Міністерство освіти і науки України

Дніпровський національний університет ім. Олеся Гончара

Факультет фізики, електроніки та комп’ютерних систем

Кафедра ЕОМ

Звіт

до лабораторної роботи №5

з теми «Канали передачі даних»

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:** розглянути застосування анонімного та іменованого каналів, розглянути різницю при їх використанні.

**Хід роботи**

**Завдання**

1. Виконати усі приклади програм.

2. Виконати усі приклади програм в проектах C# Windows Forms.

3. Написати програму. Використовуючи анонімний канал, сервер надсилає клієнту число і рядок. Клієнт повинен вивести рядок, який він отримує, задану кількість разів.

**l\_5\_3\_c:**

using System;

using System.Diagnostics;

using System.IO.Pipes;

using System.Security.Principal;

using System.Text;

namespace l\_5\_3\_c

{

public class Client

{

NamedPipeClientStream clientPipe;

int clientID;

public Client(string serverID)

{

clientID = Process.GetCurrentProcess().Id;

clientPipe = new NamedPipeClientStream(".", "pipe" + serverID, PipeDirection.InOut, PipeOptions.None, TokenImpersonationLevel.None);

}

public void Start()

{

try

{

clientPipe.Connect();

byte[] inBytes = new byte[100];

clientPipe.Read(inBytes, 0, 100);

string inStr = Encoding.ASCII.GetString(inBytes);

string[] parts = inStr.Split('|');

string message = parts[0];

int repeatCount = int.Parse(parts[1]);

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

{

Console.WriteLine("\t Received from server: {0}", message);

}

}

catch (Exception ex)

{

Console.WriteLine("\t Error: " + ex.Message);

return;

}

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine("\t Process client.");

Client client = new Client(args[0]);

client.Start();

Console.ReadKey();

}

}

}

**l\_5\_3\_s:**

using System;

using System.Diagnostics;

using System.IO.Pipes;

using System.Security.Principal;

using System.Text;

using System.Threading;

namespace l\_5\_3\_s

{

public class Server

{

const string EXE\_PATH = @"C:\Users\aleks\source\repos\l\_5\_3\_c\l\_5\_3\_c\bin\Debug\net8.0\l\_5\_3\_c.exe";

Thread[] serverThreads;

int numClients;

byte[] byteMessage;

public Server(int numClients)

{

this.numClients = numClients;

serverThreads = new Thread[numClients];

}

public void Start(string message, int repeatCount)

{

byteMessage = Encoding.ASCII.GetBytes(message + "|" + repeatCount.ToString());

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

{

serverThreads[i] = new Thread(ServerThread);

serverThreads[i].Start();

}

}

private void ServerThread()

{

try

{

int threadID = Thread.CurrentThread.ManagedThreadId;

NamedPipeServerStream serverPipe = new NamedPipeServerStream("pipe" + threadID, PipeDirection.InOut, 1);

ProcessStartInfo info = new ProcessStartInfo

{

FileName = EXE\_PATH,

Arguments = threadID.ToString()

};

Process client = Process.Start(info)!;

serverPipe.WaitForConnection();

serverPipe.Write(byteMessage, 0, byteMessage.Length);

serverPipe.WaitForPipeDrain();

}

catch (Exception ex)

{

Console.WriteLine("Error: " + ex.Message);

return;

}

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Process Server.");

Console.Write("Count of processes: ");

int count = Convert.ToInt32(Console.ReadLine());

Server server = new Server(count);

Console.Write("Enter message: ");

string message = Console.ReadLine()!;

int repeatCount = 1;

server.Start(message, repeatCount);

Console.ReadKey();

}

}

}

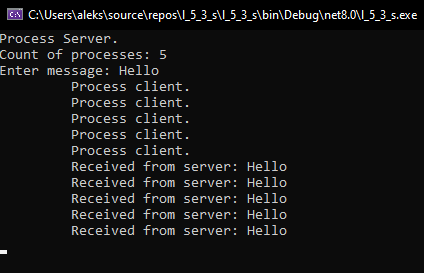


Рис. 1 – Результат програми

4. Виконати свій варіант по списку підгрупи в проекті C# Windows

Forms, використовуючи іменовані канали:

Варіант 5

Написати програму. У вікні серверного процесу по таймеру фону

задається випадковий колір. У вікні клієнтського процесу виводиться RGB-код цього кольору.

