

            top = y - rad;

        else

            top = 0;

        int bottom;

        if ((y+rad) < 0)

            bottom = y+rad;

        else

            bottom = hgh - 1;

        for (int x = 0; x < wth; x++){

            int left;

            if ((x- rad) > 0)

                left = x -rad;

            else

                left = 0;

            int right;

            if((x+rad) < (wth - 1))

                right = x + rad;

            else

                right = wth - 1;

            vector <int> m((bottom - top + 1) \* (right - left + 1));

            int k = 0;

            for (int v = top; v <= bottom; v++){

                for (int u = left; u <= right; u++){

                    m[k] = a[v][u];

                    k++;

                }

            }

            int n = m.size();

                for(int i = 0; i < n; i++){

                    for(int j = i; j < n; j++){

                        if(m[i] > m[j]){

                            int t = m[i];

                            m[i] = m[j];

                            m[j] = t;

                        }

                    }

                }

            b[y][x] = m[n / 2];

        }

    }

}

void \*liner(void \*args){

    thread\_args \*arg = (thread\_args\*) args;

    median\_filter(arg->hgh, arg->wth, \*arg->f\_mat, \*arg->s\_mat, arg->obl, arg->l\_row, arg->upr\_row);

    return NULL;

}

void Work(int &n, int &m, TMatrix &mat1, TMatrix &mat2, int &k, int &win\_s, int &threadc, TMatrix &res){

    int imed = n / threadc;

    pthread\_t threads[threadc];

    thread\_args p\_args[threadc];

    for (int i  = 0; i < k; i++){

        for(int j = 0; j < threadc; j++){

            p\_args[j].hgh = n;

            p\_args[j].wth = m;

            if (i % 2 == 0){

                p\_args[j].f\_mat = &mat1;

                p\_args[j].s\_mat = &mat2;

            }

            else{

                p\_args[j].f\_mat = &mat2;

                p\_args[j].s\_mat = &mat1;

            }

            p\_args[j].obl = win\_s;

            p\_args[j].l\_row = j \* imed;

            p\_args[j].upr\_row = (j + 1) \* imed;

            if (j == threadc - 1)

                p\_args[j].upr\_row = n;

        }

        for (int i = 0; i < threadc; i++){

            int res = pthread\_create(&threads[i], NULL, liner, &p\_args[i]);

            if (res != 0){

                cout << "Ошибка";

                exit(-2);

            }

        }

        for (int i = 0; i < threadc; i++){

            int res = pthread\_join(threads[i], NULL);

            if (res != 0){

                cout << "Ошибка";

                exit(-3);

            }

        }

        printf("\n");

    }

    if (k % 2 == 1){

        res = mat2;}

    else {

        res = mat1;}

}

Main.cpp

#include "lab.h"

#include <ctime>

int main(int argc, char\* argv[]){

    int n, m, k, threadc, win\_s;

    TMatrix res;

    cout << "Потоки или поток (количество): ";

    cin >> threadc;

    cout << "Размеры матрицы: ";

    cin >> n >> m;

    TMatrix mat1(n, vector<int>(m));

    TMatrix mat2(n, vector<int>(m));

    if (threadc > n)

        threadc = n;

    printf("Размер окна: ");

    cin >> win\_s;

    printf("Медианные фильтры(количество): ");

    cin >> k;

    printf("Матрица:\n");

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < m; j++) {

            cin >> mat1[i][j];

        }

    }

    //unsigned int start\_time =  clock(); // начальное время

    Work(n, m, mat1, mat2, k, win\_s, threadc, res);

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < m; j++) {

            cout << res[i][j] << " ";

        }

        cout << "\n";

    }

    //unsigned int end\_time = clock(); // конечное время

    //unsigned int search\_time = end\_time - start\_time; // искомо

    //cout << search\_time << "endl";

    return 0;

}

**Демонстрация работы программы**ilya@ilya-ilyal:~/osi\_11$ time ./a.out

Потоки или поток (количество): 1

Размеры матрицы: 3 3

Размер окна: 3

Медианные фильтры(количество): 1

Матрица:

