МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра ШІ

Звіт

Щодо виконання лабораторної роботи №1

**«**Створення багатопотокових програмних застосунків під платформу .NET**»**

із дисципліни «Паралельне програмування під .NET Core»

Виконав:

студент групи IТШI-20-2

Андрієнко М.С

Прийняв:

Бібічков І. Є.

Харків - 2023

**Мета роботи:** Вивчення особливостей створення багатопотокових програмних застосунків під платформу .NET. Вивчення особливостей розпаралелювання математичних алгоритмів.

# Порядок виконання роботи:

1. Узгодити з викладачем завдання.
2. Реалізувати запропоноване завдання у вигляді однопотокового застосунку.
3. Здійснити розпаралелювання алгоритму для його подальшої програмної реалізації у вигляді багатопотокового застосунку.
4. Реалізувати розроблений алгоритм як багатопотоковий застосунок. Для реалізації необхідно вибрати .NET технологію створення багатопотокових застосунків, що передбачає явне створення потоків. При цьому в застосунку має бути можливість завдання кількості потоків, що породжуються. Кількість потоків, що породжуються, може залежати, наприклад, від числа ядер процесору, але не повинна прямо залежати від розмірності завдання.
5. Рішення обох варіантів реалізації алгоритму мають бути об'єктно-орієнтованими.
6. На тестових прикладах різної розмірності і з різною кількістю потоків, що породжуються, оцінити величину прискорення, що дається багатопотоковою реалізацією алгоритму. Прискорення необхідно розраховувати як відношення часу виконання однопотокової версії алгоритму до часу виконання багатопотокової його версії. Виявлені закономірності відобразити у висновках.
7. Завдання, не відзначені зірочками, оцінюються з максимуму в 18 балів. Позначені однією зірочкою – з максимуму в 19 балів. Позначені двома та трьома зірочками – з максимуму в 20 балів. За вибір завдання, відзначеного трьома зірочками, надається компенсаційний бал. Вітається пропонування власних оригінальних варіантів завдання.
8. Підготувати звіт з лабораторної роботи.

**Індивідуальне завдання:**8\*. Циклічний зсув елементів вектору ліворуч.

**Виконання роботи:**

**Опис однопотокового варіанту алгоритму:**

Цей метод виконує циклічний зсув ліворуч вектора array на вказану кількість позицій.

**Параметри методу:**

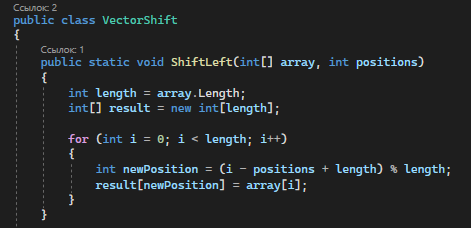
* array: Масив цілих чисел, що підлягає циклічному зрушенню.
* positions: Кількість позицій, на які потрібно перемістити елементи вліво.

**Локальні змінні:**

* length: Довжина масиву array.
* result: Локальний масив, у якому тимчасово зберігаються результати циклічного зсуву.

Циклічне зрушення вліво:

* Для кожного елемента array є нова позиція (newPosition), використовуючи формулу (i - positions + length) % length. Ця формула забезпечує циклічний зсув елементів масиву.
* Значення елемента array[i] зберігається у відповідній позиції result[newPosition].



**Короткий опис вибраної .NET технології створення багатопотокових**

**застосунків.**

**Parallel.For** – це частина бібліотеки .NET, яка надає зручний спосіб розпаралелювання виконання циклу. Вона дозволяє виконувати ітерації циклу паралельно, розподіляючи їх між кількома потоками.

**Опис багатопотокового варіанту алгоритму.**

Метод ShiftLeftParallel також виконує циклічний зсув елементів масиву array вліво на вказану кількість позицій позицій, але використовує паралельне виконання для прискорення процесу. Ось опис методу:

**Параметри методу:**

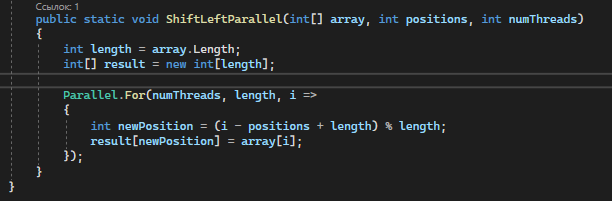
* array: Масив цілих чисел, що підлягає циклічному зрушенню.
* positions: Кількість позицій, на які потрібно перемістити елементи вліво.
* numThreads: Кількість потоків, які будуть використовуватися для паралельного виконання.

**Локальні змінні:**

* length: Довжина масиву array.
* result: Локальний масив, у якому тимчасово зберігаються результати циклічного зсуву.

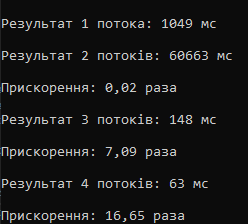
**Паралельний циклічний зсув вліво:**

* Використовується Parallel.For для паралельного виконання циклу.
* Кожен потік виконує ітерацію циклу, починаючи з numThreads і до length, включно.
* Для кожної ітерації є нова позиція (newPosition), використовуючи формулу (i - positions + length) % length.
* Значення елемента array[i] зберігається у відповідній позиції result[newPosition].

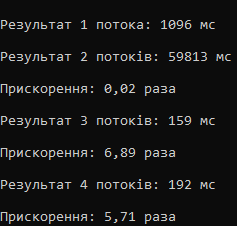


**Скріни результатів роботи кожної програми.**

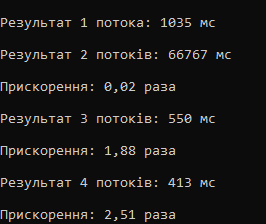
Вектор довжиною 10:



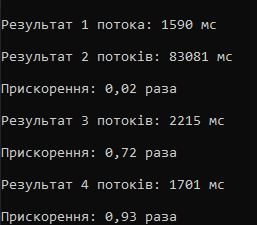
Вектор довжиною 100:



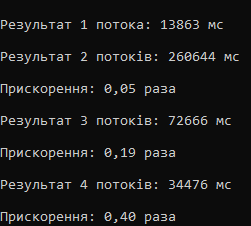
Вектор довжиною 1000:



Вектор довжиною 10000:



Вектор довжиною 10000:



Результати показують, що багатопотокова реалізація не дає очікуваного прискорення. Прискорення розраховується як відношення часу виконання однопоточного і багатопоточного варіантів. У нашому випадку результати свідчать, що використання кількох потоків навіть повільніше.

**Це може бути обумовлено кількома причинами:**

* Накладні витрати від паралелізації: впровадження паралелізму вносить додаткові витрати через синхронізацію та координацію потоків. Для невеликих операцій або за значних накладних витрат вигоди від паралелізації можуть бути компенсовані витратами.
* Залежність даних: якщо існує значна залежність даних між зсувними елементами, це може обмежити ефективність паралелізації.

Висновок:Під час виконання лабораторної роботи я дізнався про особливості паралельних обчислень під платформою .NET. Як результат роботи, я створив однопотоковий та багатопотоковий алгоритми, що реалізують одну задачу. Було проведено тести порівняння ефективності, переваг та недоліків багатопотокових алгоритмів над однопотоковими.