Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет «Высшая школа экономики»

Кафедра «Компьютерная безопасность»

**ОТЧЁТ**

**К ЛАБОРАТОРНОЙ РАБОТЕ №1**

**по дисциплине**

**«Параллельные вычисления»**

Работу выполнила студентка

группы СКБ211:

Кошелева К.А.

Москва 2025

Оглавление

**Элементы оглавления не найдены.**

# Постановка задачи

Сравнить время работы следующих вариантов программы, выполняющей транспонирование плотных матриц больших размерностей:

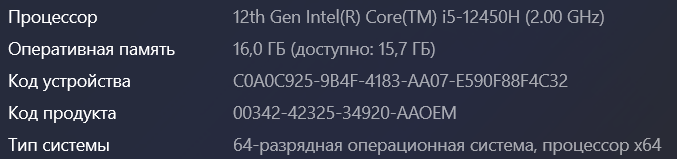
- исходная программа, без оптимизации

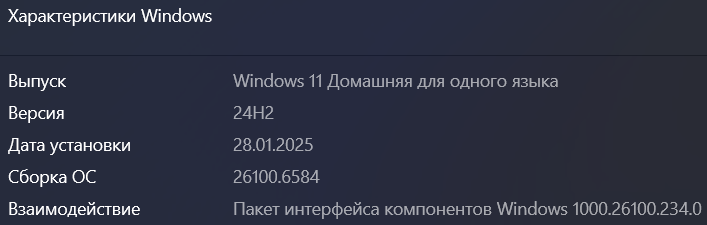
- исходная программа, оптимизированная только с помощью ключей компилятора

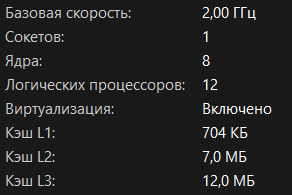
- программа, векторизованная полуавтоматически (с помощью директив и ключей компилятора и

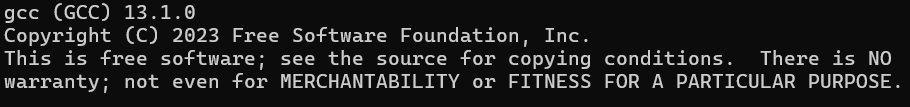
незначительной правки кода)

# Описание используемой вычислительной системы и компилятора





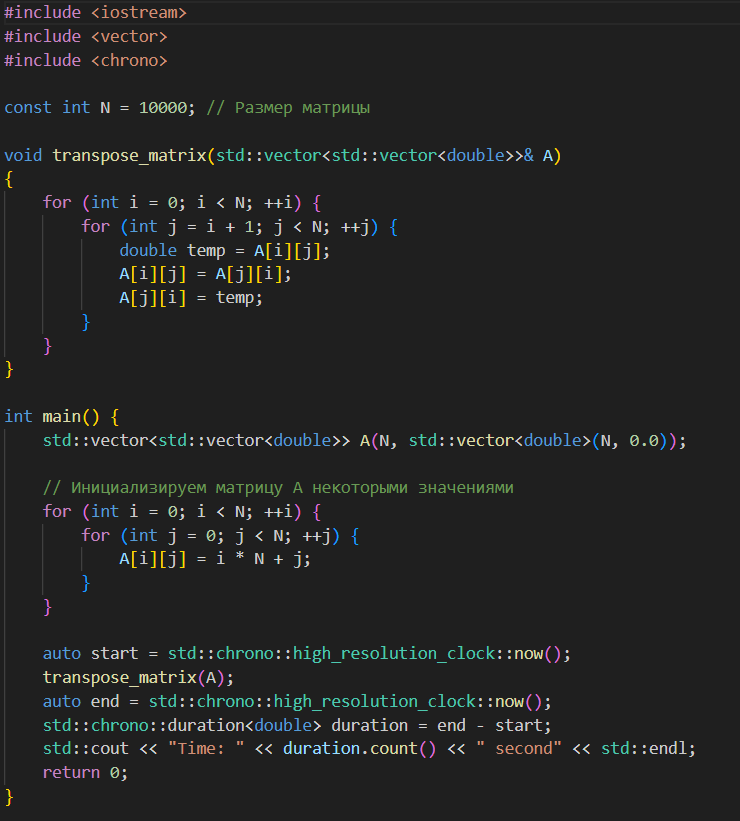


Компилятор: 