1 2 3

3 2 5

1 5 6

2 3 3

2 3 5

3 5 5

ilya@ilya-ilyal:~/osi\_11$ ./a.out

Потоки или поток (количество): 3

Размеры матрицы: 4 4

Размер окна: 3

Медианные фильтры: 2

Матрица:

1 2 3 4

3 2 4 5

2 1 1 4

4 4 5 5

2 3 4 4

3 3 4 4

3 3 4 4

4 4 4 5

**Strace**

ilya@ilya-ilyal:~/osi\_11$ strace -f ./a.out < 1.in

execve("./a.out", ["./a.out"], 0x7fff9af22438 /\* 59 vars \*/) = 0

brk(NULL) = 0x55873aa09000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7fff653fe560) = -1 EINVAL (Недопустимый аргумент)

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fc91c44f000

access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (Нет такого файла или каталога)

openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=53143, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 53143, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7fc91c442000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libstdc++.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\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=2260296, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 2275520, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fc91c200000

mprotect(0x7fc91c29a000, 1576960, PROT\_NONE) = 0

mmap(0x7fc91c29a000, 1118208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9a000) = 0x7fc91c29a000

mmap(0x7fc91c3ab000, 454656, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ab000) = 0x7fc91c3ab000

mmap(0x7fc91c41b000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x21a000) = 0x7fc91c41b000

mmap(0x7fc91c429000, 10432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fc91c429000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=125488, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 127720, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fc91c1e0000

mmap(0x7fc91c1e3000, 94208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7fc91c1e3000

mmap(0x7fc91c1fa000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a000) = 0x7fc91c1fa000

mmap(0x7fc91c1fe000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d000) = 0x7fc91c1fe000

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\0\"\233}\305\t\5?\344\337^)\350b\231\21\360"..., 68, 896) = 68

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=2216304, ...}, 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, 2260560, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fc91be00000

mmap(0x7fc91be28000, 1658880, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7fc91be28000

mmap(0x7fc91bfbd000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1bd000) = 0x7fc91bfbd000

mmap(0x7fc91c015000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x214000) = 0x7fc91c015000

mmap(0x7fc91c01b000, 52816, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fc91c01b000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libm.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\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=940560, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 942344, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fc91c0f9000

mmap(0x7fc91c107000, 507904, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe000) = 0x7fc91c107000

mmap(0x7fc91c183000, 372736, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x8a000) = 0x7fc91c183000

mmap(0x7fc91c1de000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe4000) = 0x7fc91c1de000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fc91c440000

arch\_prctl(ARCH\_SET\_FS, 0x7fc91c4413c0) = 0

set\_tid\_address(0x7fc91c441690) = 23621

set\_robust\_list(0x7fc91c4416a0, 24) = 0

rseq(0x7fc91c441d60, 0x20, 0, 0x53053053) = 0

mprotect(0x7fc91c015000, 16384, PROT\_READ) = 0

mprotect(0x7fc91c1de000, 4096, PROT\_READ) = 0

mprotect(0x7fc91c1fe000, 4096, PROT\_READ) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fc91c43e000

mprotect(0x7fc91c41b000, 45056, PROT\_READ) = 0

mprotect(0x558739810000, 4096, PROT\_READ) = 0

mprotect(0x7fc91c489000, 8192, PROT\_READ) = 0

prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

munmap(0x7fc91c442000, 53143) = 0

getrandom("\x9c\x35\x25\x10\xe4\xa4\x3d\x1e", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x55873aa09000

brk(0x55873aa2a000) = 0x55873aa2a000

futex(0x7fc91c42977c, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

newfstatat(1, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}, AT\_EMPTY\_PATH) = 0

write(1, "\320\237\320\276\321\202\320\276\320\272\320\270 \320\270\320\273\320\270 \320\277\320\276\321\202\320\276\320\272 ("..., 55Потоки или поток (количество): ) = 55

newfstatat(0, "", {st\_mode=S\_IFREG|0664, st\_size=27, ...}, AT\_EMPTY\_PATH) = 0

