Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**«Самарский НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**имени академика С.П. Королева»**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ

**Отчёт по лабораторной работе №2**

ПЕРЕМНОЖЕНИЕ ДВУХ МАТРИЦ ПО ТЕХНОЛОГИИ OPENMP

Жирнов Владислав Евгеньевич

Группа 6313-100503D

Самара 2025

# **Цель работы**

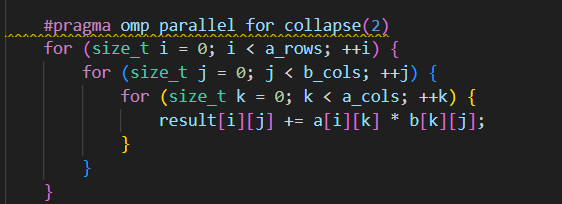
Модифицировать программу из л/р №1 для параллельной работы по технологии OpenMP.

# **2 Программа перемножения двух матриц на языке C++**

Программа main.cpp, которая представлена на языке C++, создаёт и перемножает квадратные матрицы по технологии OpenMP, а также программа вычисляет время выполнения.

В рамках выполнения лабораторной работы №2 была реализована параллельная версия алгоритма умножения матриц с использованием технологии OpenMP. Основной акцент был сделан на оптимизацию вычислительного процесса за счет распараллеливания вложенных циклов.

Для достижения параллельного выполнения операций была применена директива #pragma omp parallel for collapse(2), которая позволяет эффективно распределить итерации двух вложенных циклов между доступными потоками процессора.



Создается 10 пар квадратных матриц с размерами от 10x10 до 100x100 с шагом 10. Каждая пара матриц одного размера сохраняется в соответствующих директориях: Matrix\_1 содержит левые множители, а Matrix\_2 - правые множители. Таким образом, после завершения генерации в каждой из этих папок оказывается по 10 матриц соответствующих размеров - от 10x10 до 100x100.

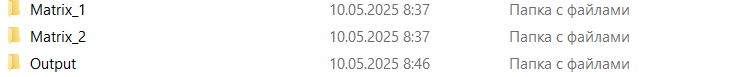


Рисунок 1 – основные каталоги, используемые при работе программы

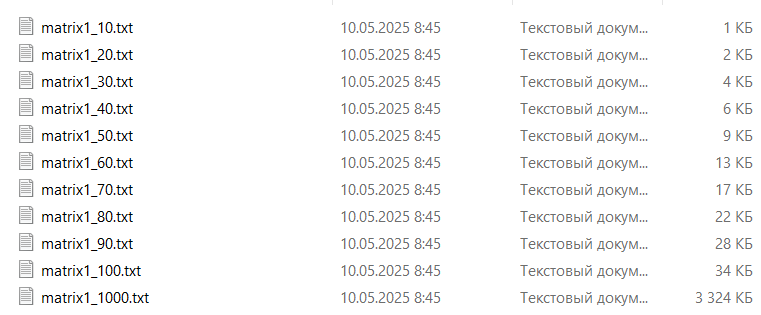
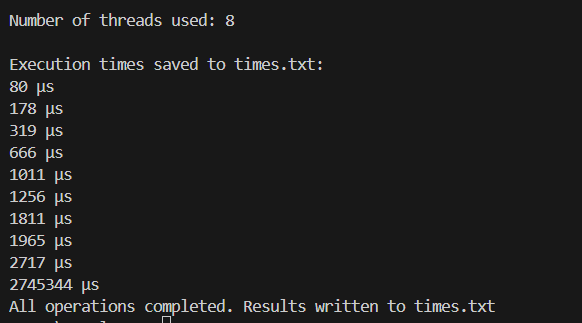


Рисунок 2 – внутренние каталоги каталога Matrix\_1.

Исходные матрицы считываются из соответствующих файлов в директориях Matrix\_1 и Matrix\_2, после чего производится их поэлементное перемножение. Результирующая матрица сохраняется в папке Output.

Особенностью реализации стало применение механизма параллельных вычислений через OpenMP, что позволило значительно сократить время выполнения операций. Временные характеристики каждого умножения фиксируются и записываются в файл times\_openmp.txt для последующего анализа производительности.

