

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: Ларченко А.О.

Преподаватель: Миронов Е.С.

Оценка: _____

Дата: 01.12.23

Москва, 2023

Постановка задачи

Вариант 11.

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.

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

Общий метод и алгоритм решения

Использованные системные вызовы:

- `int pthread_create(pthread_t *restrict thread, const pthread_attr_t *restrict attr, void*(*start_routine)(void*), void *restrict arg)` - создает поток по указателю на поток `*restrict thread`, и передает ему на запуск процедуру `(void*)`, аргументом которой является структура `*restrict arg`
- `int pthread_join(pthread_t thread, void **retval)` - ждет завершение потока `thread`, если поток завершен до вызова функции, функция сразу завершается

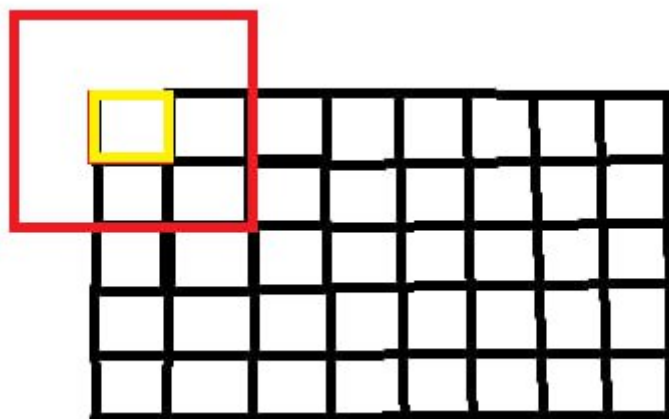
Идея: Пользователь вводит:

- размер исходной матрицы($n*m$),
- количество применений наложения медианного фильтра(q),
- размер квадратной матрицы медианного фильтра(k , $k>1$, $k\%2==1$).

Используя функцию `create_mass()` мы заполняем матрицу случайными числами. Стоит учитывать, что мы представляем матрицу в виде линейного массива, и поэтому для корректной обработки матрицы мы добавляем столбцы и линии, содержащие 0 в нашу исходную матрицу. Функцию `filter_overlay(mass, output_mass, n, m, q, k, cnt_thread)` мы используем для разбиения массива на отрезки, которые мы будем передавать созданным потокам для обработки. Также имеем 2 матрицы: исходную и результирующую, потоки берут данные только из исходной, а записывают в результирующую. Учитывая, что каждому потоку мы передаем уникальные отрезки, мы гарантируем, что у нас не будет `data race`, поэтому мы не используем мьютексы. Каждый поток начинает исполнять функцию `void *filt_by_cell(void* args)`, её аргументом является структура `struct`

threads_shell{ int *mass; vector<coord> coords; int *output_mass; int k; int n; int m; }, которая хранит вектор отрезков, исходный массив, результирующий массив, размер матрицы и размер матричного фильтра.

Суть программы можно описать как наложение матричного фильтра на основную матрицу. Каждый поток применяет его к элементам, содержащимся на переданных ему отрезках. Поток составляет вектор из элементов, которые включает фильтрующая матрица, вектор сортируется, после чего берется медианный элемент, который записывается в результирующую матрицу на место, соответствующее обрабатываемому элементу в исходной матрице.



(Обрабатывается желтая клеточка)



Код программы

function.h

```
#pragma once
#include <iostream>
#include <time.h>
#include <pthread.h>
#include <cstdlib>
#include <chrono>
```

```

#include <ratio>
#include <iomanip>

int rnd();
void create_mass(int *mass, int *output_mass, int n, int m, int addit_line, int
n_new, int m_new);
void print_full_mass(int n, int m, int *mass);
void print_correct_mass(int n_new, int m_new, int n, int m, int addit_line, int
*mass);
void filter_overlay(int *mass, int *output_mass, int n, int m, int q, int k,
int threads_cnt);
void *filt_by_cell(void* args);
void* fill_mas(void* args);

```

threads.h

```

#pragma once
#include <iostream>
#include <sys/types.h>
#include <unistd.h>
#include <sys/syscall.h>
#include <vector>
#include <algorithm>
using namespace std;

```

```

typedef struct coord{
    int l;
    int r;
}coord;

```

```

struct threads_shell{
    int *mass;
    vector<coord> coords;
    int *output_mass;
    int k;
    int n;
    int m;
};

```

11.cpp

```

// VArIant 11
#include "../include/function.h"
using namespace std;

int main(int argc, char* argv[]){
    if(argc!=2){
        perror("Incorrect number of arguments");
        exit(-1);
    }
}

