**ЛАБОРАТОРНА РОБОТА № 2**

з дисципліни

«Системне програмне забезпечення»

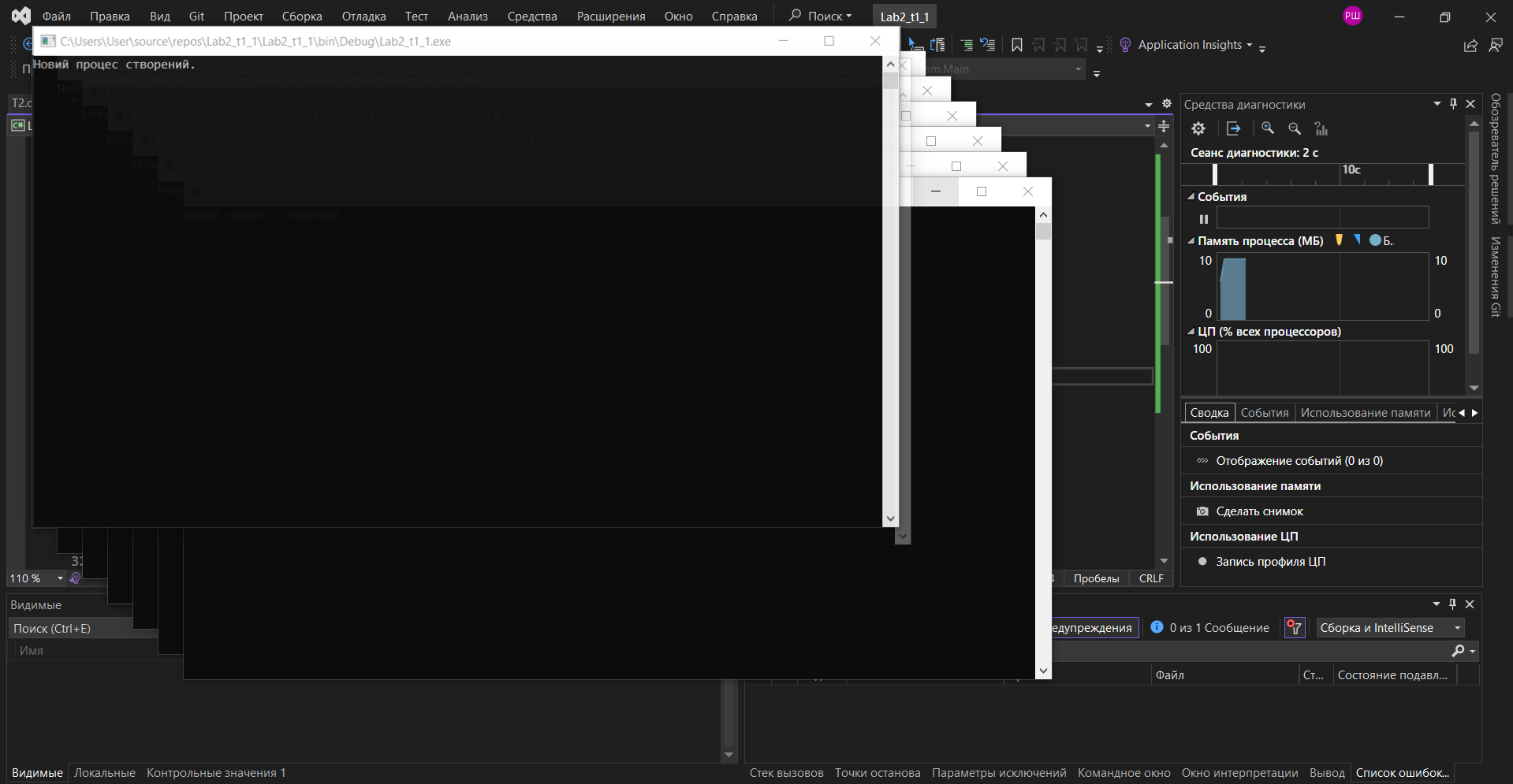
Шейко Р. О.

