Міністерство освіти і науки України

Дніпровський національний університет ім. Олеся Гончара

Факультет фізики, електроніки та комп’ютерних систем

Кафедра ЕОМ

Звіт

до лабораторної роботи №2

з теми «Процеси»

з дисципліни «Системне програмне забезпечення»

Виконав:

студент групи КІ-21-2

спеціальності

123 «Комп’ютерна інженерія»

Лебідь Олександр Павлович

Перевірив:

Герасимов Володимир Володимирович

Дніпро

2024

**Мета роботи:** вивчити принципи застосування пріоритетності запуску та

роботи процесів на мові C#, навчитися створювати дочірній процес та очікувати завершення його роботи.

**Хід роботи**

**Завдання**

1. Проаналізуйте та виконайте усі приклади розглянутих програм.

2. Виконайте усі приклади програм для 2, 3 та N процесів. Наочно покажіть паралельність виконання процесів.

3. Виконайте усі приклади програм для 2, 3 та N процесів в проектах C# Windows Forms.

4. Виконайте свій варіант відповідно до списку в підгрупи в проекті C# Windows Forms:

Варіант 5

Написати програму, яка запускає N дочірніх процесів Блокнот (notepad). Після закриття i-го Блокнота, видає повідомлення, що «i-ий Блокнот закритий».

**Form1.cs:**using System.Diagnostics;

namespace n2l4

{

public partial class Form1 : Form

{

private int \_processCount = 0;

private int \_closedProcessCount = 0;

private Dictionary<Process, string> \_processNames = new Dictionary<Process, string>();

private string \_folderPath = @"C:\DeskTop\c#";

public Form1()

{

InitializeComponent();

Directory.CreateDirectory(\_folderPath);

}

private void StartProcesses(int n)

{

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

{

string fileName = Path.Combine(\_folderPath, $"Блокнот {i + 1}.txt");

File.Create(fileName).Close();

Process process = new Process();

process.StartInfo.FileName = "notepad.exe";

process.StartInfo.Arguments = $"\"{fileName}\"";

process.EnableRaisingEvents = true;

process.Exited += EndProcesses!;

process.Start();

\_processCount++;

\_processNames[process] = $"Блокнот {i + 1}";

}

}

private void EndProcesses(object sender, EventArgs e)

{

Process process = sender as Process;

if (process != null && \_processNames.ContainsKey(process))

{

string processName = \_processNames[process];

MessageBox.Show($"{processName} закрыт");

\_processNames.Remove(process);

}

\_closedProcessCount++;

if (\_closedProcessCount == \_processCount)

{

MessageBox.Show("Все процессы Блокнота закрыты");

\_closedProcessCount = 0;

\_processCount = 0;

}

}

private void button1\_Click(object sender, EventArgs e)

{

if (int.TryParse(textBox1.Text, out int n))

{

StartProcesses(n);

}

else

{

MessageBox.Show("Введите корректное число");

}

}

}

}

**Form1.Designer.cs:**

namespace n2l4

{

partial class Form1

{

/// <summary>

/// Required designer variable.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

button1 = new Button();

textBox1 = new TextBox();

SuspendLayout();

//

// button1

//

button1.Location = new Point(151, 213);

button1.Name = "button1";

button1.Size = new Size(75, 23);

button1.TabIndex = 0;

button1.Text = "button1";

button1.UseVisualStyleBackColor = true;

button1.Click += button1\_Click;

//

// textBox1

//

textBox1.Location = new Point(151, 169);

textBox1.Name = "textBox1";

textBox1.Size = new Size(100, 23);

textBox1.TabIndex = 1;

//

// Form1

//

AutoScaleDimensions = new SizeF(7F, 15F);

AutoScaleMode = AutoScaleMode.Font;

ClientSize = new Size(800, 450);

Controls.Add(textBox1);

Controls.Add(button1);

Name = "Form1";

Text = "Form1";

ResumeLayout(false);

PerformLayout();

}

#endregion

private Button button1;

private TextBox textBox1;