read(0, "1\n3 3\n3\n1\n1 2 3\n3 2 5\n1 5 6", 4096) = 27

write(1, "\320\240\320\260\320\267\320\274\320\265\321\200\321\213 \320\274\320\260\321\202\321\200\320\270\321\206\321\213: ", 31Размеры матрицы: ) = 31

write(1, "\320\240\320\260\320\267\320\274\320\265\321\200 \320\276\320\272\320\275\320\260: ", 23Размер окна: ) = 23

write(1, "\320\234\320\265\320\264\320\270\320\260\320\275\320\275\321\213\320\265 \321\204\320\270\320\273\321\214\321\202\321\200\321"..., 57Медианные фильтры(количество): ) = 57

write(1, "\320\234\320\260\321\202\321\200\320\270\321\206\320\260:\n", 16Матрица:

) = 16

read(0, "", 4096) = 0

rt\_sigaction(SIGRT\_1, {sa\_handler=0x7fc91be918f0, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_ONSTACK|SA\_RESTART|SA\_SIGINFO, sa\_restorer=0x7fc91be42520}, 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) = 0x7fc91b5ff000

mprotect(0x7fc91b600000, 8388608, PROT\_READ|PROT\_WRITE) = 0

rt\_sigprocmask(SIG\_BLOCK, ~[], [], 8) = 0

clone3({flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, child\_tid=0x7fc91bdff910, parent\_tid=0x7fc91bdff910, exit\_signal=0, stack=0x7fc91b5ff000, stack\_size=0x7fff00, tls=0x7fc91bdff640}strace: Process 23622 attached

=> {parent\_tid=[23622]}, 88) = 23622

[pid 23622] rseq(0x7fc91bdfffe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 23621] rt\_sigprocmask(SIG\_SETMASK, [], <unfinished ...>

[pid 23622] <... rseq resumed>) = 0

[pid 23621] <... rt\_sigprocmask resumed>NULL, 8) = 0

[pid 23622] set\_robust\_list(0x7fc91bdff920, 24 <unfinished ...>

[pid 23621] futex(0x7fc91bdff910, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 23622, NULL, FUTEX\_BITSET\_MATCH\_ANY <unfinished ...>

[pid 23622] <... set\_robust\_list resumed>) = 0

[pid 23622] rt\_sigprocmask(SIG\_SETMASK, [], NULL, 8) = 0

[pid 23622] mmap(NULL, 134217728, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_NORESERVE, -1, 0) = 0x7fc9135ff000

[pid 23622] munmap(0x7fc9135ff000, 10489856) = 0

[pid 23622] munmap(0x7fc918000000, 56619008) = 0

[pid 23622] mprotect(0x7fc914000000, 135168, PROT\_READ|PROT\_WRITE) = 0

[pid 23622] rt\_sigprocmask(SIG\_BLOCK, ~[RT\_1], NULL, 8) = 0

[pid 23622] madvise(0x7fc91b5ff000, 8368128, MADV\_DONTNEED) = 0

[pid 23622] exit(0) = ?

[pid 23621] <... futex resumed>) = 0

[pid 23622] +++ exited with 0 +++

write(1, "\n", 1

) = 1

write(1, "2 3 5 \n", 72 3 5

) = 7

write(1, "2 3 5 \n", 72 3 5

) = 7

write(1, "3 5 5 \n", 73 5 5

) = 7

exit\_group(0) = ?

+++ exited with 0 +++

|  |  |  |  |
| --- | --- | --- | --- |
| Число потоков | Время исполнения(с) | Ускорение | Эффективность |
| 1 | 1,750 | 1 | 1 |
| 2 | 1,445 | 1.211 | 0,6055 |
| 3 | 1,328 | 1,318 | 0,4393 |

**Вывод**

Я узнал, что такое

1.Управлении потоками в ОС

2.Обеспечении синхронизации между потоками

3.Медианный фильтр матрицы

Я уверен, что эти знания мне пригодятся в будущем.