Групи КІ-21-2

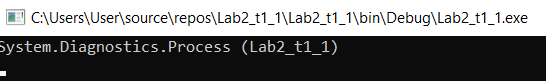
1. Проаналізуйте та виконайте усі приклади розглянутих програм.
   1. Запустимо код лістинг 2.1 з методички



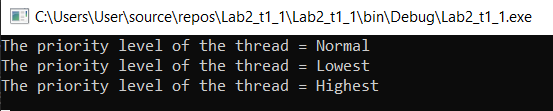
* 1. Запустимо код лістинг 2.2 з методички



* 1. Запустимо код лістинг 2.3 з методички



* 1. Запустимо код лістинг 2.4 з методички



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

*Лістинг 2.1. Файл «Program.cs»*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab2\_t2

{

internal class Program

{

private static string ExitConst = "exit";

private static string FirstConst = "1";

private static string SecondConst = "2";

private static string ThirdthConst = "3";

private static string FourthConst = "4";

public static string MenuConst = "Menu" +

$"\n{FirstConst}. Task 1;" +

$"\n{SecondConst}. Task 2;" +

$"\n{ThirdthConst}. Task 3;" +

$"\n{FourthConst}. Task 4;" +

$"\n{ExitConst}. exit";

private static string WriteNMessage = "введіть кількість N: ";

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.UTF8;

string str = string.Empty;

do

{

Console.WriteLine(MenuConst);

str = Console.ReadLine();

if (str.ToLowerInvariant().Equals(FirstConst))

T1.start(takeN(WriteNMessage));

else if (str.ToLowerInvariant().Equals(SecondConst))

T2.start(takeN(WriteNMessage));

else if (str.ToLowerInvariant().Equals(ThirdthConst))

T3.start(takeN(WriteNMessage));

else if (str.ToLowerInvariant().Equals(FourthConst))

T4.start(takeN(WriteNMessage));

} while (!str.Equals(ExitConst));

}

static int takeN(string message)

{

int n = 0;

string str = string.Empty;

do {

Console.WriteLine(message);

str = Console.ReadLine();

try {

n = int.Parse(str);

} catch (Exception e)

{

Console.WriteLine(e.ToString());

}

} while (!(n > 0));

return n;

}

}

}

*Лістинг 2.2. Файл «T1.cs»*

using System;

using System.Threading;

namespace Lab2\_t2

{

class T1

{

public static void start(int n)

{

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

createTask(i);

}

private static void createTask(int n)

{

Console.Write($"Дочiрнiй процес створений #{n}: ");

for (int i = 1; i <= 10; i++)

{

Console.Write(i + " ");

Thread.Sleep(200);

}

Console.WriteLine();

}

}

}

*Лістинг 2.3. Файл «T2.cs»*

using System.Threading;

using System;

using System.Diagnostics;

namespace Lab2\_t2

{

class T2

{

public static void start(int n)

{

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

createTask(i);

}

private static void createTask(int n)

{

Process newProc = new Process();

newProc.StartInfo.FileName =

@"C:\Users\User\source\repos\Lab1\_t1\_2\Lab1\_t1\_2\bin\Debug\Lab1\_t1\_2.exe";

newProc.StartInfo.UseShellExecute = true;

newProc.Start();

Console.WriteLine($"Новий процес {n} створений.\n");

newProc.WaitForExit();

newProc.Close();

Console.WriteLine($"Новий процес {n} закiнчений.\n");

Environment.Exit(0);

}

}

}

*Лістинг 2.3. Файл «T3.cs»*

using System.Threading;

using System;

using System.Diagnostics;

namespace Lab2\_t2

{

class T3 {

public static void start(int n)

{

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

createTask(i);

}

private static void createTask(int n)

{

Process newProc;

newProc = Process.GetCurrentProcess();

Console.WriteLine(newProc);

}

}

}

*Лістинг 2.4. Файл «T4.cs»*

using System.Diagnostics;

using System;

using System.Threading;

namespace Lab2\_t2

{

class T4 {

public static void start(int n)

{

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

createTask(i);

}

private static void createTask(int n)

{

Thread newThrd;

newThrd = Thread.CurrentThread;

Console.WriteLine("The priority level of the thread = " + newThrd.Priority);

newThrd.Priority = ThreadPriority.Lowest;