```

```

    }
    int cnt_thread=atoi(argv[1]);
    int n,m,q,k;
    cout<<"Input matrix size\n"<<"n=";
    cin>>n;
    cout<<"\nm=";
    cin>>m;
    cout<<"\nInput count of filter overlay=";
    cin>>q;
    cout<<"\nInput filter matrix size k*k (it should be odd number, and
k>1)\n";
    cout<<"k=";
    cin>>k;
    while(k%2==0 and k>1){
        cout<<"\nSorry, you should input odd size and >1. Try again.\nk=";
        cin>>k;
    }
    int addit_line=(k/2)*2; /*2!!!!
    int n_new=n+addit_line;
    int m_new=m+addit_line;

    int *mass=new int[n_new*m_new];
    int *output_mass= new int[n_new*m_new];
    // int mass[n_new*m_new];
    // int output_mass[n_new*m_new];
    for(int i=0; i<n_new*m_new;++i){
        mass[i]=0;
        output_mass[i]=0;
    }
    create_mass(mass, output_mass, n,m, addit_line, n_new, m_new);
    cout<<"Begin mass:";
    print_full_mass(n_new, m_new, mass);
    print_correct_mass(n_new, m_new, n, m, addit_line, mass);

    const auto t_begin=chrono::high_resolution_clock::now();
    filter_overlay(mass, output_mass, n, m, q, k, cnt_thread);
    const auto t_ending=chrono::high_resolution_clock::now();

    cout<<"\noutput_mass: ";
    print_full_mass(n_new, m_new, output_mass);
    print_correct_mass(n_new, m_new, n, m, addit_line, output_mass);

    chrono::duration<double> t_work=t_ending - t_begin;
    cout<<"\nTime: "<< t_work.count()<<"\n";
}

```

function.cpp

```
#include "../include/function.h"
#include "../include/threadss.h"
using namespace std;

int rnd(){
    // srand(time(NULL));
    int a =rand()%256;
    // pid_t x = syscall(__NR_gettid);
    // cout<<x<<' '<a<<'\n';
    return a;
}

void* fill_mas(void* args){
    threads_shell *part = (threads_shell*) args;

    // for(int i=(*part).l;i<=(*part).r;++i){
    //     (*part).mass[i]=rnd();
    // }
    for(int i=0;i<=(*part).coords[0].r-(*part).coords[0].l;++i){
        (*part).mass[i]=rnd();
        (*part).output_mass[i]=(*part).mass[i];
    }
    pthread_exit(0);
}

void create_mass(int *mass, int *output_mass, int n, int m, int addit_line, int
n_new, int m_new){
    for(int i=0;i<n;++i){
        srand(time(NULL)+i);
        for(int j=0; j<m;++j){
            // srand(time(NULL)+i);
            mass[(m_new*((addit_line/2)+i))+addit_line/2+j]=rnd();
            output_mass[(m_new*((addit_line/2)+i))+addit_line/2+j]=mass[(m_new*((addit_line
/2)+i))+addit_line/2+j];
        }
    }
}

void print_correct_mass(int n_new, int m_new, int n, int m, int addit_line, int
*mass){
    cout<<'\n';
    cout<<"Correct mass:\n";
    for(int i=0;i<n;++i){
        for(int j=0; j<m;++j){
```

```

        cout<<setw(3)<<mass[(m_new*((addit_line/2)+i))+addit_line/2+j]<<'
';
    }
    cout<<'\n';
}
}

void print_full_mass(int n, int m, int *mass){
    cout<<'\n';
    cout<<"Full mass:\n";
    for(int i=0; i<n;++i){
        for(int j=0; j<m;++j){
            cout<<setw(3)<<mass[i*m+j]<<' ';
        }
        cout<<'\n';
    }
}

void *filt_by_cell(void* args){
    threads_shell *t_s = (threads_shell*) args;
    // int cur_j=t_s.coords[0].l;
    int addit_line_to2=(t_s).k/2;
    int n_new=(t_s).n+2*addit_line_to2;
    int m_new=(t_s).m+2*addit_line_to2;
    for(int i=0; i<(t_s).coords.size();++i){
        coord tmp=(t_s).coords[i];
        for(int j=tmp.l; j<tmp.r;++j){
            // vector<int> tmp_store((t_s).k*(t_s).k);
            vector<int> tmp_store;
            int cur_m=j/m_new;
            int cur_n=j-cur_m*(m_new);
            for(int y=cur_m-addit_line_to2; y<=cur_m+addit_line_to2;++y){
                for(int x=cur_n-addit_line_to2; x<=cur_n+addit_line_to2;++x){
                    tmp_store.push_back((t_s).mass[y*m_new+x]);
                    // tmp_store.push_back(y*m_new+x);
                }
            }
            sort(tmp_store.begin(), tmp_store.end());
            // cout<<(t_s).mass[j]<<": "<<tmp_store.size()<<' ';
            // for(int z=0; z<tmp_store.size();++z){
            //     cout<<tmp_store[z]<<' ';
            // }
            // cout<<" Midle ="<<tmp_store[tmp_store.size()/2];
            // cout<<'\n';
            (t_s).output_mass[j]=tmp_store[tmp_store.size()/2];
        }
        // cur_j=tmp.r; !! исправить тут, проверка на примерах k=3 и k=5
        // cout<<" +_+ ";
    }
}