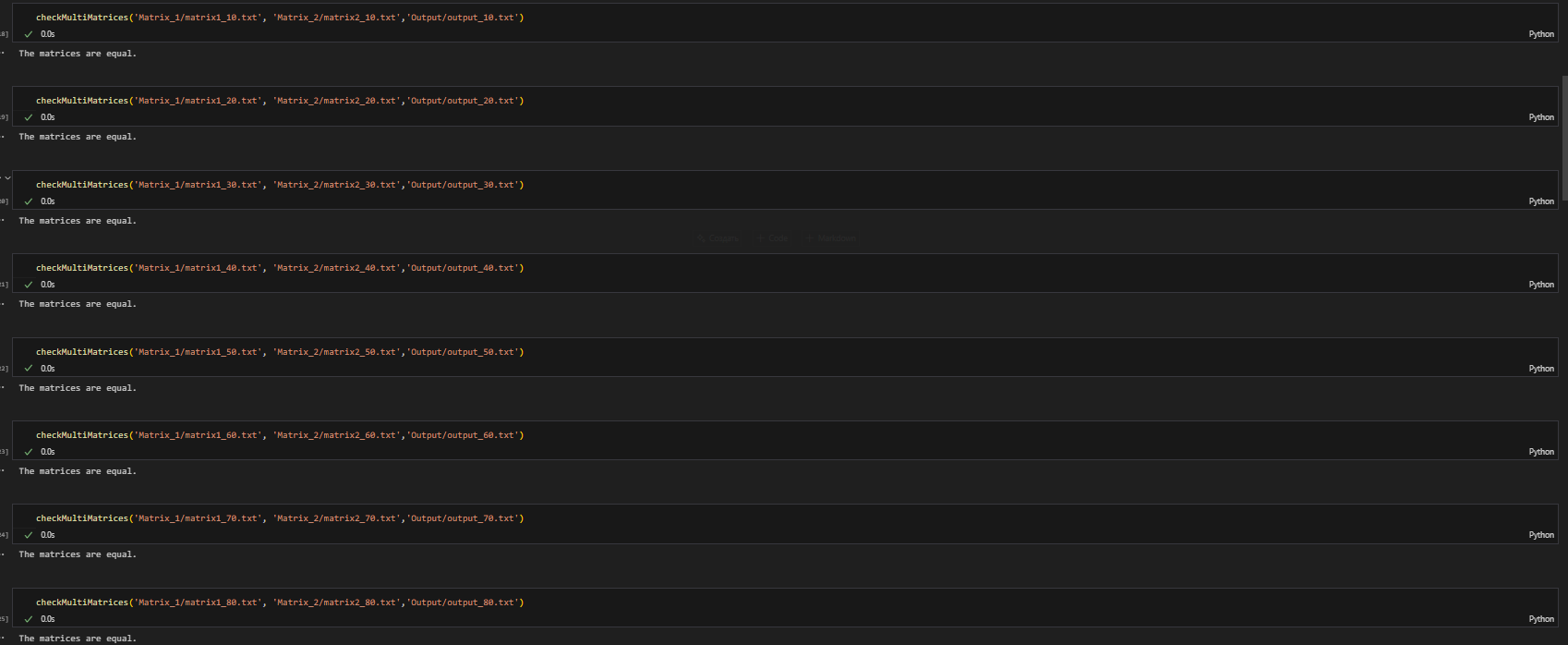
Измерение времени выполнения операций для матриц различных размерностей с помощью OpenMP:  


Для верификации результатов работы C++ программы был разработан Python-скрипт Check\_mul.ipynb. В этом скрипте используется функция dot() из библиотеки NumPy, которая выполняет точное матричное умножение и служит эталоном для сравнения.

# **3 Программа проверки корректности перемножения двух матриц на языке Python**

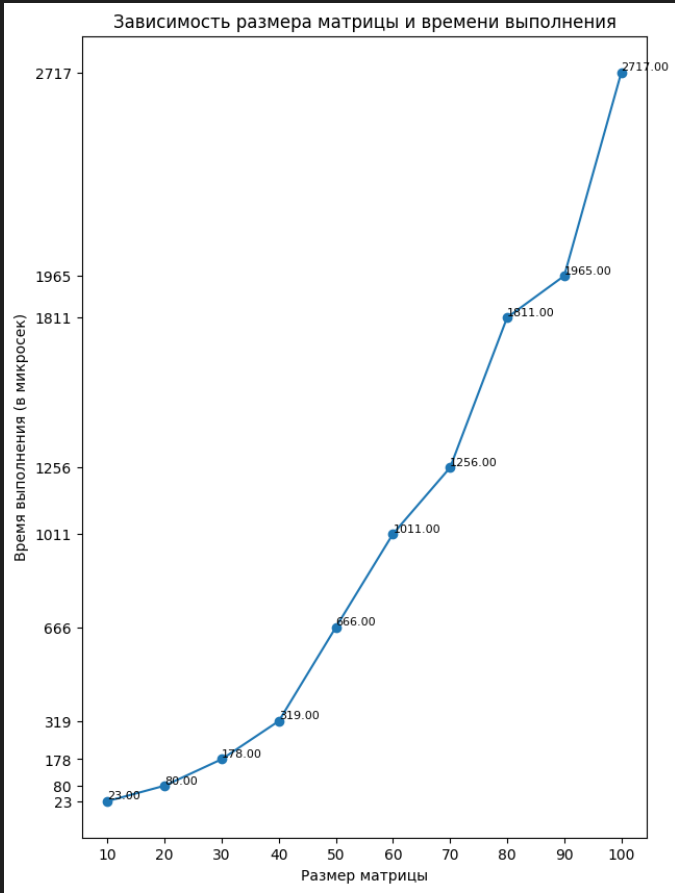
Для верификации результатов работы C++ программы был разработан Python-скрипт Check\_mul.ipynb. В этом скрипте используется функция dot() из библиотеки NumPy, которая выполняет точное матричное умножение и служит эталоном для сравнения

Результат проверки корректности перемножения двух матриц оказался положительным, следовательно ошибок в алгоритме перемножения матриц допущено не было.

Результат работы программы корректности перемножения двух матриц:  


# **Программа для измерения время вычисления произведения двух матриц, на языке Python**

Разработанный Python-скрипт анализирует временные характеристики матричных операций, записанные в ходе выполнения C++ программы. После обработки данных строится графическая зависимость, наглядно демонстрирующая как изменяется среднее время вычислений при увеличении размерности матриц.



Вывод: Результаты показывают, что для работы с крупными матрицами необходимо либо дальнейшее оптимизирование алгоритма, либо использование более мощного аппаратного обеспечения, так как временные затраты растут нелинейно при увеличении размерности.