Console.WriteLine("The priority level of the thread = " + newThrd.Priority);

newThrd.Priority = ThreadPriority.Highest;

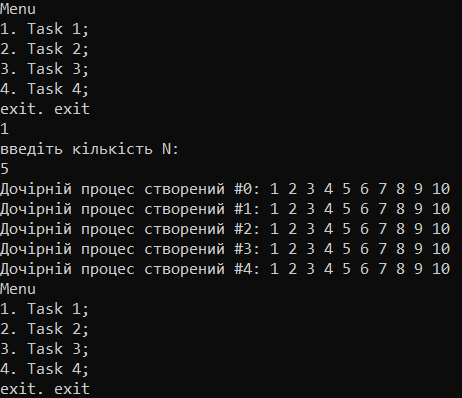
Console.WriteLine("The priority level of the thread = " + newThrd.Priority);

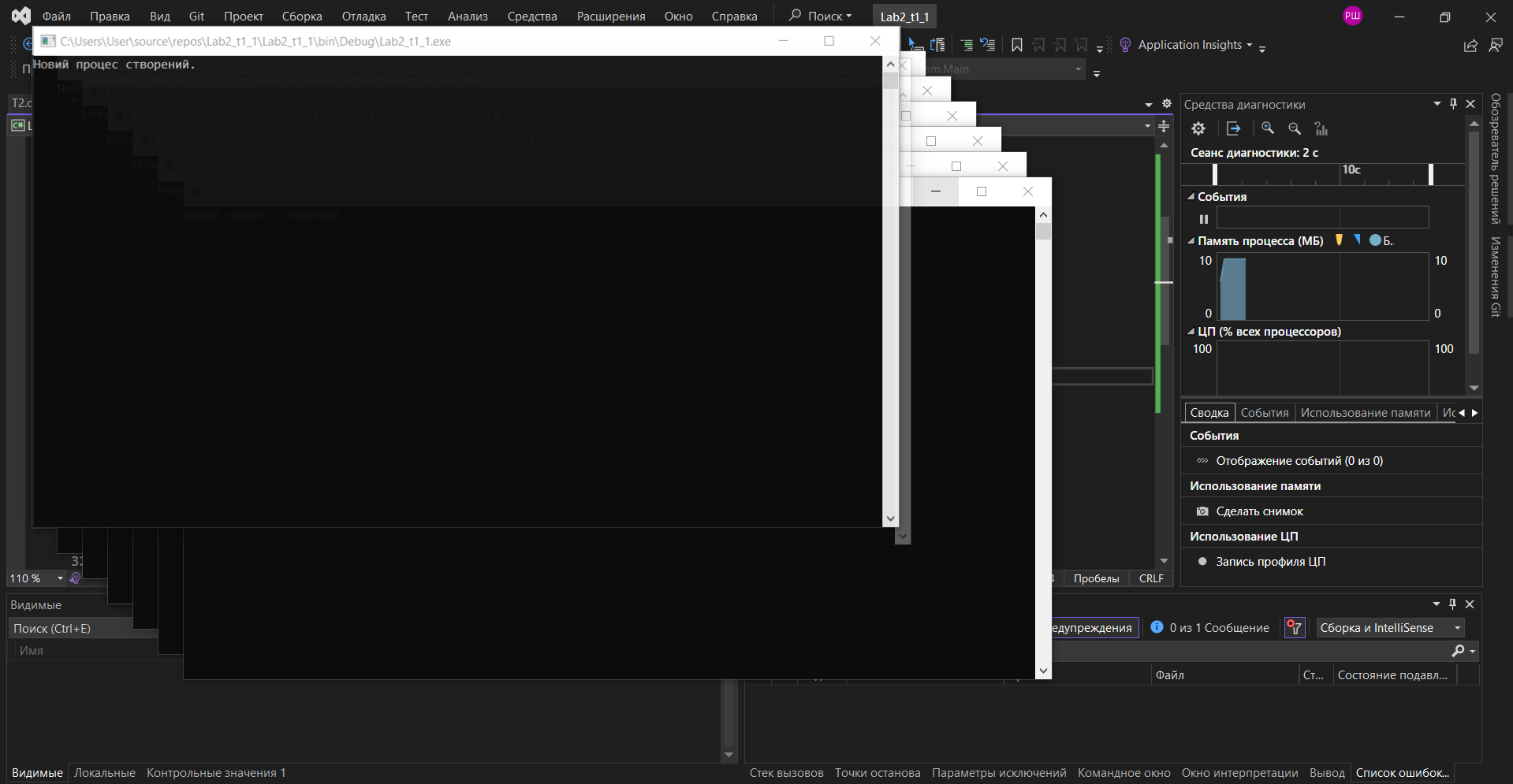
}

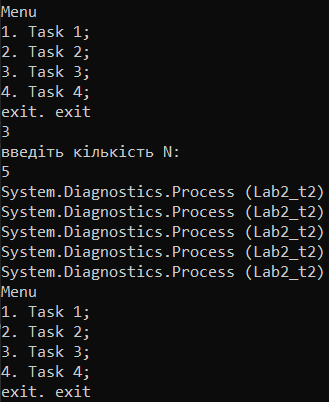
}

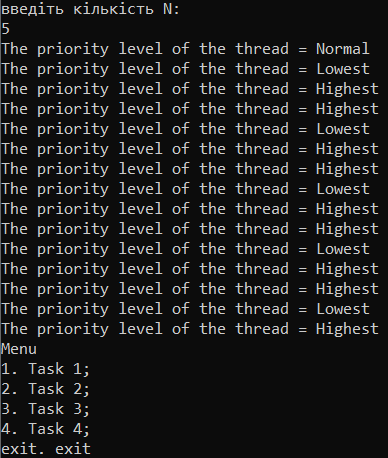
}

Запустимо приклади:









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

*Лістинг 3.1*

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab2\_t3\_1

{

public partial class Form1 : Form

{

private Label label;

private Label label1;

public Form1()

{

InitializeComponent();

label = new Label { Parent = this, Location = new System.Drawing.Point(10, 10) };

label1 = new Label { Parent = this, Location = new System.Drawing.Point(10, 30) };

label.SetBounds(10, 10, label.Size.Width + 200, label.Size.Height);

label1.SetBounds(10, 30, label.Size.Width + 200, label.Size.Height);

}

private void ClearProgram()

{

label.Text = "";

label1.Text = "";

}

private class T1

{

public static void start(int n)

{

Thread[] tt = new Thread[n];

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

{

tt[i] = new Thread(() => createTask(i));