```

```

    }
    // cout<<"done\n";
}

void filter_overlay(int *mass, int *output_mass, int n, int m, int q, int k,
int threads_cnt){

    int addit_line=(k/2)*2; /*2!!!!
    int n_new=n+addit_line;
    int m_new=m+addit_line;

    int work_cell_count=n*m;
    int main_step=work_cell_count/threads_cnt;
    if(main_step<=0){
        main_step=1;
        threads_cnt=work_cell_count;
    }
    int step=main_step;
    pthread_t tid[threads_cnt];
    threads_shell parts[threads_cnt];
    int current_i=m_new*(addit_line/2)+addit_line/2;
    int thread_num=0;
    int current_n=addit_line/2;
    int incomp_step=0;
    bool wait_flag=false;
    while(thread_num<threads_cnt){
        parts[thread_num].mass=mass;
        parts[thread_num].output_mass=output_mass;
        parts[thread_num].k=k;
        parts[thread_num].m=m;
        parts[thread_num].n=n;
        if (thread_num==threads_cnt-1){
main_step=(n//n_new-addit_line/2)*m_new-current_i-(n_new-addit_line/2-1)*addit_line;
e;
            if(main_step==step ){
step=(n_new-addit_line/2)*m_new-current_i-(n_new-addit_line/2-current_n-1)*addit_line -addit_line/2;
            }
main_step=(n_new-addit_line/2)*m_new-current_i-(n_new-addit_line/2-current_n-1)
*addit_line -addit_line/2;
        }

        coord tmp;
        tmp.l=current_i;
        // cout<<'\\n'<<(current_n+1)*m_new-addit_line/2<<'\\n';

```



```

        if(current_i+step>(current_n+1)*m_new-addit_line/2){
            tmp.r=(current_n+1)*m_new-addit_line/2;
            parts[thread_numb].coords.push_back(tmp);
            step=step-(tmp.r-tmp.l);
            current_n++;
            current_i=(current_n)*m_new+addit_line/2;

        }else if(current_i+step==(current_n+1)*m_new-addit_line/2){
            tmp.r=(current_n+1)*m_new-addit_line/2;
            parts[thread_numb].coords.push_back(tmp);
            current_n++;
            current_i=(current_n)*m_new+addit_line/2;
            thread_numb++;
            step=main_step;

        } else{
            current_i+=step;
            tmp.r=current_i;
            parts[thread_numb].coords.push_back(tmp);
            thread_numb++;
            step=main_step;
        }

    }
    cout<<'\\n';
    for(int i=0; i<threads_cnt;++i){
        cout<<"i= "<<i;
        for(int j=0; j<parts[i].coords.size();++j){
            cout<<" l= "<<parts[i].coords[j].l<<' '<<" r=
"<<parts[i].coords[j].r;
        }
        cout<<'\\n';
    }

    int *tmp_mass= new int[n_new*m_new];
    for(int i=0; i<n_new*m_new;++i){
        tmp_mass[i]=mass[i];
    }
    // cout<<"mass:";
    // print_full_mass(n_new, m_new, mass);
    // cout<<"tmp_mass:";
    // print_full_mass(n_new, m_new, tmp_mass);
    // cout<<"\\noutput_mass";
    // print_full_mass(n_new, m_new, output_mass);

    for(int u=0; u<q;++u){
        for(int j=0; j<threads_cnt;++j){

```

```

        if(pthread_create(&tid[j], NULL, filt_by_cell, &parts[j])!=0){
            perror("Create thread error ");
        }

    }

    for(int i=0; i<threads_cnt;++i){
        pthread_join(tid[i], NULL);
    }
    // cout<<"tmp_mass:";
    // print_full_mass(n_new, m_new, tmp_mass);
    // cout<<"\noutput_mass";
    // print_full_mass(n_new, m_new, output_mass);

    int *tmp_pointer=tmp_mass;
    tmp_mass=output_mass;
    output_mass=tmp_pointer;
    for(int i=0;i<threads_cnt;++i){
        parts[i].mass=tmp_mass;
        parts[i].output_mass=output_mass;
    }
}

// cout<<"tmp_mass:";
// print_full_mass(n_new, m_new, tmp_mass);
// cout<<"\noutput_mass";
// print_full_mass(n_new, m_new, output_mass);

int *tmp_pointer=tmp_mass;
tmp_mass=output_mass;
output_mass=tmp_pointer;
// delete tmp_mass;

}

