Лабораторная работа №3

Вариант №1

Задание 1. В соответствии с вариантом задания, написать на C++ программу программу, реализующую многопоточность на основе технологии OpenMP: она должна вычислять значение функции с помощью редукции. Измерять время работы программ для различных значений параметра N. Результаты занести в отчёт.

Код программы:

#include <iostream>

#include <omp.h>

#include <vector>

#include <chrono>

int main() {

using namespace std;

setlocale(LC\_ALL, "Russian");

const int N = 1000000; // Увеличим размер массивов для более наглядной демонстрации

std::vector<int> A(N, 2);

std::vector<int> B(N, 3);

int result = 1;

auto start\_time = std::chrono::high\_resolution\_clock::now();

#pragma omp parallel for reduction(\*:result)

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

if ((A[i] + B[i]) != 0) {

result \*= (A[i] + B[i]);

}

}

auto end\_time = std::chrono::high\_resolution\_clock::now();

std::chrono::duration<double> duration = end\_time - start\_time;

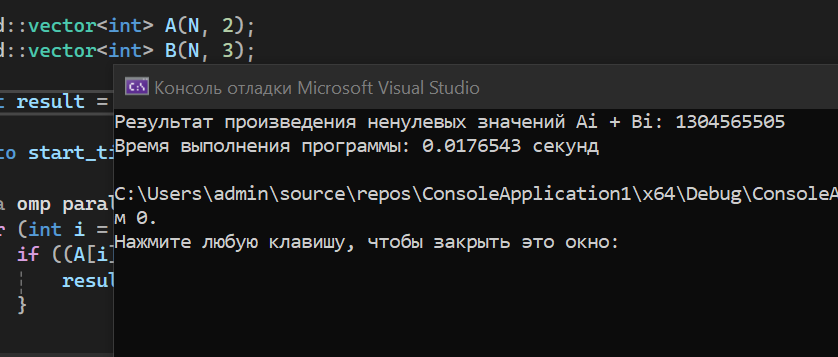
std::cout << "Результат произведения ненулевых значений Ai + Bi: " << result << std::endl;

std::cout << "Время выполнения программы: " << duration.count() << " секунд" << std::endl;

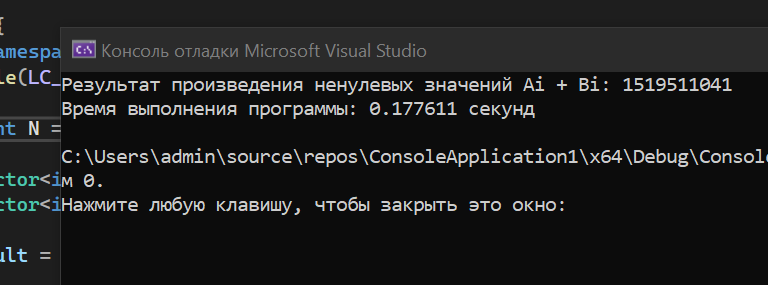
return 0;

}

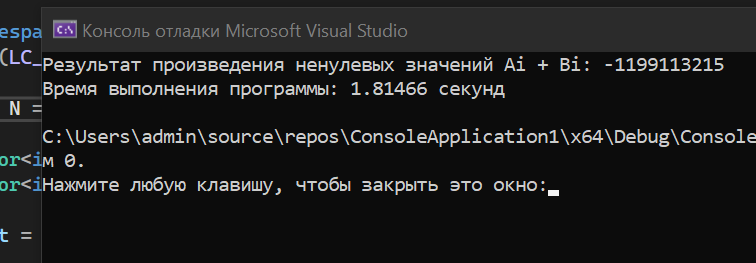
N = 1000000



N = 10000000



N = 100000000



Задание 2. Модифицировать программу, составленную по Заданию 1, убрав возможность параллельного использования вычислительных ресурсов. Измерять время работы программы для тех же значений параметров, что были использованы при выполнении Задания 1.

Код программы:

#include <iostream>

#include <vector>

#include <chrono>

int main() {

using namespace std;

setlocale(LC\_ALL, "Russian");

const int N = 1000000; // Увеличим размер массивов для более наглядной демонстрации

std::vector<int> A(N, 2);

std::vector<int> B(N, 3);

int result = 1;

auto start\_time = std::chrono::high\_resolution\_clock::now();

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

if ((A[i] + B[i]) != 0) {

result \*= (A[i] + B[i]);

}

}

auto end\_time = std::chrono::high\_resolution\_clock::now();

std::chrono::duration<double> duration = end\_time - start\_time;

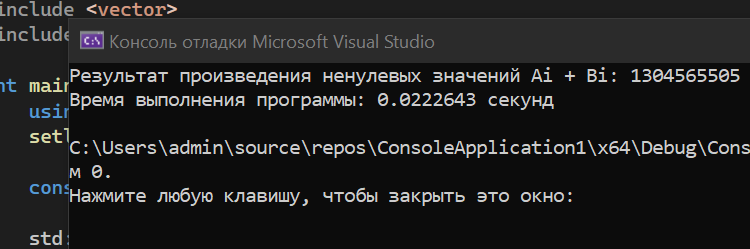
std::cout << "Результат произведения ненулевых значений Ai + Bi: " << result << std::endl;

std::cout << "Время выполнения программы: " << duration.count() << " секунд" << std::endl;

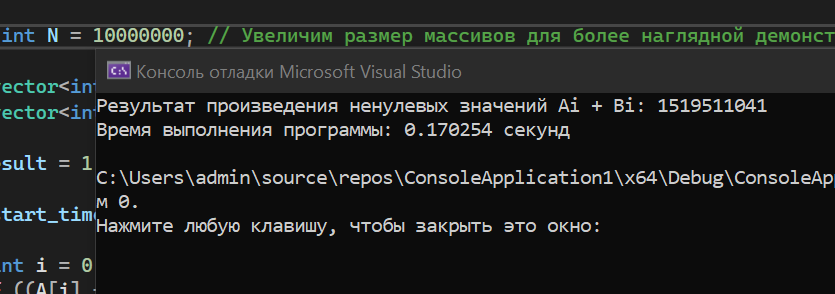
return 0;

}

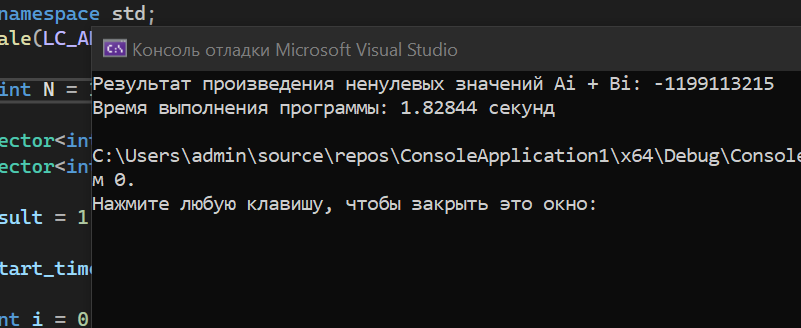
N = 1000000



N = 10000000



N = 100000000



N = 1000000000