tt[i].Start();

}

}

private static void createTask(int n)

{

Form1 form1 = Application.OpenForms[0] as Form1;

if (form1 == null)

return;

string str = "Дочiрнiй процес створений #{ " + "}: ";

form1.SetText(form1.label, str);

for (int i = 1; i <= 10; i++)

{

form1.SetText(form1.label1, i + " ");

Thread.Sleep(200);

}

form1.SetText(form1.label1, "\n");

}

}

private void SetText(Control ctrl, string text)

{

if (ctrl.InvokeRequired)

{

BeginInvoke((Action)(() => SetText(ctrl, text)));

return;

}

ctrl.Text = text;

}

private void toolTip1\_Popup(object sender, PopupEventArgs e)

{

}

private void Form1\_Load(object sender, EventArgs e)

{

}

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

{

}

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

{

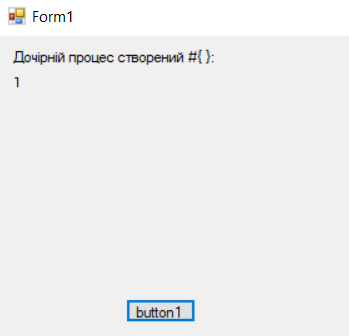
ClearProgram();

T1.start(5);

}

}

}



*Лістинг 3.2*

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab2\_t3\_2

{

public partial class Form1 : Form

{

private Label label;

private Label label1;

public Form1()

{

InitializeComponent();

label = new Label { Parent = this, Location = new System.Drawing.Point(10, 10) };

label1 = new Label { Parent = this, Location = new System.Drawing.Point(10, 30) };