```

Протокол работы программы

Тестирование:

arsenii@PC-Larcha14:~/Desktop/Unikall/Git_project/OSI/lab2\$./lab_2 5

Input matrix size

n=3

0 0 0 0 0 0 0 0 0

0	0	0	0	0	0	0	0	0
0	0	245	178	116	113	156	0	0
0	0	104	119	250	229	39	0	0
0	0	46	56	14	163	121	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Correct mass:

245 178 116 113 156

104 119 250 229 39

46 56 14 163 121

output_mass:

Full mass:

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	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	0	0	0

Correct mass:

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

Time: 0.001919

arsenii@PC-Larcha14:~/Desktop/Unikall/Git_project/OSI/laba_2\$./lab_2 1

Input matrix size

n=3

m=3

Input count of filter overlay=1

Input filter matrix size k*k (it should be odd number, and k>1)

k=3

Begin mass:

Full mass:

0	0	0	0	0
0	243	12	96	0
0	118	242	234	0
0	96	47	118	0
0	0	0	0	0

Correct mass:

243 12 96

118 242 234
96 47 118

output_mass:

Full mass:

```
0  0  0  0  0
0  0 96  0  0
0 47 118 47  0
0  0 96  0  0
0  0  0  0  0
```

Correct mass:

```
0 96  0
47 118 47
0 96  0
```

Time: 0.000503177

Таблица зависимости времени выполнения от исходных данных и количества потоков:

№ Теста	Время	Размер матрицы (n*m)	Количество применений фильтра	Размер матрицы фильтра (k)	Количество потоков
1	0.0013784	5*7	3	3	1
	0.0010507 7	5*7	3	3	2
	0.0014158 6	5*7	3	3	3
2	0.0020999 2	5*7	3	5	1
	0.0018686 3	5*7	3	5	2
	0.0016873 7	5*7	3	5	3
	0.0005167 6	5*7	3	5	4
	0.0023845 1	5*7	3	5	5
3	0.0602606	100*100	3	5	1


```

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\0P\237\2\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"..., 48, 848) = 48
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\244;\374\204(337f#\315I\214\234f256\271\32"...,
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) =
0x7fd0b5200000
mmap(0x7fd0b5228000, 1658880, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7fd0b5228000
mmap(0x7fd0b53bd000, 360448, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7fd0b53bd000
mmap(0x7fd0b5415000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x214000) = 0x7fd0b5415000
mmap(0x7fd0b541b000, 52816, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fd0b541b000
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\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) =
0x7fd0b5856000
mmap(0x7fd0b5864000, 507904, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) = 0x7fd0b5864000
mmap(0x7fd0b58e0000, 372736, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8a000) = 0x7fd0b58e0000
mmap(0x7fd0b593b000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe4000) = 0x7fd0b593b000
close(3) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0) = 0x7fd0b5854000
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0) = 0x7fd0b5851000
arch_prctl(ARCH_SET_FS, 0x7fd0b5851740) = 0
set_tid_address(0x7fd0b5851a10) = 9534
set_robust_list(0x7fd0b5851a20, 24) = 0
rseq(0x7fd0b58520e0, 0x20, 0, 0x53053053) = 0
mprotect(0x7fd0b5415000, 16384, PROT_READ) = 0
mprotect(0x7fd0b593b000, 4096, PROT_READ) = 0
mprotect(0x7fd0b595b000, 4096, PROT_READ) = 0
mprotect(0x7fd0b581b000, 45056, PROT_READ) = 0
mprotect(0x560feac72000, 4096, PROT_READ) = 0
mprotect(0x7fd0b59ab000, 8192, PROT_READ) = 0

