Федеральное государственное бюджетное образовательное учреждение высшего образования

«Чувашский государственный университет имени И.Н. Ульянова»

Лабораторная работа №1.

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ OPENMP.

По дисциплине «Параллельное программирование»

Выполнил: ст. гр. ИВТ-42-21

Матвеев Л.Е.

Проверила: Московкина Е. А.

г. Чебоксары, 2025

**Ход работы.**

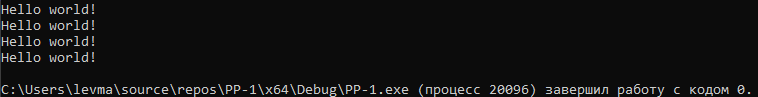
***Задание 1.***

Создан проект в visual studio с поддержкой OpenMP.

***Задание 2.***

Программа, в которой создаётся 4 потока, каждый из которых выводит “Hello World!”

Результаты:



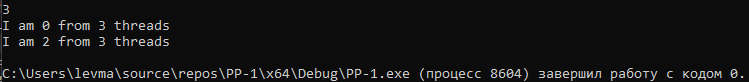
***Задание 3.***

Программа, в которой создается k потоков, и каждый поток выводит на экран свой номер и общее количество потоков в параллельной области. Выводятся только потоки с чётным номером.

**Входные данные**: k – количество потоков в параллельной области.

**Выходные данные**: k строк вида «I am thread from threads!».

**Результаты**:



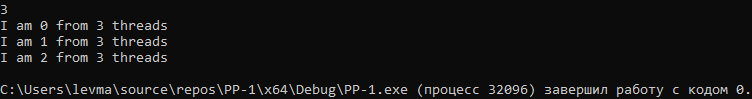
***Задание 4.***

Программа, в которой создается k потоков, и каждый поток выводит на экран свой номер через частную переменную thread\_id. Она должна быть частной чтобы у каждого потока был свой правильный номер.

**Входные данные**: целое число k – количество потоков в параллельной области.

**Выходные данные**: k строк вида «I am .».

**Результаты**:



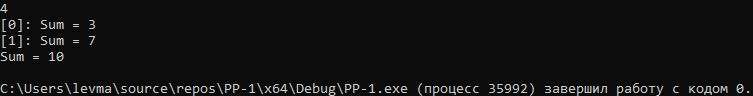
***Задание 5.***

Программа, в которой два потока параллельно вычисляют сумму чисел от 1 до N. Распределяется работа по потокам с помощью оператора if языка С. Для сложения результатов вычисления потоков используется OpenMP-параметр reduction.

**Входные данные**: целое число N – количество чисел.

**Выходные данные**: каждый поток выводит свою частичную сумму в формате «[Номер\_потока]: Sum = », один раз выводится общая сумма в формате «Sum = ».

**Результаты**:

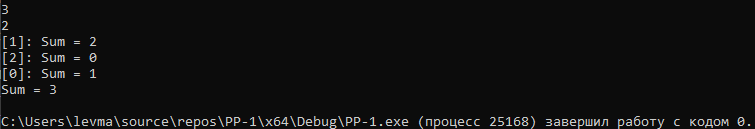


***Задание 6.***

Программа, в которой k потоков параллельно вычисляют сумму чисел от 1 до N. Работа распределяется по потокам с помощью OpenMP-директивы for.

**Входные данные**: целое число k – количество нитей, целое число N – количество чисел. **Выходные данные**: каждая нить выводит свою частичную сумму в формате «[Номер\_нити]: Sum = », один раз выводится общая сумма в формате «Sum = ».

**Результаты**:



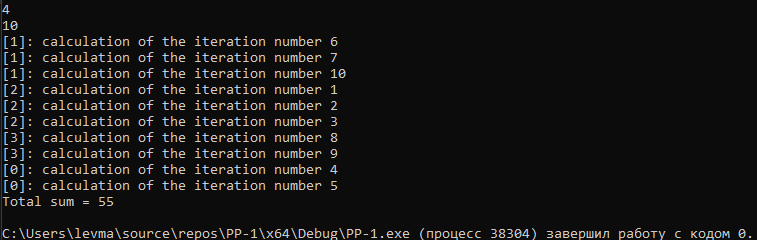
***Задание 7.***

Модифицированная программа 6 задания. Дополнительно выводится информация о номере потока и номере итерации цикла.