}

private void Form1\_Load(object sender, EventArgs e) {

}

private void ClearProgram()

{

label.Text = "";

label1.Text = "";

}

private class T2

{

public static void start(int n)

{

Thread[] tt = new Thread[n];

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

{

tt[i] = new Thread(() => createTask(i));

tt[i].Start();

}

}

private static void createTask(int n)

{

Process newProc = new Process();

newProc.StartInfo.FileName = @"C:\Users\User\source\repos\Lab1\_t1\_2\Lab1\_t1\_2\bin\Debug\Lab1\_t1\_2.exe";

newProc.StartInfo.UseShellExecute = true;

newProc.Start();

Console.WriteLine($"Новий процес {n} створений.\n");

newProc.WaitForExit();

newProc.Close();

Console.WriteLine($"Новий процес {n} закiнчений.\n");

}

}

private void SetText(Control ctrl, string text)

{

if (ctrl.InvokeRequired)

{

BeginInvoke((Action)(() => SetText(ctrl, text)));

return;

}

ctrl.Text = text;

}

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

{

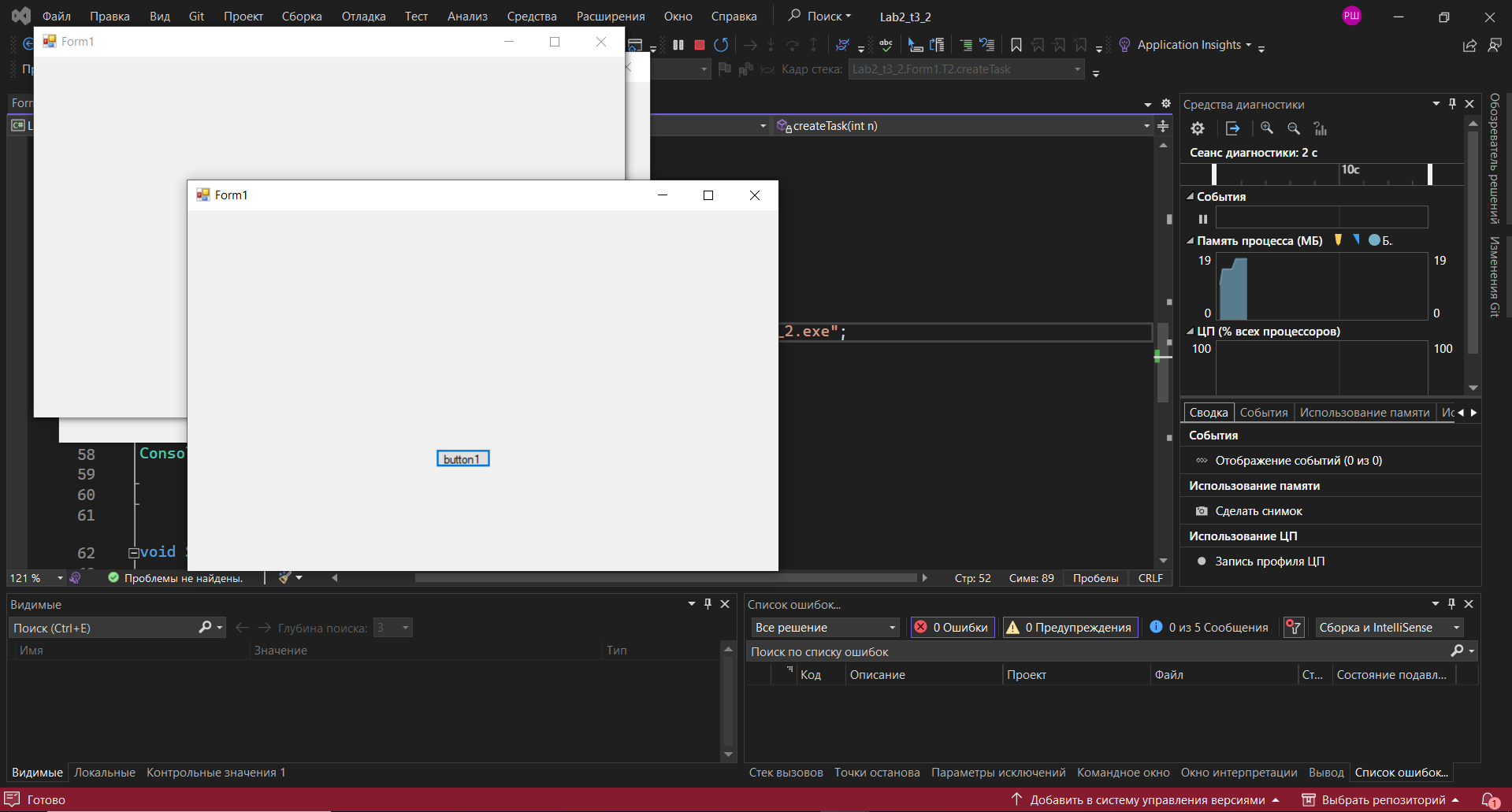
ClearProgram();