Среда разработки: VS Code v1.101.0

# Листинг программы на C++

## Без оптимизации

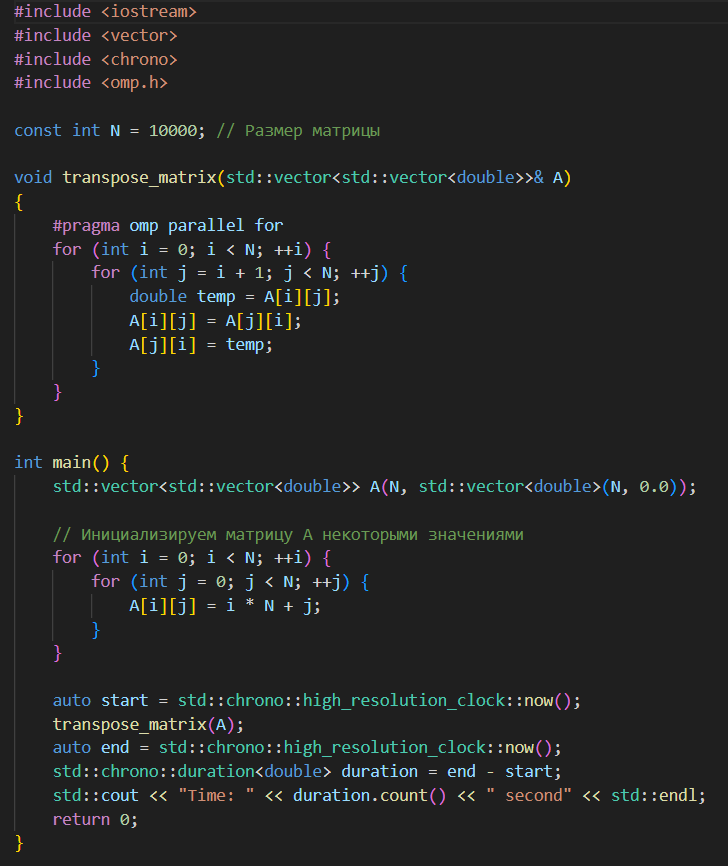


Компиляция: g++ -o trans\_1 lab1.cpp

## Оптимизация только с помощью ключей компилятора

Компиляция: g++ -O3 -march=native -o trans\_2 lab1.cpp

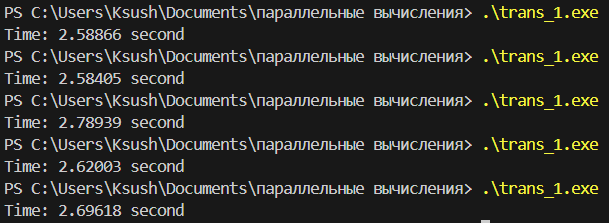
## Полуавтоматическая векторизация



Компиляция: g++ -O3 -fopenmp -march=native -o trans\_3 lab1\_omp.cpp

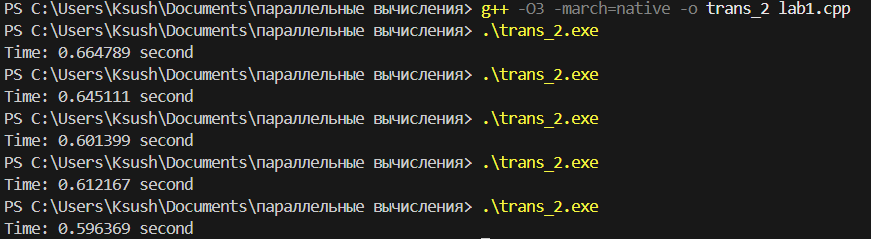
# Результаты выполнения задания

## Без оптимизации



Среднее время выполнения: 2.655662 секунды

## Оптимизация только с помощью ключей компилятора



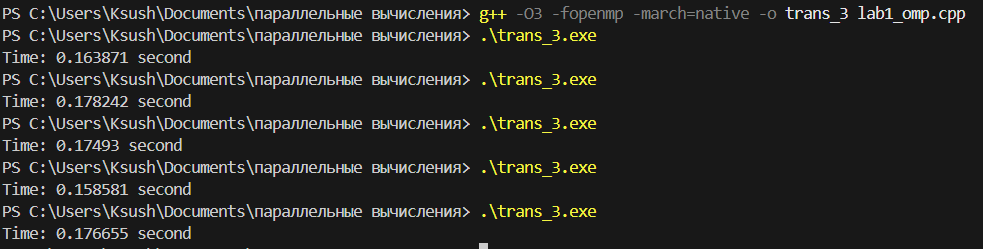
Среднее время выполнения: 0.623967 секунд

Используемые ключи компилятора:

- O3 (задействует агрессивные методы оптимизации компилятора)

-march=native (оптимизирует код для текущей архитектуры процессора)

## Полуавтоматическая векторизация



Среднее время выполнения: 0.1704558 секунд

Полуавтоматическая векторизация с использованием директивы #pragma omp parallel for из OpenMP для ускорения выполнения программы путем распараллеливания циклов.

# Таблица результатов последовательных оптимизаций

|  |  |  |
| --- | --- | --- |
| **Вид оптимизации** | **Среднее время выполнения (c)** | **Ускорение (в %)** |
| Без оптимизации | 2.655662 | ----- |
| C помощью ключей компилятора | 0.623967 | 0.77 |
| Полуавтоматическая векторизация | 0.1704558 | 0.94 |

# Листинг на ассемблере фрагмента кода

# Вывод по результатам выполнения задания

Оптимизация программы транспонирования плотных матриц с использованием различных подходов позволила добиться существенного сокращения времени выполнения программы. Использование ключей компилятора обеспечило ускорение более чем в 4 раза по сравнению с неоптимизированной версией. Применение директив OpenMP для распараллеливания вычислений позволило ещё больше снизить время выполнения — почти в 15 раз по сравнению с исходной версией. Таким образом, последовательное применение методов оптимизации показало значительный рост производительности при работе с вычислительно затратными операциями над большими объёмами данных.