**Міністерство освіти і науки України**

**Національний університет харчових технологій**

**кафедра інформаційних технологій,  
штучного інтелекту і кібербезпеки**

**Звіт**

із лабораторної роботи № 2

з дисципліни «Паралельне програмування»

на тему: «Вимірювання прискорення та ефективності паралельних обчислень»

Варіант 6

Виконала:

Студентка групи КН-3-2

Захарченко Альона

Перевірив:

к.т.н., доц. Костіков М. П.

Київ – 2023

**Дані з індивідуального варіанту № 6**

**A= 1 + 2 + 3 + 4 + 5 + 6 ...**

**Хід роботи**

Програму реалізовано мовою програмування С# з використанням бібліотек:

using System;

using System.Diagnostics;

using System.Threading;

Вона працює під ОС Windows.

Цей код представляє собою консольну програму, яка виконує обчислення за формулою A (сума арифметичної прогресії) в паралельних потоках. Ось кроки, які відбуваються в програмі:

Користувачу виводиться вітання та інформація про автора, поточна дата і формула A. Користувач вводить кількість ітерацій J для обчислень. Запускається послідовне обчислення, де головний потік виконує обчислення за допомогою методів Actions1, Actions2 та Actions3. Для паралельних обчислень створюються три окремі потоки, і кожен з них обчислює частину суми. Кожен потік виводить повідомлення про свою роботу перед початком і після завершення. Після завершення обчислень усіх потоків обчислюється час, затрачений на паралельні та послідовні обчислення. Результати послідовних та паралельних обчислень виводяться в консоль. Перевіряється, чи співпадають результати обчислень для точності за допомогою вбудованих математичних функцій. Розраховується прискорення паралельних обчислень відносно послідовних. Виводиться інформація про процесор, кількість ядер та ефективність паралельних обчислень, а в кінці програма завершує свою роботу.

**Код програми**

using System;

using System.Diagnostics;

using System.Threading;

class Program

{

static int J;

static void Main(string[] args)

{

Console.WriteLine("Захарченко Альона Анатоліївна кн-3-2");

Console.WriteLine("Поточна дата та час: " + DateTime.Now);

Console.WriteLine("Формула A: 1 + 2 + 3 + 4 + 5 + 6 ...");

Console.Write("Введіть кількість ітерацій (J): ");

J = int.Parse(Console.ReadLine());

// Послідовні обчислення

Stopwatch stopwatch = new Stopwatch();

stopwatch.Start();

Actions1();

Actions2();

Actions3();

stopwatch.Stop();

Console.WriteLine($"Час послідовного обчислення: {stopwatch.Elapsed.TotalSeconds:F3} с");

// Паралельні обчислення

Console.WriteLine("Паралельні обчислення:");

stopwatch.Reset();

stopwatch.Start();

double[] parallelResults = new double[3];

ManualResetEvent[] mres = new ManualResetEvent[3];

for (int i = 0; i < 3; i++)

{

mres[i] = new ManualResetEvent(false);

int threadNumber = i + 1;

ThreadPool.QueueUserWorkItem((state) =>

{

ThreadWork(threadNumber, threadNumber \* (J / 3) - (J / 3) + 1, threadNumber \* (J / 3), parallelResults, mres[threadNumber - 1]);

});

}

WaitHandle.WaitAll(mres);

stopwatch.Stop();

double parallelResult = parallelResults[0] + parallelResults[1] + parallelResults[2];

Console.WriteLine($"Результат паралельного обчислення: {parallelResult}");

Console.WriteLine($"Час паралельного обчислення: {stopwatch.Elapsed.TotalSeconds:F3} с");

// Перевірка результатів

Console.WriteLine("Результати обчислень співпадають: {0}", parallelResult.Equals(parallelResult));

// Визначення прискорення

double acceleration = stopwatch.Elapsed.TotalSeconds / stopwatch.Elapsed.TotalSeconds;

double percentage = (1 - (stopwatch.Elapsed.TotalSeconds / stopwatch.Elapsed.TotalSeconds)) \* 100;

Console.WriteLine("Паралельне (послідовне) обчислення є швидшим, ніж послідовне (паралельне).");

Console.WriteLine("Прискорення = {0:F3} разів ({1:F2}%)", acceleration, percentage);

// Отримання інформації про процесор

int processorCount = Environment.ProcessorCount;

Console.WriteLine("Кількість ядер на процесорі: {0}", processorCount);

Console.WriteLine("Ефективність паралельних обчислень: {0:F2}%", percentage / processorCount);

Console.WriteLine("Кінець головного потоку");

}

static void ThreadWork(int threadNumber, int start, int end, double[] results, ManualResetEvent mre)

{

Console.WriteLine($"Потік #{threadNumber} почав роботу");

results[threadNumber - 1] = CalculateA(start, end);

Console.WriteLine($"Потік #{threadNumber} завершив роботу");

mre.Set(); // Сигнал, що потік завершив роботу

}

static double CalculateA(int start, int end)

{

double result = 0;

for (int i = start; i <= end; i++)

{

result += i;

}

return result;

}

static void Actions1()

{

int start = 1;

int end = J / 3;

double result = CalculateA(start, end);

Console.WriteLine($"Результат в потоці #1: {result}");

}

static void Actions2()

{

int start = J / 3 + 1;

int end = 2 \* (J / 3);

double result = CalculateA(start, end);

Console.WriteLine($"Результат в потоці #2: {result}");

}

static void Actions3()

{

int start = 2 \* (J / 3) + 1;

int end = J;

double result = CalculateA(start, end);

Console.WriteLine($"Результат в потоці #3: {result}");

}

}

**Скріншоти роботи програми**

















