Лабораторна робота №3

Горецький Максим

1. На прямокутній ділянці поверхні екологи рівномірно встановили датчики, які щосекунди фіксують приріст забруднення, що відбувається хаотично. Час, за яких інформація з датчика потрапляє на обробку, становить десять мілісекунд.

А. Визначити значення, місце та час, коли стався найбільший приріст; час, коли на поверхні відбувся найбільший приріст; місце, де за період спостереження відбулося найбільше забруднення. В програмі передбачити декілька потоків. Визначте час роботи програми в безпотоковому та потоковому варіанті обчислень.

Б. Визначити час, коли кожен з потоків переходить в стан закінчення роботи. Підберіть параметри дослідження так, щоб період від стану предстартовий головної програми до стану завершення роботи становим не менше 10 секунд. Визначте стан головної програми і потоків через 10 секунд після старту. Оптимізуйте програму з метою рівномірного завантаження потоків.

using System;

using System.Threading;

using System.Diagnostics;

class Lab3

{

static int rows = 10;

static int cols = 10;

static int timePoints = 15;

static int[,,] data = new int[rows, cols, timePoints];

static Random rand = new Random();

static int maxValue = 0;

static int maxRow = 0, maxCol = 0, maxTime = 0;

static int maxTimeSum = 0;

static int maxTimeIndex = 0;

static int maxLocationSum = 0;

static int maxLocationRow = 0, maxLocationCol = 0;

static object lockObj = new object();

static void GenerateData()

{

Console.WriteLine("Генерація даних з датчиків...\n");

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

for (int j = 0; j < cols; j++)

for (int t = 0; t < timePoints; t++)

{

data[i, j, t] = rand.Next(1, 100);

Thread.Sleep(10);

}

}

static void FindMaxValue()

{

Console.WriteLine("Потік 1: Пошук максимального приросту...");

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

for (int j = 0; j < cols; j++)

for (int t = 0; t < timePoints; t++)

{

if (data[i, j, t] > maxValue)

{

lock (lockObj)

{

maxValue = data[i, j, t];

maxRow = i;

maxCol = j;

maxTime = t;

}

}

}

Console.WriteLine($"Потік 1: Завершено. Макс = {maxValue} в точці ({maxRow},{maxCol}) час {maxTime}");

}

static void FindMaxTimePoint()

{

Console.WriteLine("Потік 2: Пошук часу з найбільшим забрудненням...");

for (int t = 0; t < timePoints; t++)

{

int sum = 0;

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

for (int j = 0; j < cols; j++)

sum += data[i, j, t];

if (sum > maxTimeSum)

{

lock (lockObj)

{

maxTimeSum = sum;

maxTimeIndex = t;

}

}

}

Console.WriteLine($"Потік 2: Завершено. Час {maxTimeIndex} сума {maxTimeSum}");

}

static void FindMaxLocation()

{

Console.WriteLine("Потік 3: Пошук місця з найбільшим забрудненням...");

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

for (int j = 0; j < cols; j++)

{

int sum = 0;

for (int t = 0; t < timePoints; t++)

sum += data[i, j, t];

if (sum > maxLocationSum)

{

lock (lockObj)

{

maxLocationSum = sum;

maxLocationRow = i;

maxLocationCol = j;

}

}

}

Console.WriteLine($"Потік 3: Завершено. Точка ({maxLocationRow},{maxLocationCol}) сума {maxLocationSum}");

}

static void TaskA\_WithThreads()

{

Console.WriteLine("\n=== ЗАВДАННЯ А (з потоками) ===\n");

GenerateData();

Stopwatch sw = Stopwatch.StartNew();

Thread t1 = new Thread(FindMaxValue);

Thread t2 = new Thread(FindMaxTimePoint);

Thread t3 = new Thread(FindMaxLocation);

Console.WriteLine($"Старт потоків: {DateTime.Now:HH:mm:ss.fff}\n");

t1.Start();

t2.Start();

t3.Start();

t1.Join();

Console.WriteLine($"Потік 1 завершився: {DateTime.Now:HH:mm:ss.fff}");

t2.Join();

Console.WriteLine($"Потік 2 завершився: {DateTime.Now:HH:mm:ss.fff}");

t3.Join();

Console.WriteLine($"Потік 3 завершився: {DateTime.Now:HH:mm:ss.fff}");

sw.Stop();

Console.WriteLine("\n--- РЕЗУЛЬТАТИ ---");

Console.WriteLine($"Найбільший приріст: {maxValue} в точці ({maxRow},{maxCol}) час {maxTime}с");

Console.WriteLine($"Час з найбільшим забрудненням: {maxTimeIndex}с (сума {maxTimeSum})");

Console.WriteLine($"Місце з найбільшим забрудненням: ({maxLocationRow},{maxLocationCol}) (сума {maxLocationSum})");

Console.WriteLine($"\nЧас виконання з потоками: {sw.ElapsedMilliseconds} мс");

}

static void TaskA\_NoThreads()

{

Console.WriteLine("\n=== ЗАВДАННЯ А (без потоків) ===\n");

maxValue = 0;

maxTimeSum = 0;

maxLocationSum = 0;

Stopwatch sw = Stopwatch.StartNew();

FindMaxValue();

FindMaxTimePoint();

FindMaxLocation();

sw.Stop();

Console.WriteLine("\n--- РЕЗУЛЬТАТИ ---");

Console.WriteLine($"Найбільший приріст: {maxValue} в точці ({maxRow},{maxCol}) час {maxTime}с");

Console.WriteLine($"Час з найбільшим забрудненням: {maxTimeIndex}с (сума {maxTimeSum})");

Console.WriteLine($"Місце з найбільшим забрудненням: ({maxLocationRow},{maxLocationCol}) (сума {maxLocationSum})");

Console.WriteLine($"\nЧас виконання без потоків: {sw.ElapsedMilliseconds} мс");

}

static void TaskB()

{

Console.WriteLine("\n\n=== ЗАВДАННЯ Б ===\n");

Console.WriteLine("Запуск з контролем станів через 10 секунд...\n");

rows = 20;

cols = 20;

timePoints = 30;

data = new int[rows, cols, timePoints];

GenerateData();

bool[] finished = new bool[3];

Stopwatch sw = Stopwatch.StartNew();

Thread t1 = new Thread(() => { FindMaxValue(); finished[0] = true; });

Thread t2 = new Thread(() => { FindMaxTimePoint(); finished[1] = true; });

Thread t3 = new Thread(() => { FindMaxLocation(); finished[2] = true; });

Console.WriteLine($"Старт: {DateTime.Now:HH:mm:ss.fff}");

t1.Start();

t2.Start();

t3.Start();

Thread.Sleep(10000);

Console.WriteLine($"\n--- Стан через 10 секунд ---");

Console.WriteLine($"Потік 1: {(finished[0] ? "Завершено" : "Працює")}");

Console.WriteLine($"Потік 2: {(finished[1] ? "Завершено" : "Працює")}");

Console.WriteLine($"Потік 3: {(finished[2] ? "Завершено" : "Працює")}");

Console.WriteLine($"Головна програма: Очікує завершення потоків");

t1.Join();

t2.Join();

t3.Join();

sw.Stop();

Console.WriteLine($"\nЗагальний час: {sw.ElapsedMilliseconds} мс");

}

static void Main()

{

TaskA\_WithThreads();

TaskA\_NoThreads();

TaskB();

}

}