```

```

0 prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) =
munmap(0x7fd0b595d000, 80547) = 0
getrandom("\x18\xe8\x31\x17\x4c\xd8\x2a\x92", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x560fec42c000
brk(0x560fec44d000) = 0x560fec44d000
futexp(0x7fd0b582977c, FUTEX_WAKE_PRIVATE, 2147483647) = 0
= 0 newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH)
write(1, "Input matrix size\n", 18Input matrix size
) = 18
write(1, "n=", 2n=) = 2
= 0 newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH)
read(0, 5
"5\n", 1024) = 2
write(1, "\n", 1
) = 1
write(1, "m=", 2m=) = 2
read(0, 7
"7\n", 1024) = 2
write(1, "\n", 1
) = 1
write(1, "Input count of filter overlay=", 30Input count of filter overlay=) = 30
read(0, 3
"3\n", 1024) = 2
write(1, "\nInput filter matrix size k*k (i"..., 65
Input filter matrix size k*k (it should be odd number, and k>1)
) = 65
write(1, "k=", 2k=) = 2
read(0, 3
"3\n", 1024) = 2
write(1, "Begin mass:\n", 12Begin mass:
) = 12
write(1, "Full mass:\n", 11Full mass:
) = 11
write(1, " 0 0 0 0 0 0 0 0 0 "..., 37 0 0 0 0 0 0 0 0 0 0
) = 37
write(1, " 0 214 125 166 199 228 69 107 "..., 37 0 214 125 166 199 228 69 107 0
) = 37
write(1, " 0 223 43 39 27 204 160 21 "..., 37 0 223 43 39 27 204 160 21 0
) = 37
write(1, " 0 24 154 32 8 251 219 66 "..., 37 0 24 154 32 8 251 219 66 0
) = 37
write(1, " 0 76 181 153 37 192 233 56 "..., 37 0 76 181 153 37 192 233 56 0
) = 37
write(1, " 0 155 181 103 43 215 13 12 "..., 37 0 155 181 103 43 215 13 12 0

```



```

)= 37
write(1, " 0 0 0 0 0 0 0 0 0 0"..., 37 0 0 0 0 0 0 0 0 0 0
)= 37
write(1, "\n", 1
)
= 1
write(1, "Correct mass:\n", 14Correct mass:
)
= 14
write(1, "214 125 166 199 228 69 107 \n", 29214 125 166 199 228 69 107
)= 29
write(1, "223 43 39 27 204 160 21 \n", 29223 43 39 27 204 160 21
)= 29
write(1, " 24 154 32 8 251 219 66 \n", 29 24 154 32 8 251 219 66
)= 29
write(1, " 76 181 153 37 192 233 56 \n", 29 76 181 153 37 192 233 56
)= 29
write(1, "155 181 103 43 215 13 12 \n", 29155 181 103 43 215 13 12
)= 29
rt_sigaction(SIGRT_1, {sa_handler=0x7fd0b5291870, sa_mask=[],
sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO,
sa_restorer=0x7fd0b5242520}, 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) = 0x7fd0b49ff000
mprotect(0x7fd0b4a00000, 8388608, PROT_READ|PROT_WRITE) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE
AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLE
ARTID,
child_tid=0x7fd0b51ff910, parent_tid=0x7fd0b51ff910, exit_signal=0, stack=0x7fd0b49ff000,
stack_size=0x7fff00, tls=0x7fd0b51ff640} => {parent_tid=[9543]}, 88) = 9543
strace: Process 9543 attached
[pid 9534] rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
[pid 9543] rseq(0x7fd0b51ffe0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 9534] mmap(NULL, 8392704, PROT_NONE,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0 <unfinished ...>
[pid 9543] <... rseq resumed>) = 0
[pid 9534] <... mmap resumed>) = 0x7fd0b41fe000
[pid 9543] set_robust_list(0x7fd0b51ff920, 24 <unfinished ...>
[pid 9534] mprotect(0x7fd0b41ff000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>
[pid 9543] <... set_robust_list resumed>) = 0
[pid 9534] <... mprotect resumed>) = 0
[pid 9543] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9534] rt_sigprocmask(SIG_BLOCK, ~[], <unfinished ...>
[pid 9543] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9534] <... rt_sigprocmask resumed>[], 8) = 0
[pid 9534]

```

```

clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x7fd0b49fe910, parent_tid=0x7fd0b49fe910, exit_signal=0, stack=0x7fd0b41fe000,
stack_size=0x7fff00, tls=0x7fd0b49fe640} <unfinished ...>
[pid 9543] mmap(NULL, 134217728, PROT_NONE,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_NORESERVE, -1, 0strace: Process 9544 attached
) = 0x7fd0ac1fe000
[pid 9534] <... clone3 resumed> => {parent_tid=[9544]}, 88) = 9544
[pid 9544] rseq(0x7fd0b49fefe0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 9543] munmap(0x7fd0ac1fe000, 65019904 <unfinished ...>
[pid 9534] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9544] <... rseq resumed>) = 0
[pid 9534] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9543] <... munmap resumed>) = 0
[pid 9534] mmap(NULL, 8392704, PROT_NONE,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0 <unfinished ...>
[pid 9544] set_robust_list(0x7fd0b49fe920, 24 <unfinished ...>
[pid 9534] <... mmap resumed>) = 0x7fd0af7ff000
[pid 9543] munmap(0x7fd0b4000000, 2088960 <unfinished ...>
[pid 9534] mprotect(0x7fd0af800000, 8388608, PROT_READ|PROT_WRITE) = 0
[pid 9544] <... set_robust_list resumed>) = 0
[pid 9543] <... munmap resumed>) = 0
[pid 9534] rt_sigprocmask(SIG_BLOCK, ~[], <unfinished ...>
[pid 9543] mprotect(0x7fd0b0000000, 135168, PROT_READ|PROT_WRITE <unfinished ...>
[pid 9534] <... rt_sigprocmask resumed>[], 8) = 0
[pid 9544] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9543] <... mprotect resumed>) = 0
[pid 9534]

