Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «МОСКОВСКИЙ ИНСТИТУТ ЭЛЕКТРОННОЙ ТЕХНИКИ»

Институт системной и программной инженерии

и информационных технологий (СПИНТех)

**Отчёт**

по дисциплине «Технологии параллельного программирования»

**Лабораторная работа №2**

**Вариант-7**

Руководитель

\_\_\_\_\_\_\_\_\_\_ Можжухина A. В.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

Студент группы ПИН-23

\_\_\_\_\_\_\_\_\_\_ Исламов Р. Р.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

*Москва*

*2023*

# 1 Листинг

// Lab2.cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.

//

/\*

Задание. Выполнить Пример для различных значений параметров NMAX и LIMIT, замеряя время выполнения, результаты занести в отчет. Выделить такие NMAX при LIMIT, при которых совпадает время выполнения многопоточной программы и однопоточной.

\*/

#include <omp.h>

#include <iostream>

#include <locale.h>

#ifndef \_OPENMP

static\_assert(false, "openmp support required");

#endif

const int NMAX = 5000;

const int LIMIT = 000;

static int\*\* matrix\_init(int n, int m)

{

int\*\* matrix;

matrix = new int\* [n];

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

matrix[i] = new int[m] {};

}

return matrix;

}

void init\_rand(int\*\* a) {

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

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

a[i][j] = rand() % 10; //запись в матрицу случайных чисел от 1 до 9

}

//std::cout << std::endl;

}

}

void show\_matrix(int\*\* a) {

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

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

std::cout << a[i][j] << " ";

}

std::cout << std::endl;

}

}

void OpenMP(int\*\* a) {

int i, j;

float sum;

#pragma omp parallel shared(a) private(i) if (NMAX>LIMIT)

{

#pragma omp for private(j,sum)

for (i = 0; i < NMAX; i++)

{

sum = 0;

for (j = 0; j < NMAX; j++) {

if (j + 1 == NMAX)

continue;

if (a[i][j] + a[i][j + 1] == 7) {

sum += 1;

//std::cout << a[i][j] << " " << a[i][j + 1] << std::endl;

}

}

//printf("Количество сёмерок в строке %d всех попарных сумм элементов равна %f\n", i, sum);

}

} /\* Завершение параллельного фрагмента \*/

}

int main()

{

setlocale(LC\_ALL, "Russian");

int\*\* a;

a = matrix\_init(NMAX, NMAX);

init\_rand(a);

//show\_matrix(a);

double openMp\_start\_time = omp\_get\_wtime();

OpenMP(a);

double openMp\_end\_time = omp\_get\_wtime();

std::cout << openMp\_end\_time - openMp\_start\_time << std::endl;

return 0;

}

# 2 Результат работы

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Тест | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| NMAX | 1000 | 1000 | 2000 | 2000 | 5000 | 5000 | 10000 | 10000 |
| NLIMIT | 700 | 1000 | 1000 | 2000 | 2000 | 5000 | 5000 | 10000 |
| Результат | 0.0092331 | 0.003107 | 0.0117607 | 0.0123922 | 0.0302575 | 0.0879143 | 0.0408087 | 0.304209 |
| OpenMP | On | Off | On | Off | On | Off | On | Off |

Из полученных данных следует, что оптимальный NLimit=2000, так как до этого не имеет смысла применять OpenMP, но после уже можно и следует его использовать