Міністерство освіти і науки України

Львівський національний університет Ім. І. Франка

Факультет прикладної математики та інформатики

Кафедра дискретного аналізу та інтелектуальних систем

**Паралельні та розподілені обчислення**

**Лабораторна робота №8**

**Програмно-апаратна архітектура паралельних обчислень CUDA**

Роботу виконала:

Студентка ПМІ-33

Багінська Маргарита

Прийняв:

доц. Пасічник Т.В.

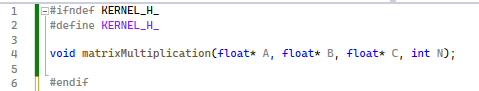
Львів 2023

**Тема**: Програмно-апаратна архітектура паралельних обчислень CUDA

**Мета**: Реалізувати алгоритм множення матриць використовуючи програмно-апаратну архітектуру паралельних обчислень CUDA.

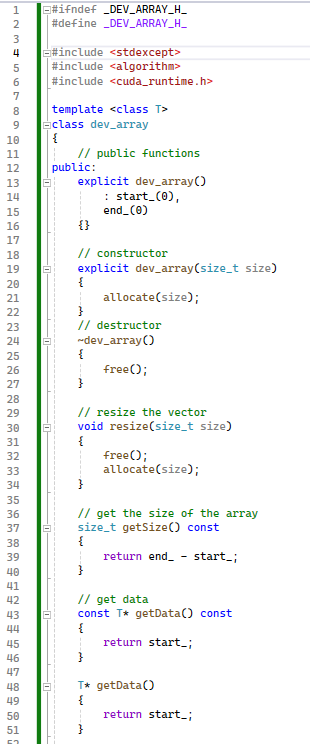
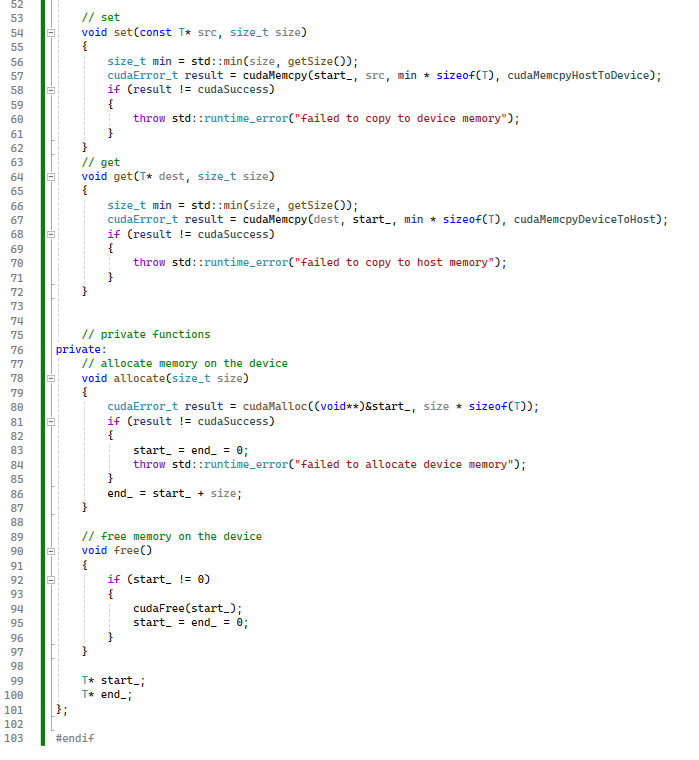
**Виконання**