```

```

clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x7fd0affff910, parent_tid=0x7fd0affff910, exit_signal=0, stack=0x7fd0af7ff000,
stack_size=0x7fff00, tls=0x7fd0affff640}strace: Process 9545 attached
<unfinished ...>
[pid 9544] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9534] <... clone3 resumed> => {parent_tid=[9545]}, 88) = 9545
[pid 9545] rseq(0x7fd0afffffe0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 9543] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>
[pid 9534] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9545] <... rseq resumed>) = 0
[pid 9534] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9543] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9534] futex(0x7fd0b51ff910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9543,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
[pid 9545] set_robust_list(0x7fd0affff920, 24 <unfinished ...>
[pid 9544] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>

```

```

[pid 9545] <... set_robust_list resumed>) = 0
[pid 9543] madvise(0x7fd0b49ff000, 8368128, MADV_DONTNEED <unfinished ...>
[pid 9545] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9544] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9545] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9543] <... madvise resumed>) = 0
[pid 9544] madvise(0x7fd0b41fe000, 8368128, MADV_DONTNEED <unfinished ...>
[pid 9543] exit(0 <unfinished ...>
[pid 9544] <... madvise resumed>) = 0
[pid 9543] <... exit resumed>) = ?
[pid 9544] exit(0 <unfinished ...>
[pid 9545] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>
[pid 9534] <... futex resumed>) = 0
[pid 9544] <... exit resumed>) = ?
[pid 9543] +++ exited with 0 +++
[pid 9545] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9534] futex(0x7fd0b49fe910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9544,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
[pid 9545] madvise(0x7fd0af7ff000, 8368128, MADV_DONTNEED <unfinished ...>
[pid 9544] +++ exited with 0 +++
[pid 9534] <... futex resumed>) = -1 EAGAIN (Resource temporarily unavailable)
[pid 9545] <... madvise resumed>) = 0
[pid 9534] futex(0x7fd0affff910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9545,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
[pid 9545] exit(0) = ?
[pid 9534] <... futex resumed>) = 0
[pid 9545] +++ exited with 0 +++
rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE
AD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLE
ARTID, child_tid=0x7fd0affff910, parent_tid=0x7fd0affff910, exit_signal=0,
stack=0x7fd0af7ff000, stack_size=0x7fff00, tls=0x7fd0affff640}strace: Process 9546 attached
=> {parent_tid=[9546]}, 8) = 9546
[pid 9546] rseq(0x7fd0affffe0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 9534] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9546] <... rseq resumed>) = 0
[pid 9534] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9546] set_robust_list(0x7fd0affff920, 24 <unfinished ...>
[pid 9534] rt_sigprocmask(SIG_BLOCK, ~[], <unfinished ...>
[pid 9546] <... set_robust_list resumed>) = 0
[pid 9534] <... rt_sigprocmask resumed>[], 8) = 0
[pid 9546] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9534]

```

```

clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CL
ONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEAR
TID, child_tid=0x7fd0b49fe910, parent_tid=0x7fd0b49fe910, exit_signal=0, stack=0x7fd0b41fe000,

```

```

stack_size=0x7fff00, tls=0x7fd0b49fe640} <unfinished ...>
[pid 9546] <... rt_sigprocmask resumed>NULL, 8) = 0
strace: Process 9547 attached
[pid 9534] <... clone3 resumed> => {parent_tid=[9547]}, 88) = 9547
[pid 9547] rseq(0x7fd0b49fefe0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 9534] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9546] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>
[pid 9534] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9547] <... rseq resumed>      = 0
[pid 9534] rt_sigprocmask(SIG_BLOCK, ~[], <unfinished ...>
[pid 9546] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9534] <... rt_sigprocmask resumed>[], 8) = 0
[pid 9547] set_robust_list(0x7fd0b49fe920, 24 <unfinished ...>
[pid 9534]