Таблица распределения итераций цикла по потокам в зависимости от параметра schedule:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Номер итерации** | **Значение параметра schedule** | | | | | | |
| **static** | **static,1** | **static,2** | **dynamic** | **dynamic,2** | **guided** | **guided,2** |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| **3** | 0 | 2 | 1 | 2 | 1 | 0 | 0 |
| **4** | 1 | 3 | 1 | 3 | 1 | 1 | 1 |
| **5** | 1 | 0 | 2 | 0 | 2 | 1 | 1 |
| **6** | 1 | 1 | 2 | 0 | 2 | 2 | 2 |
| **7** | 2 | 2 | 3 | 0 | 3 | 2 | 2 |
| **8** | 2 | 3 | 3 | 0 | 3 | 3 | 3 |
| **9** | 3 | 0 | 0 | 0 | 0 | 0 | 3 |
| **10** | 3 | 1 | 0 | 0 | 0 | 0 | 0 |

**Результаты**:



***Задание 8.***

Программа, которая вычисляет число π с точностью до N знаков после запятой. С помощью следующей формулы:

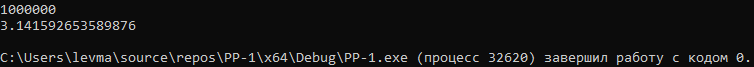


Работа распределяется с помощью OpenMP-директивы for.

**Входные данные**: одно целое число N (точность вычисления).

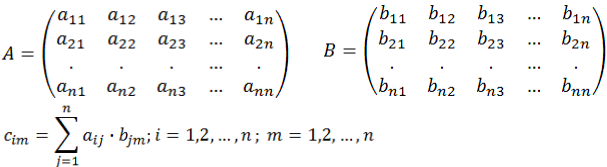
**Выходные данные**: одно вещественное число pi.

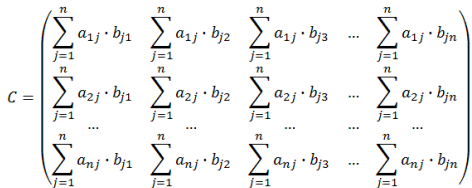
**Результаты**:



***Задание 9.***

Программа, которая вычисляет произведение двух квадратных матриц размера . Используется следующая формула:

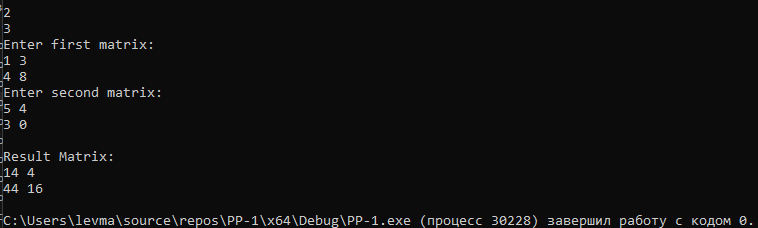




**Входные данные**: целое число n, 1 ≤ 𝑛 ≤ 10, n 2 вещественных элементов матрицы A и n 2 вещественных элементов матрицы B.

**Выходные данные**: n 2 вещественных элементов матрицы С

**Результаты**:



***Задание 10.***

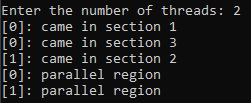
Программа, содержащая 3 параллельные секции, внутри каждой из которых выводится сообщение с информацией о секции. Вне секций внутри параллельной области выводится обычное сообщение.

**Входные данные**: k – количество потоков в параллельной области.

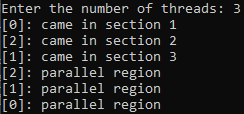
**Выходные данные**: k-строк вида «[]: came in section », k-строк вида «[]: parallel region».

**Результаты**:

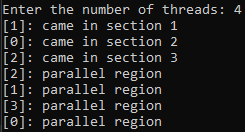
*2 потока:*



*3 потока:*

**

*4 потока:*

**

***Задание 11.***

Программа задания 6, вычисление чисел от 1 до N с использованием директивы atomic вместо reduction.

Результаты:



***Задание 12.***

Программа вычисления числа π с использованием директивы critical вместо reduction.

Результаты:

