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 <iostream>

#include <ctime>

#include <omp.h>

#include <stdio.h>

using namespace std;

const int NMAX = 100;

const int LIMIT = 10;

int main()

{

setlocale(LC\_ALL, "Russian");

unsigned int start\_time1 = clock(); // начальное время

srand(time(0)); // автоматическая рандомизация

{

int i, j;

float sum;

float a[NMAX][NMAX];

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

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

a[i][j] = i + j;

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

{

#pragma omp for private(i,j,sum)

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

{

sum = 0;

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

sum += a[i][j];

printf("Сумма элементов строки %d равна %f\n", i, sum);

}

}

}

unsigned int end\_time1 = clock(); // конечное время

unsigned int search\_time1 = end\_time1 - start\_time1; // искомое время

//cout << "Задача выполнена с использованием OpenMP: " << search\_time1;

//cout << endl;

unsigned int start\_time2 = clock(); // начальное время

srand(time(0)); // автоматическая рандомизация

{

int i, j;

float sum;

float a[NMAX][NMAX];

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

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

a[i][j] = i + j;

{

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

{

sum = 0;

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

sum += a[i][j];

printf("Сумма элементов строки %d равна %f\n", i, sum);

}

}

}

unsigned int end\_time2 = clock(); // конечное время

unsigned int search\_time2 = end\_time2 - start\_time2; // искомое время

cout << "Задача выполнена с использованием OpenMP: " << search\_time1;

cout << endl;

cout << "Задача выполнена без использованием OpenMP: " << search\_time2;

cout << endl;

}

# 2 Результат работы

const int NMAX = 10;

const int LIMIT = 1;



const int NMAX = 100;

const int LIMIT = 100;

const int NMAX = 100;

const int LIMIT = 10;



const int NMAX = 100;

const int LIMIT = 100;



const int NMAX = 200;

const int LIMIT = 10;



const int NMAX = 300;

const int LIMIT = 10;



const int NMAX = 300;

const int LIMIT = 1;



Данные результаты дают ясно понять, что не всегда распараллеливание имеет смысл.