T2.start(3);

}

}

}



*Лістинг 3.3*

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab2\_t3\_3

{

public partial class Form1 : Form

{

private Label label;

private Label label1;

public Form1()

{

InitializeComponent();

label = new Label { Parent = this, Size = new Size(200, 100), Location = new System.Drawing.Point(10, 10) };

label1 = new Label { Parent = this, Size = new Size(200, 100), Location = new System.Drawing.Point(10, 30) };

}

private void Task3ToolStripMenuItem\_Click(object sender, EventArgs e)

{

ClearProgram();

T3.start(3, this);

}

private void ClearProgram()

{

label.Text = "";

label1.Text = "";

}

private class T3

{

public static void start(int n, Form1 form1)

{

Thread[] tt = new Thread[n];

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

{

tt[i] = new Thread(() => createTask(i, form1));

tt[i].Start();

}

}

private static void createTask(int n, Form1 form1)

{

Process newProc = Process.GetCurrentProcess();

string processInfo = newProc.ToString();

form1.SetText(form1.label, form1.label.Text + processInfo);

//form1.SetText(form1.label1, "");

Thread.Sleep(1000);

}

}

private void SetText(Control ctrl, string text)

{

if (ctrl.InvokeRequired)

{

BeginInvoke((Action)(() => SetText(ctrl, text)));

return;

}

ctrl.Text = text;

}

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

{

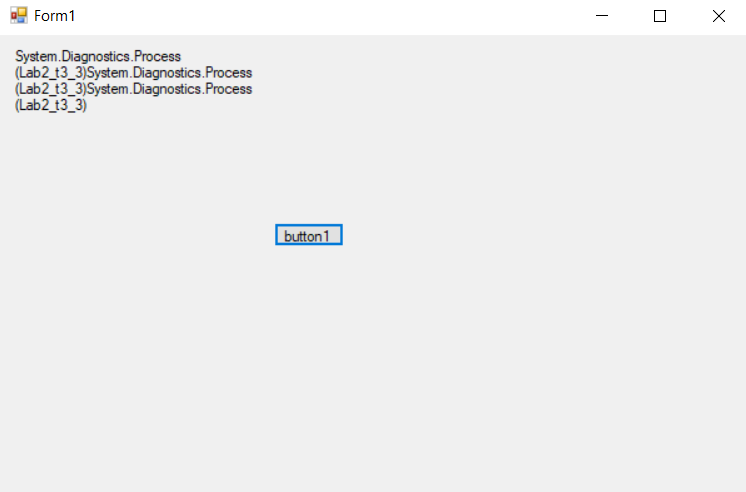
ClearProgram();

T3.start(3, this);

}

}

}



*Лістинг 3.4*

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab2\_t3\_4

{

public partial class Form1 : Form

{

private Label label;

private Label label1;

public Form1()

{

InitializeComponent();

label = new Label { Parent = this, Size = new Size(200, 100), Location = new System.Drawing.Point(10, 10) };

label1 = new Label { Parent = this, Size = new Size(200, 100), Location = new System.Drawing.Point(100, 400) };

label1.SetBounds(label1.Location.X, label1.Location.Y, label1.Width + 200, label1.Height);

}

private void ClearProgram()

{

label.Text = "";

label1.Text = "";

}

private class T4

{

public static void start(int n, Form1 form1)