}

}

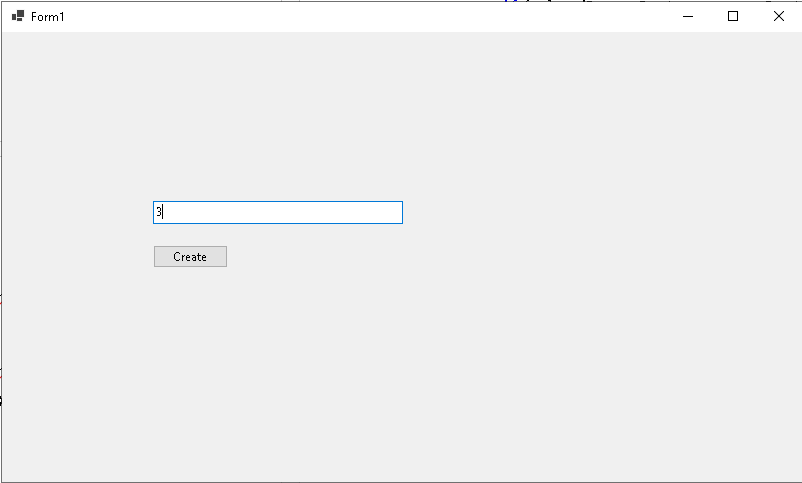


Рис. 1 – Введення к-сть дочірніх процесів Блокнот

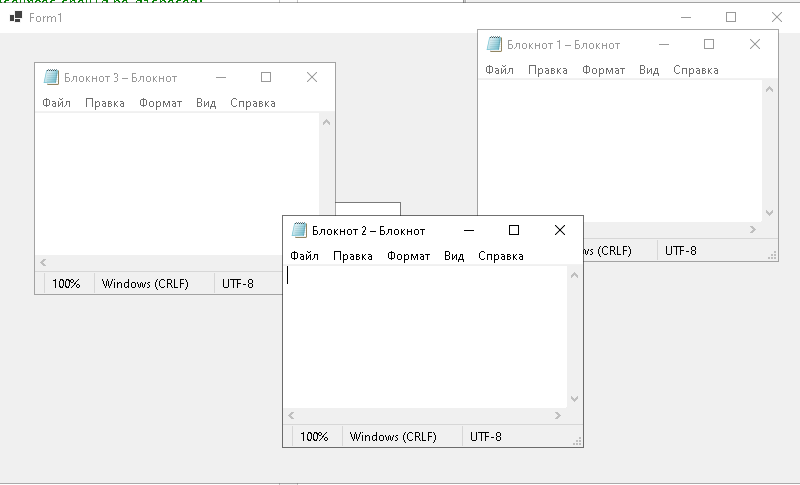


Рис. 2 – Запуск N дочірніх процесів Блокнот

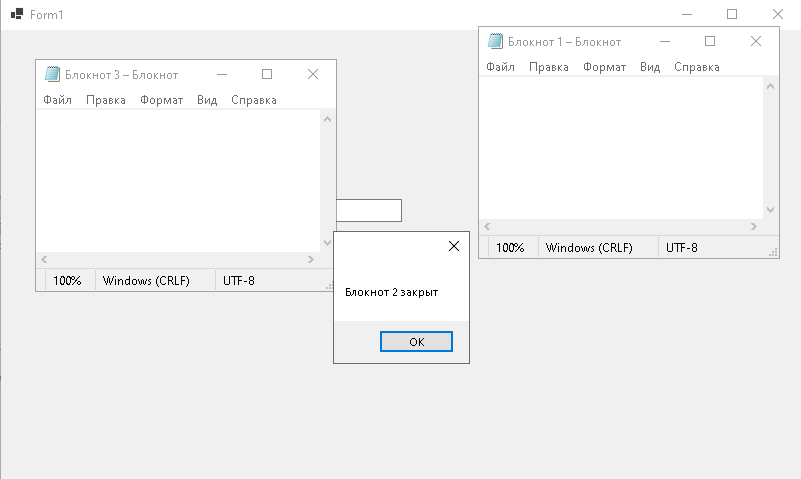


Рис. 3 – Повідомлення при закриття i-го Блокнота

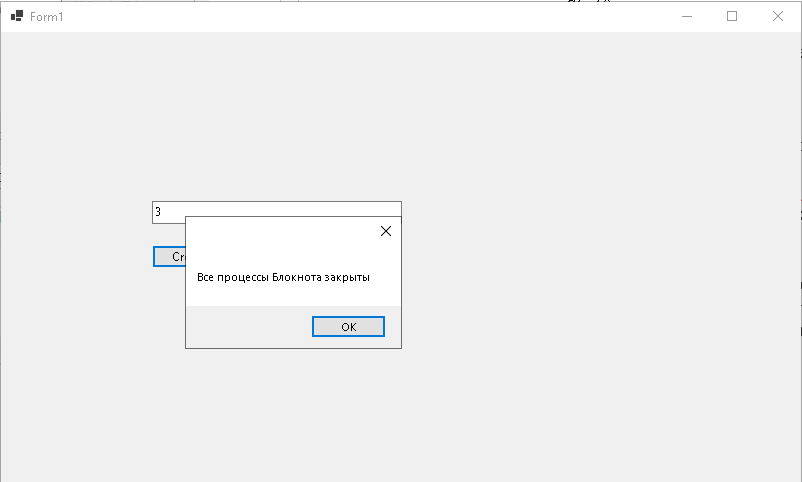


Рис. 4 – Повідомлення при закриття усіх процесів Блокнота

**Контрольні питання**

1. **Що таке процес?**

**Процес –** це програма, яка виконується на комп’ютері. Він має свій власний простір пам’яті і складається з одного або більше потоків, які виконують код.

1. **Який клас і який простір імен використовується для створення**

**процесу?**

Для створення процесу в C# використовується клас Process з простору імен System.Diagnostics.

1. **Як створити і запустити процес?**

Щоб створити і запустити процес, ви можете використати метод Start класу Process. Наприклад:

Process process = new Process();

process.StartInfo.FileName = "notepad.exe";

process.Start();

1. **Як зупинити виконання процесу?**

Щоб зупинити виконання процесу, ви можете використати метод Kill класу Process. Наприклад:

process.Kill();

1. **Як визначити завершення процесу?**

Щоб визначити, чи завершився процес, ви можете перевірити властивість HasExited класу Process.

1. **Що запускає метод Start() класу Process?**

Метод Start() класу Process запускає процес, який вказаний в ProcessStartInfo.FileName.

**Висновок:** Під час виконання лабораторної роботи було вивчено принципи застосування пріоритетності запуску та роботи процесів на мові C#, вивчено створювання дочірніх процесів та очікування завершення їх роботи.