Министерство образования и науки Российской Федерации

Национальный исследовательский университет «МИЭТ»

Институт системной и программной инженерии и

информационных технологий

Технология программирования Open MP

Лабораторная работа №5

Вариант 1

Выполнил студент группы ПИН-31Д: Ельяшевич Н.А.

Проверилa: Можжухина А.В.

Зеленоград, 2024 г.

Задание. Модифицировать программы, составленные в Л.Р. №№3,4, распределив вычислительную нагрузку по секциям (как минимум двумя способами). Измерять время работы программы для тех же значений параметров, что были использованы при выполнении Л.Р. №3. Результаты сравнить и занести в отчёт.

Модифицированная программа из Лабораторной №3:

#include <iostream>

#include <vector>

#include <chrono>

#include <thread>

**void** **calculatePartialResults**(std::vector<**int**>& A, std::vector<**int**>& B, **int** start, **int** end, **int**& partialResult) {

**for** (**int** i = start; i < end; ++i) {

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

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

}

}

}

**int** **main**() {

**using** **namespace** std;

setlocale(LC\_ALL, "Russian");

**const** **int** N = **1000000**;

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

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

**int** result1 = **1**;

**int** result2 = **1**;

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

**thread** t1(calculatePartialResults, ref(A), ref(B), **0**, N / **2**, ref(result1));

**thread** t2(calculatePartialResults, ref(A), ref(B), N / **2**, N, ref(result2);

t1.join();

t2.join();

**int** finalResult = result1 \* result2;

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

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

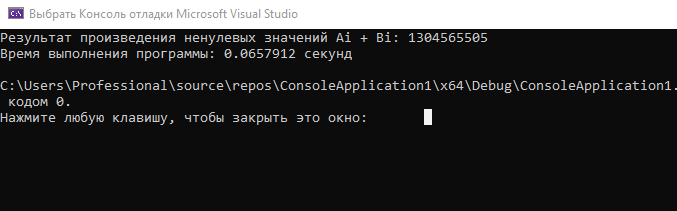
cout << "Результат произведения ненулевых значений Ai + Bi: " << finalResult << endl;

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

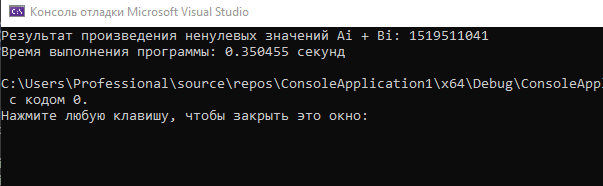
**return** **0**;

}

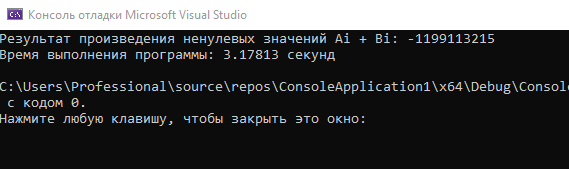
N = 1000000



N = 10000000



N = 100000000



Модифицированная программа из Лабораторной №4:

#include <iostream>

#include <omp.h>

#include <vector>

#include <chrono>

**int** **main**() {

**using** **namespace** std;

setlocale(LC\_ALL, "Russian");

**const** **int** N = **1000000**;

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) schedule(static, N / omp\_get\_num\_threads())

**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;

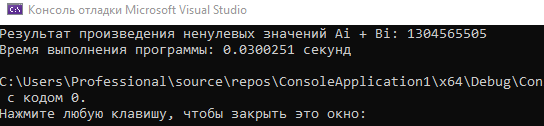
std::cout << "Результат произведения ненулевых значений Ai + Bi: " << result << std::endl;

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

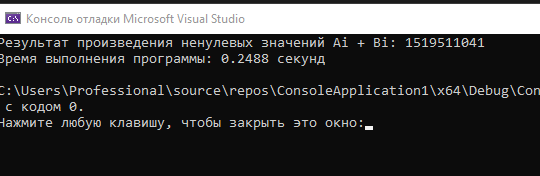
**return** **0**;

}

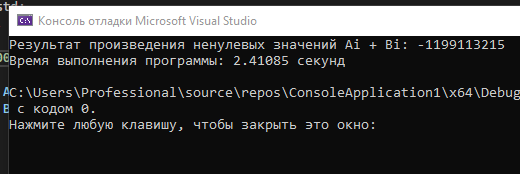
N = 1000000



N = 10000000



N = 100000000



Вывод:

Модифицированная программа из Лабораторной №4 оказалась быстрее при любом N .