* Файл **kernel.h**



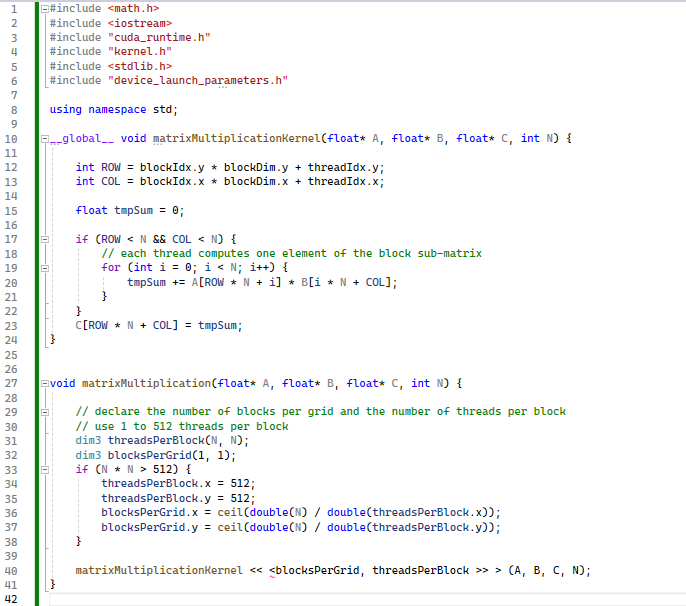
Файл "kernel.h" містить оголошення функції matrixMultiplication, яка призначена для виконання множення двох матриць, використовуючи технологію CUDA.

* Файл **dev\_array.h**

Файл "dev\_array.h" містить клас шаблонів dev\_array, який є обгорткою над пам'яттю пристрою (пам'ять GPU) для простої алокації пам'яті, зміни розміру, зчитування та запису. Цей клас допомагає забезпечити легкий та безпечний спосіб керування пам'яттю пристроїв.

* Файл kernel.cu



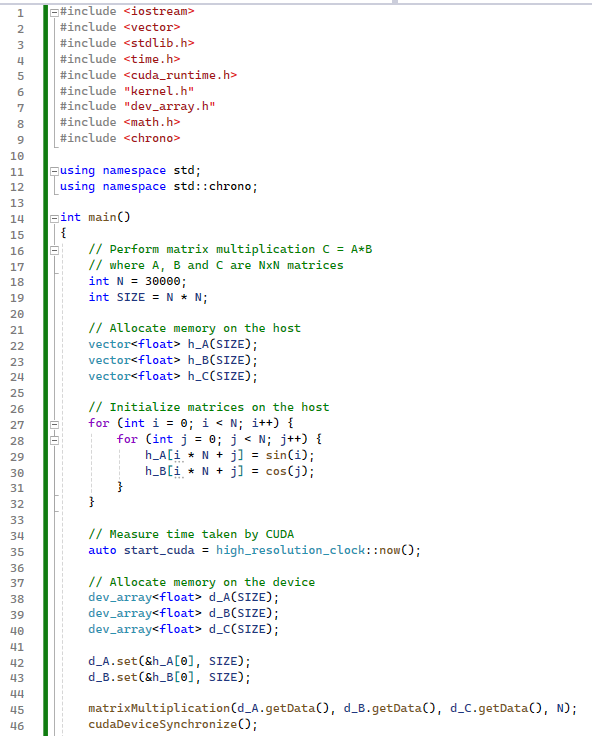
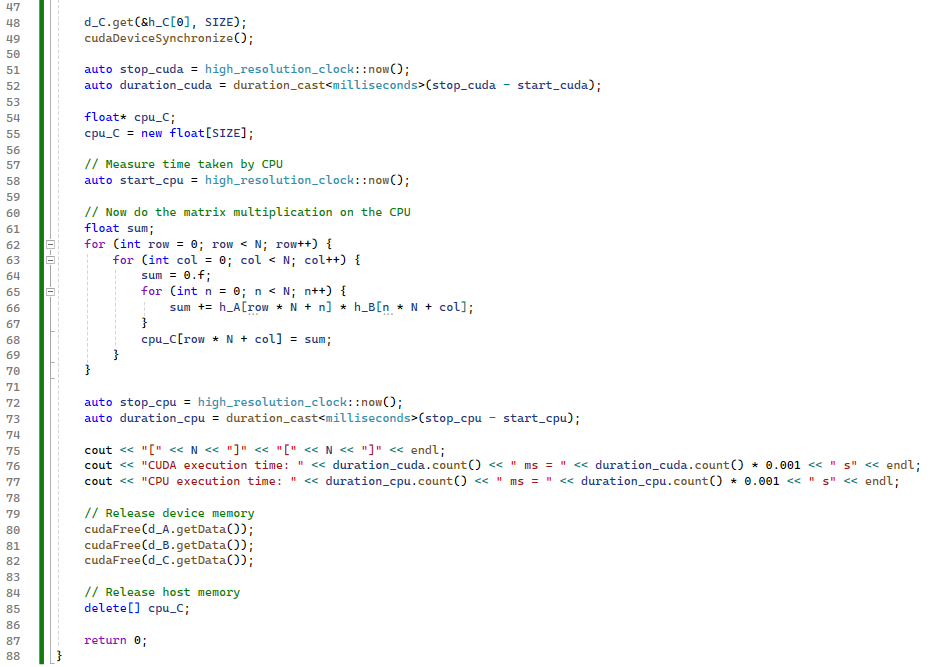
Файл "kernel.cu" містить визначення функції **matrixMultiplication**, яка використовує ядро CUDA **matrixMultiplicationKernel** для перемноження двох матриць на GPU. Основні частини коду:

**\_\_global\_\_ void matrixMultiplicationKernel(float\* A, float\* B, float\* C, int N):**

Це ядро CUDA, яке виконує матричне множення на рівні елементів. Кожен потік відповідає за обчислення одного елемента результуючої матриці C. Ітерація через рядки і стовпці вхідних матриць A та B відбувається всередині цього ядра. Ядро виконується у паралель для кожної комбінації blockIdx та threadIdx, середнє значення цих значень перемножується з розмірами блоку, щоб отримати індекс рядка (ROW) та стовпця (COL).

Функція **void matrixMultiplication(float\* A, float\* B, float\* C, int N):**

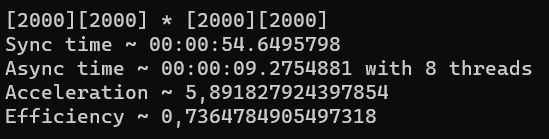
керівництво CUDA, яке приймає вказівники на дані матриць A, B та C та розмір матриці N. Функція налаштовує рядки, стовпці блоків на сітці та потоки на блок для керування виконанням ядра. Він використовує блоки та потоки для паралельного обчислення елементів результуючої матриці C.

* Файл matrixmul.cu

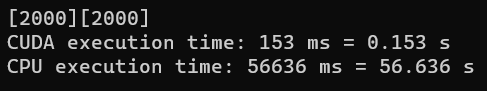
В цьому файлі реалізовано порівняння часу виконання матричного множення на CPU та GPU (CUDA). Використовуючи структуру chrono, код обчислює час виконання операцій на CPU та GPU.

**Результати**

Згадаємо результати третьої лабораторної роботи, де ми множили матриці синхронно на паралельно на CPU.

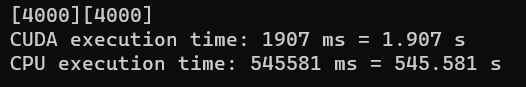


Тепер результати CUDA:



Під CPU execution time мається на увазі час роботи послідовного алгоритму, але навіть порівняно з часом паралельного алгоритму на результати CUDA вражають.

Збільшимо розмірність.



Різниця насправді колосальна. Те, що синхронно рахувалось CPU 9 хвилин, на CUDA ядрах виконалось за менш ніж 2 секунди.

**Висновки**: У результаті виконання лабораторної роботи було реалізовано алгоритм множення матриць використовуючи програмно-апаратну архітектуру паралельних обчислень CUDA.