```

```

clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTSL|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x7fd0b51ff910, parent_tid=0x7fd0b51ff910, exit_signal=0, stack=0x7fd0b49ff000,
stack_size=0x7fff00, tls=0x7fd0b51ff640} <unfinished ...>
[pid 9546] madvise(0x7fd0af7ff000, 8368128, MADV_DONTNEED <unfinished ...>
[pid 9547] <... set_robust_list resumed>) = 0
strace: Process 9548 attached
[pid 9546] <... madvise resumed>      = 0
[pid 9547] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9534] <... clone3 resumed> => {parent_tid=[9548]}, 88) = 9548
[pid 9548] rseq(0x7fd0b51ffe0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 9546] exit(0 <unfinished ...>
[pid 9534] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9548] <... rseq resumed>      = 0
[pid 9547] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9534] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9548] set_robust_list(0x7fd0b51ff920, 24 <unfinished ...>
[pid 9546] <... exit resumed>      = ?
[pid 9534] futex(0x7fd0affff910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9546,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
[pid 9548] <... set_robust_list resumed>) = 0
[pid 9534] <... futex resumed>      = -1 EAGAIN (Resource temporarily unavailable)
[pid 9546] +++ exited with 0 +++
[pid 9534] futex(0x7fd0b49fe910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9547,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
[pid 9548] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9547] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>
[pid 9548] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9547] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9547] madvise(0x7fd0b41fe000, 8368128, MADV_DONTNEED) = 0
[pid 9548] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>

```

```

[pid 9547] exit(0 <unfinished ...>
[pid 9548] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9547] <... exit resumed>)      = ?
[pid 9548] madvise(0x7fd0b49ff000, 8368128, MADV_DONTNEED <unfinished ...>
[pid 9534] <... futex resumed>)      = 0
[pid 9548] <... madvise resumed>)    = 0
[pid 9547] +++ exited with 0 +++
[pid 9534] futex(0x7fd0b51ff910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9548,
  NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
[pid 9548] exit(0)                  = ?
[pid 9534] <... futex resumed>)      = 0
[pid 9548] +++ exited with 0 +++
rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THRE
AD|CLONE_SYSVSEM|CLONE_SETTTL|CLONE_PARENT_SETTID|CLONE_CHILD_CLE
ARTID, child_tid=0x7fd0b51ff910, parent_tid=0x7fd0b51ff910, exit_signal=0,
  stack=0x7fd0b49ff000, stack_size=0x7fff00, tls=0x7fd0b51ff640}strace: Process 9549 attached
=> {parent_tid=[9549]}, 88) = 9549
[pid 9549] rseq(0x7fd0b51ffe0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 9534] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9549] <... rseq resumed>)      = 0
[pid 9534] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9549] set_robust_list(0x7fd0b51ff920, 24 <unfinished ...>
[pid 9534] rt_sigprocmask(SIG_BLOCK, ~[], <unfinished ...>
[pid 9549] <... set_robust_list resumed>) = 0
[pid 9534] <... rt_sigprocmask resumed>[], 8) = 0
[pid 9549] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9534]

```

```

clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CL
ONE_SYSVSEM|CLONE_SETTTL|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
  child_tid=0x7fd0b49fe910, parent_tid=0x7fd0b49fe910, exit_signal=0, stack=0x7fd0b41fe000,
  stack_size=0x7fff00, tls=0x7fd0b49fe640} <unfinished ...>
[pid 9549] <... rt_sigprocmask resumed>NULL, 8) = 0
strace: Process 9550 attached
[pid 9534] <... clone3 resumed>=> {parent_tid=[9550]}, 88) = 9550
[pid 9549] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>
[pid 9534] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9549] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9534] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9550] rseq(0x7fd0b49fe9e0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 9534] rt_sigprocmask(SIG_BLOCK, ~[], <unfinished ...>
[pid 9549] madvise(0x7fd0b49ff000, 8368128, MADV_DONTNEED <unfinished ...>
[pid 9534] <... rt_sigprocmask resumed>[], 8) = 0
[pid 9550] <... rseq resumed>)      = 0
[pid 9534]

