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: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

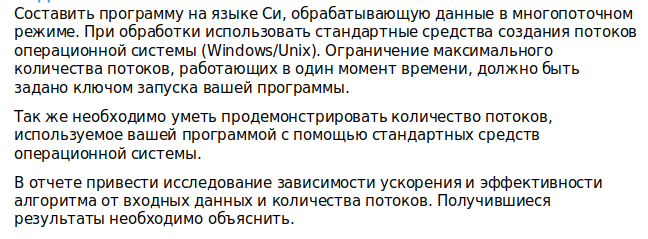
Дата: 01.12.23

Москва, 2023

**Постановка задачи**

**Вариант 11.**

**Задание:**

****

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

**Общий метод и алгоритм решения**

Использованные системные вызовы:

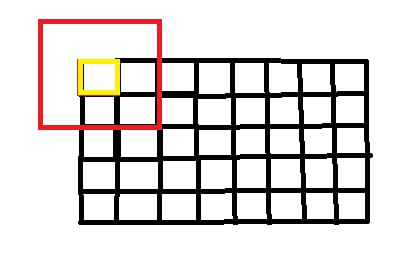
* int pthread\_create(pthread\_t \*restrict thread, const pthread\_attr\_t \*restrict attr, void\*(\*start\_routine)(void\*), void \*restrict arg) - создает поток по указателю на поток \*restrict thread, и передает ему на запуск процедуру (void\*), аргументом которой является структура \*restrict arg
* int pthread\_join(pthread\_t thread, void \*\*retval) - ждет завершение потока thread, если поток завершен до вызова функции, функция сразу завершается

**Идея:** Пользователь вводит:

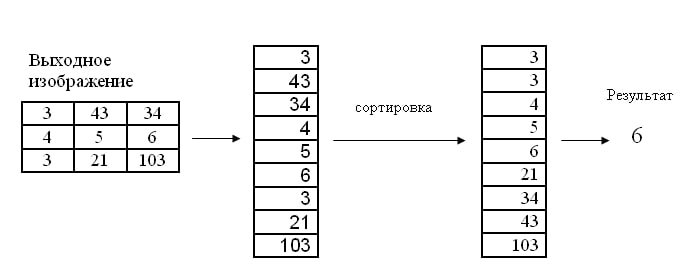
* размер исходной матрицы(n\*m),
* количество применений наложения медианного фильтра(q),
* размер квадратной матрицы медианного фильтра(k, k>1, k%2==1).

Используя функцию create\_mass() мы заполняем матрицу рандомными 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 отрезкам. 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\_numb=0;

int current\_n=addit\_line/2;

int incomp\_step=0;

bool wait\_flag=false;

while(thread\_numb<threads\_cnt){

parts[thread\_numb].mass=mass;

parts[thread\_numb].output\_mass=output\_mass;

parts[thread\_numb].k=k;

parts[thread\_numb].m=m;

parts[thread\_numb].n=n;

if (thread\_numb==threads\_cnt-1){

// main\_step=(n\_new-addit\_line/2)\*m\_new-current\_i-(n\_new-addit\_line/2-1)\*addit\_lin

e;

if(main\_step==step ){

step=(n\_new-addit\_line/2)\*m\_new-current\_i-(n\_new-addit\_line/2-current\_n-1)\*addi

t\_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/laba\_2$ ./lab\_2 5

Input matrix size

n=3

m=5

Input count of filter overlay=3

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 0

0 126 39 6 112 88 0

0 209 162 254 245 101 0

0 127 72 102 155 185 0

0 0 0 0 0 0 0

Correct mass:

126 39 6 112 88

209 162 254 245 101

127 72 102 155 185

output\_mass:

Full mass:

0 0 0 0 0 0 0

0 0 0 39 0 0 0

0 0 39 72 39 0 0

0 0 0 72 0 0 0

0 0 0 0 0 0 0

Correct mass:

0 0 39 0 0

0 39 72 39 0

0 0 72 0 0

Time: 0.00181648

arsenii@PC-Larcha14:~/Desktop/Unikall/Git\_project/OSI/laba\_2$ ./lab\_2 5

Input matrix size

n=3

m=5

Input count of filter overlay=3

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

k=5

Begin mass:

Full mass:

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 |
| 0.00105077 | 5\*7 | 3 | 3 | 2 |
| 0.00141586 | 5\*7 | 3 | 3 | 3 |
| 2 | 0.00209992 | 5\*7 | 3 | 5 | 1 |
| 0.00186863 | 5\*7 | 3 | 5 | 2 |
| 0.00168737 | 5\*7 | 3 | 5 | 3 |
| 0.00051676 | 5\*7 | 3 | 5 | 4 |
| 0.00238451 | 5\*7 | 3 | 5 | 5 |
| 3 | 0.0602606 | 100\*100 | 3 | 5 | 1 |
| 0.0127603 | 100\*100 | 3 | 5 | 10 |
| 0.0139044 | 100\*100 | 3 | 5 | 25 |

**Strace:**

arsenii@PC-Larcha14:~/Desktop/Unikall/Git\_project/OSI/laba\_2$ strace -f ./lab\_2 **3**

execve("./lab\_2", ["./lab\_2", "**3**"], 0x7ffd0ec883b0 /\* 56 vars \*/) = 0

brk(NULL) = 0x560fec42c000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffd179ce3a0) = -1 EINVAL (Invalid argument)

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1,

0) = 0x7fd0b5971000

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=80547, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 80547, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7fd0b595d000

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) =

0x7fd0b5600000

mprotect(0x7fd0b569a000, 1576960, PROT\_NONE) = 0

mmap(0x7fd0b569a000, 1118208, PROT\_READ|PROT\_EXEC,

MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9a000) = 0x7fd0b569a000

mmap(0x7fd0b57ab000, 454656, PROT\_READ,

MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ab000) = 0x7fd0b57ab000

mmap(0x7fd0b581b000, 57344, PROT\_READ|PROT\_WRITE,

MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x21a000) = 0x7fd0b581b000

mmap(0x7fd0b5829000, 10432, PROT\_READ|PROT\_WRITE,

MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fd0b5829000

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) =

0x7fd0b593d000

mmap(0x7fd0b5940000, 94208, PROT\_READ|PROT\_EXEC,

MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7fd0b5940000

mmap(0x7fd0b5957000, 16384, PROT\_READ,

MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a000) = 0x7fd0b5957000

mmap(0x7fd0b595b000, 8192, PROT\_READ|PROT\_WRITE,

MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d000) = 0x7fd0b595b000

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\244;\374\204(\337f#\315I\214\234\f\256\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\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

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

munmap(0x7fd0b595d000, 80547) = 0

getrandom("\x18\xe8\x31\x17\x4c\xd8\x2a\x92", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x560fec42c000

brk(0x560fec44d000) = 0x560fec44d000

futex(0x7fd0b582977c, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

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

write(1, "Input matrix size\n", 18Input matrix size

) = 18

write(1, "n=", 2n=) = 2

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

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 "..., 37 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 "..., 37 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(0x7fd0b51fffe0, 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|CL

ONE\_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|CL

ONE\_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]}, 88) = 9546

[pid 9546] rseq(0x7fd0afffffe0, 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\_CLEARTID,

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|CL

ONE\_SYSVSEM|CLONE\_SETTLS|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(0x7fd0b51fffe0, 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\_SETTLS|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(0x7fd0b51fffe0, 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\_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 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(0x7fd0b49fefe0, 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|CL

ONE\_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(0x7fd0afffffe0, 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 "..., 37 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 "..., 37 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/laba\_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\*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. Некоторый 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.