

Московский Авиационный Институт
(Национальный Исследовательский Университет)
Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

Лабораторная работа №3 по курсу
«Операционные системы»

Тема работы

Студент: Москвин Артём Артурович
Группа: М8О-208Б-20
Вариант: 1
Преподаватель: Миронов Евгений Сергеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2021

Содержание

1. Репозиторий
2. Постановка задачи
3. Общие сведения о программе
4. Общий метод и алгоритм решения
5. Исходный код
6. Демонстрация работы программы
7. Выводы

Репозиторий

https://github.com/Pert002/os_lab3

Постановка задачи

Составить программу на языке Си, обрабатывающую данные в многопоточном режиме. При обработки использовать стандартные средства создания потоков операционной системы (Windows/Unix). Ограничение потоков должно быть задано ключом запуска вашей программы. Так же необходимо уметь продемонстрировать количество потоков, используемое вашей программой с помощью стандартных средств операционной системы.

Вариант 12: наложить K раз фильтр, использующий матрицу свёртки, на матрицу, состоящую из вещественных чисел. Размер окна 3×3 .

Общие сведения о программе

Программа представляет из себя один файл `main.cpp`.

Общий метод и алгоритм решения

Исходная матрица разбивается на множество матриц 3×3 , которые преобразуются с помощью матрицы свёртки в вещественные числа, составляющие результирующую матрицу. Каждое такое преобразование является отдельным потоком.

Исходный код

```
#include <iostream>
#include <vector>
#include <thread>
#include <string>

using namespace std;

vector<double> res;

void bebra(double (*buffer)[3], double(*conv)[3], int current)
{
    for (int i = 0; i < 3; ++i)
    {
        for (int j = 0; j < 3; ++j)
        {
            res[current] += buffer[i][j] * conv[i][j];
        }
    }
}
```

```

    }
}

int main(int argc, char *argv[])
{
    int thread_amount;
    if (argc < 2)
    {
        cout << "Enter thread amount:\n";
        cin >> thread_amount;
    }
    else
    {
        thread_amount = stoi(argv[1]);
    }
    cout << "Thread amount is " << thread_amount << endl;
    res.resize(thread_amount);
    for (int i = 0; i < thread_amount; ++i)
    {
        res[i] = 0.0;
    }
    vector<thread> th(thread_amount);
    int current = 0;
    int k;
    cout << "Enter k:\n";
    cin >> k;
    int lines, columns;
    cout << "Enter amount of lines and columns:\n";
    do
    {
        cin >> lines >> columns;
        if ((lines - 2 * k <= 0) || (columns - 2 * k <= 0))
        {
            cout << "Error, try again:\n";
        }
    } while ((lines - 2 * k <= 0) || (columns - 2 * k <= 0));
    vector<vector<double>> orig(lines, vector<double> (columns, 0.0));
    cout << "Enter original matrix:\n";
    for (int i = 0; i < lines; ++i)
    {
        for (int j = 0; j < columns; ++j)
        {
            cin >> orig[i][j];
        }
    }
    cout << "Enter conv. 3x3 matrix:\n";
    double conv[3][3];
    for (int i = 0; i < 3; ++i)
    {

```

```

        for (int j = 0; j < 3; ++j)
        {
            cin >> conv[i][j];
        }
    }
    vector<vector<double>> result(lines, vector<double> (columns, 0.0));
    for (int t = 1; t <= k; ++t)
    {
        for (int I = 0; I < lines - 2 * t; ++I)
        {
            for (int J = 0; J < columns - 2 * t; ++J)
            {
                double buffer[3][3];
                for (int i = 0; i < 3; ++i)
                {
                    for (int j = 0; j < 3; ++j)
                    {
                        buffer[i][j] = orig[i + I][j + J];
                    }
                }
                th[current] = thread(bebra, buffer, conv, current);
                result[I][J] = res[current];
                ++current;
                if (current == thread_amount)
                {
                    current = 0;
                    for (int i = 0; i < thread_amount; ++i)
                    {
                        th[i].join();
                        res[i] = 0.0;
                    }
                }
            }
            for (int i = 0; i < current; ++i)
            {
                th[i].join();
                res[i] = 0.0;
            }
            current = 0;
        }
        for (int i = 0; i < lines - 2 * t; ++i)
        {
            for (int j = 0; j < columns - 2 * t; ++j)
            {
                orig[i][j] = result[i][j];
            }
        }
    }
    cout << "\nResult:\n";
    for (int i = 0; i < lines - 2 * k; ++i)

```

```

{
    for (int j = 0; j < columns - 2 * k; ++j)
    {
        cout << orig[i][j] << " ";
    }
    cout << endl;
}
return 0;
}

```

Демонстрация работы программы

Ввод в консоль:

```

perr@DESKTOP-L3DASJ6:/mnt/d/C C++/2 курс/OS/os_lab3/src$ strace -o res.txt ./a.out < test.txt
Enter thread amount:
Thread amount is 3
Enter k:
Enter amount of lines and columns:
Enter original matrix:
Enter conv. 3x3 matrix:

Result:
5

