

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕЛЕКТРИЧНОЇ ІНЖЕНЕРІЇ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра автоматизації та інформаційних систем

Навчальна дисципліна
«ПАРАЛЕЛЬНІ ТА РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ»

ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ № 5

Виконав
студент групи КН-23-1
Іщенко.Є.В
Перевірила
доцент кафедри АІС
Істоміна Н. М.

Кременчук 2025

Лабораторна робота № 5

Тема: Дослідження ієрархічної організації пам'яті

Мета: набути навичок оцінювання взаємодії з пам'яттю комп'ютера.

Хід роботи:

Отже у межах лабораторної роботи потрібно виконати такі дії:

1. Створити програмний застосунок, що проводить перемноження двох довільних матриць (із глибиною та шириною не менше, ніж 250 елементів) згідно з варіантом 1 перебору циклу.
2. До звіту підготувати таблицю для занесення експериментальних даних
3. Провести 5 дослідів з першим варіантом застосунку.
4. Змінити застосунок так, щоб перебір циклу здійснювався згідно з варіантом 2.
5. Провести 5 дослідів із другим варіантом застосунку.
6. Оцінити, наскільки порядок доступу до даних впливає на швидкість роботи застосунку.
7. У звіті навести характеристики свого процесора та оперативної пам'яті.
8. До звіту додати код двох варіантів застосунків.

Завдання:

Створимо програмний застосунок до завдання, що проводить перемноження двох довільних матриць (із глибиною та шириною не менше, ніж 250 елементів) згідно з варіантом 1 перебору циклу.

```

using System;
using System.Diagnostics;

class Program
{
    static void Main(string[] args)
    {
        Console.OutputEncoding = System.Text.Encoding.Default;

        int L = 250; // Розмір матриці

        // Ініціалізація матриць
        double[,] arrayA = GenerateRandom(L, L);
        double[,] arrayB = GenerateRandom(L, L);
        double[,] arrayR = new double[L, L];

        // Вимірювання часу
        Stopwatch stopwatch = new Stopwatch();

        stopwatch.Start();

        Variant1(arrayA, arrayB, arrayR, L);

        stopwatch.Stop();

        Console.WriteLine($"Час виконання: {stopwatch.ElapsedMilliseconds} мс");
    }

    static double[,] GenerateRandom(int rows, int cols)
    {
        Random rand = new Random();
        double[,] matrix = new double[rows, cols];

        for (int i = 0; i < rows; i++)
            for (int j = 0; j < cols; j++)
                matrix[i, j] = rand.NextDouble() * 100;

        return matrix;
    }

    static void Variant1(double[,] arrayA, double[,] arrayB, double[,] arrayD, int L)
    {
        for (int i = 0; i < L; i++)
            for (int k = 0; k < L; k++)
                for (int j = 0; j < L; j++)
                    arrayD[i, j] += arrayA[i, k] * arrayB[k, j];
    }
}

```

Рисунок 1.1 – Код

На рисунках 1.2 – 1.6 наведено 5 дослідів з першим варіантом застосунку.

Час виконання: 59 мс

Рисунок 1.2 – Перший дослід з першим варіантом застосунку

Час виконання: 73 мс

Рисунок 1.3 – Другий дослід з першим варіантом застосунку

Час виконання: 90 мс

Рисунок 1.4 – Третій дослід з першим варіантом застосунку

Час виконання: 73 мс

Рисунок 1.5 – Четвертий дослід з першим варіантом застосунку

Час виконання: 58 мс

Рисунок 1.6 – П'ятий дослід з першим варіантом застосунку

Створимо програмний застосунок до завдання, що проводить перемноження двох довільних матриць (із глибиною та шириною не менше, ніж 250 елементів) згідно з варіантом 2 перебору циклу.

```
using System;
using System.Diagnostics;

class Program
{
    static void Main(string[] args)
    {
        Console.OutputEncoding = System.Text.Encoding.Default;

        int L = 250; // Розмір матриці

        // Ініціалізація матриць
        double[,] arrayA = GenerateRandom(L, L);
        double[,] arrayB = GenerateRandom(L, L);
        double[,] arrayR = new double[L, L];

        // Вимірювання часу
        Stopwatch stopwatch = new Stopwatch();

        stopwatch.Start();

        Variant2(arrayA, arrayB, arrayR, L);

        stopwatch.Stop();

        Console.WriteLine($"Час виконання: {stopwatch.ElapsedMilliseconds} мс");
    }

    static double[,] GenerateRandom(int rows, int cols)
    {
        Random rand = new Random();
        double[,] matrix = new double[rows, cols];

        for (int i = 0; i < rows; i++)
            for (int j = 0; j < cols; j++)
                matrix[i, j] = rand.NextDouble() * 100;

        return matrix;
    }

    static void Variant2(double[,] arrayA, double[,] arrayB, double[,] arrayD, int L)
    {
        for (int j = 0; j < L; j++)
            for (int k = 0; k < L; k++)
                for (int i = 0; i < L; i++)
                    arrayD[i, j] += arrayA[i, k] * arrayB[k, j];
    }
}
```

Рисунок 1.6 – Код

На рисунках 1.7 – 1.11 наведено 5 дослідів з другим варіантом застосунку.

Час виконання: 64 мс

Рисунок 1.7 – Перший дослід з другим варіантом застосунку

Час виконання: 67 мс

Рисунок 1.8 – Другий дослід з другим варіантом застосунку

Час виконання: 63 мс

Рисунок 1.9 – Третій дослід з другим варіантом застосунку

Час виконання: 64 мс

Рисунок 1.10 – Четвертий дослід з другим варіантом застосунку

Час виконання: 63 мс

Рисунок 1.11 – П'ятий дослід з другим варіантом застосунку

Таблиця 1.1 – Отримані експериментальні дані

Перебір циклу	Витрачений час, мс					Середнє значення
	номер дослідів					
	№ 1	№ 2	№ 3	№ 4	№ 5	
Варіант 1	59	73	90	73	58	70,6
Варіант 2	64	67	63	64	53	62.2

Висновки:

На цій лабораторній роботі ми досліджували ієрархічну організацію пам'яті, набули навичок оцінювання взаємодії з пам'яттю комп'ютера. Створили дві програми мовою C# з двома варіантами множення матриць. Вирахували швидкість обчислення процесором операції, запустивши 5

операцій і обчисливши середнє значення. Другий варіант спрацював у більшості досліджень – швидше за другий.