{

Thread[] tt = new Thread[n];

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

{

tt[i] = new Thread(() => createTask(i, form1));

tt[i].Start();

}

}

private static void createTask(int n, Form1 form1)

{

Thread newThrd = Thread.CurrentThread;

Action<string> updateLabel = new Action<string>((text) =>

{

form1.SetText(form1.label, text);

});

Action<string> updateLabel1 = new Action<string>((text) =>

{

form1.SetText(form1.label1, text);

});

newThrd.Priority = ThreadPriority.Lowest;

form1.Invoke(updateLabel, $"Thread #{n} Priority: {newThrd.Priority}");

form1.Invoke(updateLabel1, $"Thread #{n} Priority changed to Lowest: {newThrd.Priority}");

Thread.Sleep(2000);

newThrd.Priority = ThreadPriority.Highest;

form1.Invoke(updateLabel1, $"Thread #{n} Priority changed to Highest: {newThrd.Priority}");

}

}

private void SetText(Control ctrl, string text)

{

if (ctrl.InvokeRequired)

{

BeginInvoke((Action)(() => SetText(ctrl, text)));

return;

}

ctrl.Text = text;

}

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

{

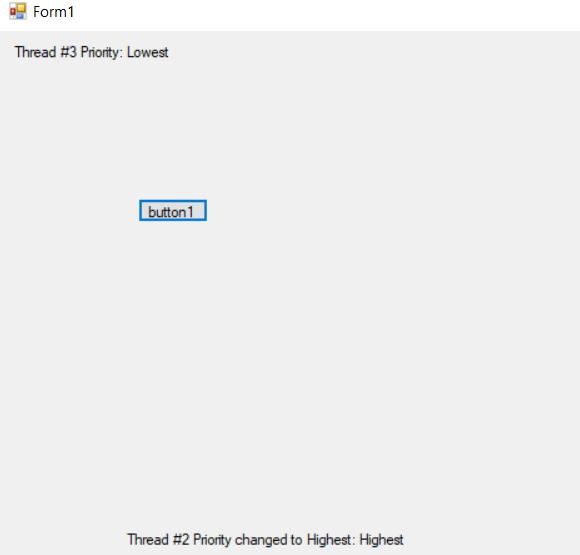
ClearProgram();

T4.start(3, this);

}

}

}



1. 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»

Лістинг 4

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab\_2\_t4\_v5

{

/\*

\*

\*

Написати програму, в якій перший потік при кожному натисканні на

кнопку «Завантажити» читає обраний файл з зображенням і виводить його у

вікно програми. При кожному натисканні на кнопку «Поворот»,

другий потік виконує поворот зображення на 90 градусів.

\*

\*

\*/

public partial class Form1 : Form

{

private Image image;

private readonly object imageLock = new object();

public Form1()

{

InitializeComponent();

Size = new Size(400, 400);

pictureBox.Parent = this;

pictureBox.Dock = DockStyle.Fill;

pictureBox.SizeMode = PictureBoxSizeMode.Zoom;

loadButton.Parent = this;

loadButton.Location = new Point(10, 10);

loadButton.Text = "Завантажити";

loadButton.Click += LoadButton\_Click;

rotateButton.Parent = this;

rotateButton.Location = new Point(100, 10);

rotateButton.Text = "Поворот";

rotateButton.Click += RotateButton\_Click;

}

private void Form1\_Load(object sender, EventArgs e) { }

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

{

var openFileDialog = new OpenFileDialog

{

Filter = "Image Files (\*.bmp, \*.jpg, \*.png, \*.gif)|\*.bmp;\*.jpg;\*.png;\*.gif"

};

if (openFileDialog.ShowDialog() == DialogResult.OK)

{

lock (imageLock)

{

image = Image.FromFile(openFileDialog.FileName);

}

pictureBox.Image = image;

}

}

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

{

var rotationThread = new Thread(RotateImage);

rotationThread.Start();

}

private void RotateImage()

{

lock (imageLock)

{

if (image != null)

{

image.RotateFlip(RotateFlipType.Rotate90FlipNone);

pictureBox.Image = image;

}

}

}

}

}