```

Содержимое файла res.txt:

```

execve("./a.out", ["/a.out"], 0x7fffc7d8ca90 /* 16 vars */) = 0
brk(NULL)                                = 0x7ffffcfa0000
access("/etc/ld.so.nohwcap", F_OK)       = -1 ENOENT (No such file or directory)
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
fstat(3, {st_mode=S_IFREG|0644, st_size=31208, ...}) = 0
mmap(NULL, 31208, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f72e7594000
close(3)                                  = 0
access("/etc/ld.so.nohwcap", F_OK)       = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/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\3\0>\0\1\0\0\0\220\304\10\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=1594864, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f72e7590000
mmap(NULL, 3702848, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f72e6e70000
mprotect(0x7f72e6e70000, 2097152, PROT_NONE) = 0
mmap(0x7f72e71e9000, 49152, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x179000) = 0x7f72e71e9000
mmap(0x7f72e71f5000, 12352, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f72e71f5000
close(3)                                  = 0
access("/etc/ld.so.nohwcap", F_OK)       = -1 ENOENT (No such file or directory)
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\3\0>\0\1\0\0\0\300*\0\0\0\0\0"... , 832) = 832

```

```

fstat(3, {st_mode=S_IFREG|0644, st_size=96616, ...}) = 0

mmap(NULL, 2192432, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f72e6c50000

mprotect(0x7f72e6c67000, 2093056, PROT_NONE) = 0

mmap(0x7f72e6e66000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x16000) =
0x7f72e6e66000

close(3) = 0

access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpthread.so.0", 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\000b\0\0\0\0\0"... , 832) = 832

fstat(3, {st_mode=S_IFREG|0755, st_size=144976, ...}) = 0

mmap(NULL, 2221184, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f72e6a30000

mprotect(0x7f72e6a4a000, 2093056, PROT_NONE) = 0

mmap(0x7f72e6c49000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19000) =
0x7f72e6c49000

mmap(0x7f72e6c4b000, 13440, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) =
0x7f72e6c4b000

close(3) = 0

access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)

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\0\240\35\2\0\0\0\0"... , 832) = 832

fstat(3, {st_mode=S_IFREG|0755, st_size=2030928, ...}) = 0

mmap(NULL, 4131552, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f72e6630000

mprotect(0x7f72e6817000, 2097152, PROT_NONE) = 0

mmap(0x7f72e6a17000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) =
0x7f72e6a17000

mmap(0x7f72e6a1d000, 15072, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) =
0x7f72e6a1d000

close(3) = 0

access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)

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\200\272\0\0\0\0\0"... , 832) = 832

fstat(3, {st_mode=S_IFREG|0644, st_size=1700792, ...}) = 0

mmap(NULL, 3789144, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f72e6290000

mprotect(0x7f72e642d000, 2093056, PROT_NONE) = 0

mmap(0x7f72e662c000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19c000) =
0x7f72e662c000

close(3) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f72e7580000

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f72e7570000

arch_prctl(ARCH_SET_FS, 0x7f72e7570740) = 0

mprotect(0x7f72e6a17000, 16384, PROT_READ) = 0

mprotect(0x7f72e662c000, 4096, PROT_READ) = 0

mprotect(0x7f72e6c49000, 4096, PROT_READ) = 0

```

```

mprotect(0x7f72e6e6000, 4096, PROT_READ) = 0
mprotect(0x7f72e71e9000, 40960, PROT_READ) = 0
mprotect(0x7f72e7808000, 4096, PROT_READ) = 0
mprotect(0x7f72e7429000, 4096, PROT_READ) = 0
munmap(0x7f72e7594000, 31208) = 0
set_tid_address(0x7f72e7570a10) = 37
set_robust_list(0x7f72e7570a20, 24) = 0
rt_sigaction(SIGRTMIN, {sa_handler=0x7f72e6a35cb0, sa_mask=[], sa_flags=SA_RESTORER|SA_SIGINFO,
sa_restorer=0x7f72e6a42980}, NULL, 8) = 0
rt_sigaction(SIGRT_1, {sa_handler=0x7f72e6a35d50, sa_mask=[], sa_flags=SA_RESTORER|SA_RESTART|SA_SIGINFO,
sa_restorer=0x7f72e6a42980}, NULL, 8) = 0
rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=8192*1024}) = 0
brk(NULL) = 0x7ffffcfa0000
brk(0x7ffffcfc1000) = 0x7ffffcfc1000
futex(0x7f72e71f609c, FUTEX_WAKE_PRIVATE, 2147483647) = 0
futex(0x7f72e71f60a8, FUTEX_WAKE_PRIVATE, 2147483647) = 0
fstat(1, {st_mode=S_IFCHR|0660, st_rdev=makedev(4, 1), ...}) = 0
ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0
write(1, "Enter thread amount:\n", 21) = 21
fstat(0, {st_mode=S_IFREG|0777, st_size=47, ...}) = 0
read(0, "3\r\n1\r\n3 3\r\n1 2 3 4 5 6 7 8 9\r\n0 "..., 4096) = 47
write(1, "Thread amount is 3\n", 19) = 19
write(1, "Enter k:\n", 9) = 9
write(1, "Enter amount of lines and column...", 35) = 35
write(1, "Enter original matrix:\n", 23) = 23
write(1, "Enter conv. 3x3 matrix:\n", 24) = 24
read(0, "", 4096) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7f72e5a80000
mprotect(0x7f72e5a81000, 8388608, PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7f72e627ffb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SET
TID|CLONE_CHILD_CLEARTID, parent_tidptr=0x7f72e62809d0, tls=0x7f72e6280700, child_tidptr=0x7f72e62809d0) =
38
futex(0x7f72e62809d0, FUTEX_WAIT, 38, NULL) = 0
write(1, "\nResult:\n", 9) = 9
write(1, "5 \n", 3) = 3
exit_group(0) = ?
+++ exited with 0 +++

```


Содержимое файла test.txt:

3

1

3 3

1 2 3 4 5 6 7 8 9

0 0 0 0 1 0 0 0 0

Выводы

Проделав лабораторную работу, я приобрёл практические навыки в управлении потоками в ОС и обеспечил синхронизацию между ними.