# Звіт до лабораторної 6 (Critical Section. Race Condition. Atomic Operations.)

Студента групи ТТП-32

Остренка Олександра

## 1. Умова лабораторної:

## Изображение выглядит как текст, Шрифт, снимок экрана, бумага Автоматически созданное описание

## 2. Реалізація:

### 2.1. Розпаралелення обчислення добутку матриць.

Ідея – розбити вздовж рядків матрицю А на стільки частин, скільки визначено потоків, відповідно кожен потік буде обробляти якусь частину матриці А.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Після створення, у циклі, запускаємо усі потоки.

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Функція, яка буде виконуватись відповідними потоками, отримає два integer – початковий рядок матриці А, з якого треба почати обчислення, та кінцевий.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Послідовно, циклічно, обчислюємо кожну координату новоствореної матриці. Для виводу використовуємо mutex, щоб «перехопити» консоль на момент виводу обчисленої координати.

### 2.2. Тестування множення матриць.

Демонстрація паралелізму:

Изображение выглядит как текст, снимок экрана, черно-белый, монохромный

Автоматически созданное описаниеНескладно побачити, що різні потоки перехоплюють консоль і виводять обчислені результати, і робиться це паралельно.

Щоб продемонструвати швидкодію обчислення в залежності від кількості потоків, виконаємо 5ть обчислень для 1го, 2х, 4х, 8ми та 16ти потоків (кількість ядер процесору на машині - 4).

* Один потік:  
    
    
    
    
  
* Два потоки:  
    
    
    
    
  
* Чотири потоки:  
    
    
    
    
  
* Вісім потоків:  
    
    
    
    
  
* Шістнадцять потоків:  
    
    
    
    
  

Легко помітити, що максимальна швидкодія зупиняється на восьми потоках (intel вказує, що процесор має 4ри реальних та 4ри віртуальних), відповідно, така кількість потоків являється оптимальною. Як каже Chat GPT: “Більше потоків, ніж ядер процесору, може призвести до погіршення продуктивності. Це через те, що процесору потрібно буде витрачати час на перемикання контексту між потоками, що може вимагати додаткових ресурсів і впливати на швидкість виконання.”

### 2.3. Збільшення спільної комірки пам’яті.