```

```

clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x7fd0affff910, parent_tid=0x7fd0affff910, exit_signal=0, stack=0x7fd0af7ff000,
stack_size=0x7fff00, tls=0x7fd0affff640} <unfinished ...>
[pid 9549] <... madvise resumed>    = 0
[pid 9550] set_robust_list(0x7fd0b49fe920, 24 <unfinished ...>
[pid 9549] exit(0strace: Process 9551 attached
<unfinished ...>
[pid 9550] <... set_robust_list resumed>) = 0
[pid 9534] <... clone3 resumed> => {parent_tid=[9551]}, 88) = 9551
[pid 9551] rseq(0x7fd0affffe0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 9549] <... exit resumed>    = ?
[pid 9534] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9551] <... rseq resumed>    = 0
[pid 9550] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
[pid 9534] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9551] set_robust_list(0x7fd0affff920, 24 <unfinished ...>
[pid 9549] +++ exited with 0 +++
[pid 9550] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9534] futex(0x7fd0b49fe910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9550,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
[pid 9551] <... set_robust_list resumed>) = 0
[pid 9551] rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
[pid 9550] rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
[pid 9550] madvise(0x7fd0b41fe000, 8368128, MADV_DONTNEED <unfinished ...>
[pid 9551] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>
[pid 9550] <... madvise resumed>    = 0
[pid 9551] <... rt_sigprocmask resumed>NULL, 8) = 0
[pid 9550] exit(0 <unfinished ...>
[pid 9551] madvise(0x7fd0af7ff000, 8368128, MADV_DONTNEED <unfinished ...>
[pid 9550] <... exit resumed>    = ?
[pid 9551] <... madvise resumed>    = 0
[pid 9534] <... futex resumed>    = 0
[pid 9550] +++ exited with 0 +++
[pid 9551] exit(0 <unfinished ...>
[pid 9534] futex(0x7fd0affff910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9551,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
[pid 9551] <... exit resumed>    = ?
[pid 9534] <... futex resumed>    = 0
[pid 9551] +++ exited with 0 +++
write(1, "\n", 1
)
    = 1
write(1, "output_mass: \n", 14output_mass:
)
    = 14
write(1, "Full mass:\n", 11Full mass:

```

```

)          = 11
write(1, " 0 0 0 0 0 0 0 0 0 0 "..., 37 0 0 0 0 0 0 0 0 0 0
)= 37
write(1, " 0 0 39 39 39 43 56 0 "..., 37 0 0 39 39 39 43 56 0 0
)= 37
write(1, " 0 39 43 43 69 69 69 56 "..., 37 0 39 43 43 69 69 69 56 0
)= 37
write(1, " 0 43 43 76 103 103 69 56 "..., 37 0 43 43 76 103 103 69 56 0
)= 37
write(1, " 0 43 43 76 76 56 56 13 "..., 37 0 43 43 76 76 56 56 13 0
)= 37
write(1, " 0 0 43 43 43 37 13 0 "..., 37 0 0 43 43 43 37 13 0 0
)= 37
write(1, " 0 0 0 0 0 0 0 0 0 0 "..., 37 0 0 0 0 0 0 0 0 0 0
)= 37
write(1, "\n", 1
)
          = 1
write(1, "Correct mass:\n", 14Correct mass:
)
          = 14
write(1, " 0 39 39 39 43 56 0\n", 29 0 39 39 39 43 56 0
)= 29
write(1, " 39 43 43 69 69 69 56\n", 29 39 43 43 69 69 69 56
)= 29
write(1, " 43 43 76 103 103 69 56\n", 29 43 43 76 103 103 69 56
)= 29
write(1, " 43 43 76 76 56 56 13\n", 29 43 43 76 76 56 56 13
)= 29
write(1, " 0 43 43 43 37 13 0\n", 29 0 43 43 43 37 13 0
)= 29
write(1, "\n", 1
)
          = 1
write(1, "Time: 0.00384272\n", 17Time: 0.00384272
)
          = 17
lseek(0, -1, SEEK_CUR)          = -1 ESPIPE (Illegal seek)
exit_group(0)          = ?
+++ exited with 0 +++

```

Количество потоков - 3, количество применений фильтра - 3

```

...arsenii@PC-Larcha14:~/Desktop/Unikall/Git_project/OSI/lab_2$ strace -f ./lab_2 3
execve("./lab_2", ["/lab_2", "3"]...
...write(1, "Input count of filter overlay=", 30Input count of filter overlay=) = 30
read(0, 3  ...

```

Следовательно, количество созданных потоков - $3 \cdot 3 = 9$.

Вывод

В этой лабораторной работе я познакомился с системными вызовами и многопоточным программированием. После межпроцессорного взаимодействия многопоточное программирование казалось не такой сложной задачей. В отличие от процессов, создание потоков для ОС проще и быстрее, а также у потоки проще обмениваются данными.

Для себя я понял, что многопоточное программирование - очень мощный инструмент, но не под каждую задачу. Некоторые задачи обрабатываются быстрее на 1 потоке, нежели на нескольких, поэтому перед использованием многопоточного программирования, нужно оценить эффективность их применения для решения поставленной задачи (см Таблицу зависимости времени выполнения от исходных данных и количества потоков).

В итоге у меня получился исправно работающий код, считаю, что с поставленной задачей справился успешно.