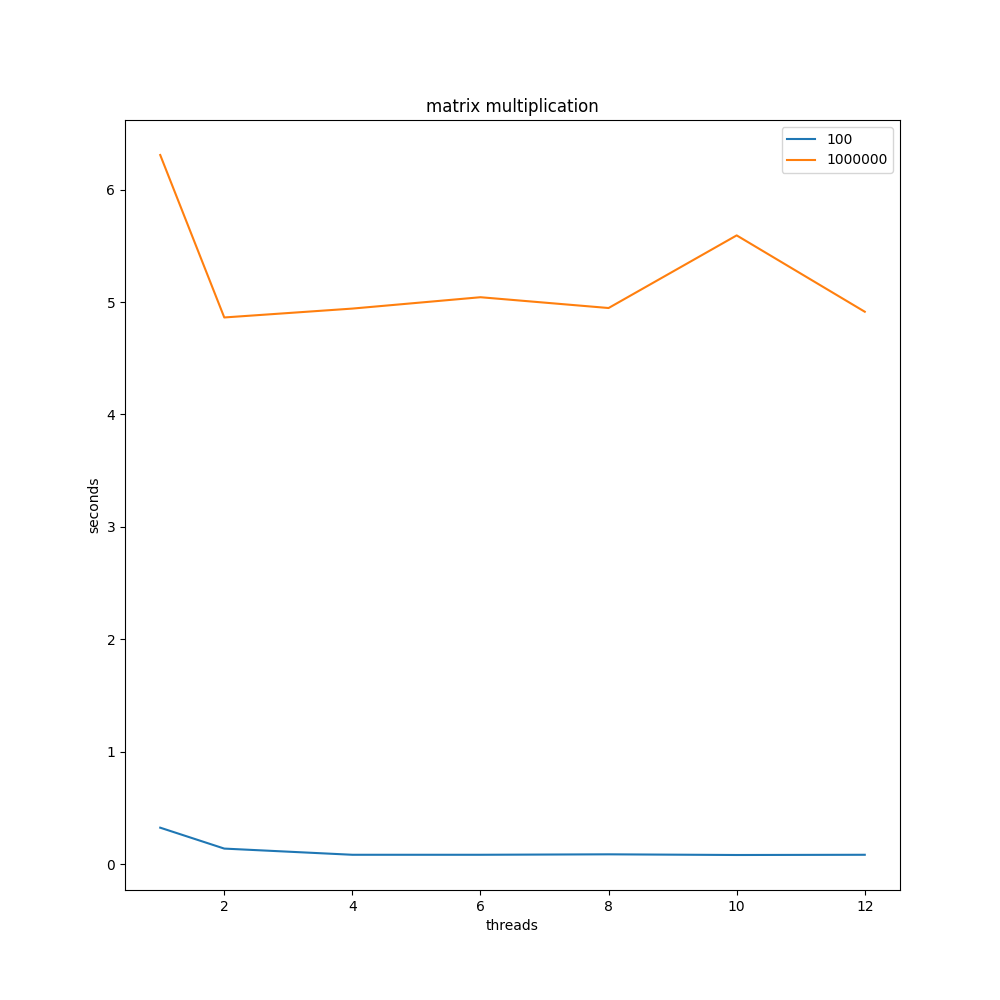


***Задание 13.***

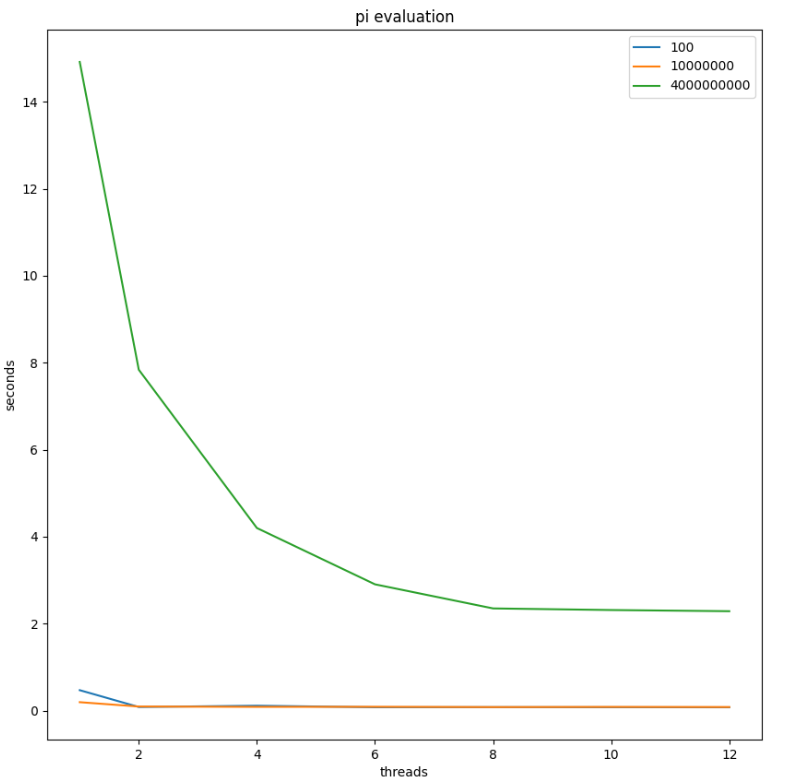
1. График масштабируемости для каждого значения параметра N по заданию 9 матрица.

при N = 1 000 000, максимальное ускорение достигается при 2 потоках.

При N = 100, максимальное ускорение достигается при 4 - 12 потоках



1. Число π.



1. Число π с critical.