**MainForm.cs:**  
using System;

using System.Drawing;

using System.IO;

using System.IO.Pipes;

using System.Windows.Forms;

using System.Threading.Tasks;

public class ServerForm : Form

{

private System.Windows.Forms.Timer timer;

private Panel panel;

private NamedPipeServerStream pipeServer;

public ServerForm()

{

timer = new System.Windows.Forms.Timer();

panel = new Panel { Dock = DockStyle.Fill };

Controls.Add(panel);

timer.Tick += OnTick!;

timer.Interval = 1000;

timer.Start();

pipeServer = new NamedPipeServerStream("testpipe", PipeDirection.Out);

Task.Run(() => pipeServer.WaitForConnection());

}

private void OnTick(object sender, EventArgs e)

{

var random = new Random();

var color = Color.FromArgb(random.Next(256), random.Next(256), random.Next(256));

panel.BackColor = color;

var colorString = $"{color.R},{color.G},{color.B}";

var sw = new StreamWriter(pipeServer);

sw.AutoFlush = true;

sw.WriteLine(colorString);

}

}

public class ClientForm : Form

{

private Label label;

private NamedPipeClientStream pipeClient;

public ClientForm()

{

label = new Label { Dock = DockStyle.Fill, AutoSize = false, TextAlign = ContentAlignment.MiddleCenter };

Controls.Add(label);

pipeClient = new NamedPipeClientStream(".", "testpipe", PipeDirection.In);

pipeClient.Connect();

\_ = Task.Run(() =>

{

var sr = new StreamReader(pipeClient);

while (true)

{

var colorString = sr.ReadLine();

var rgb = colorString.Split(',');

var color = Color.FromArgb(int.Parse(rgb[0]), int.Parse(rgb[1]), int.Parse(rgb[2]));

UpdateColor(color);

}

});

}

private void UpdateColor(Color color)

{

if (label.InvokeRequired)

{

label.Invoke(new Action(() => label.Text = $"R: {color.R}, G: {color.G}, B: {color.B}"));

}

else

{

label.Text = $"R: {color.R}, G: {color.G}, B: {color.B}";

}

}

}

**Program.cs:**

static class Program

{

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Thread serverThread = new Thread(() => Application.Run(new ServerForm()));

serverThread.Start();

Thread clientThread = new Thread(() => Application.Run(new ClientForm()));

clientThread.Start();

}

}

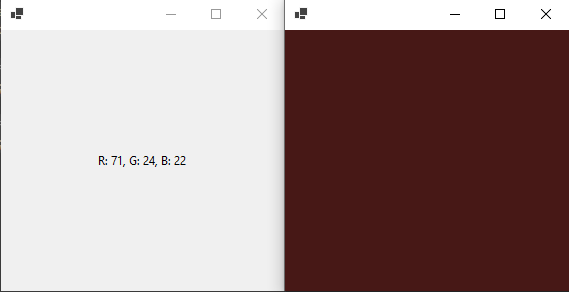


Рис. 2 – Перший результат програми

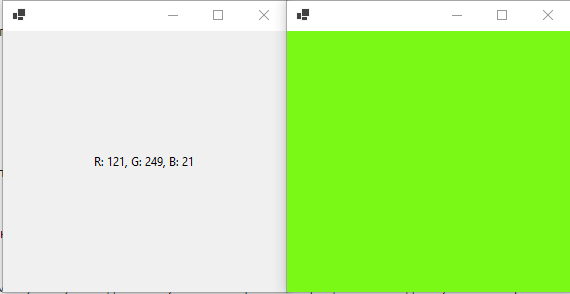


Рис. 3 – Другий результат програми

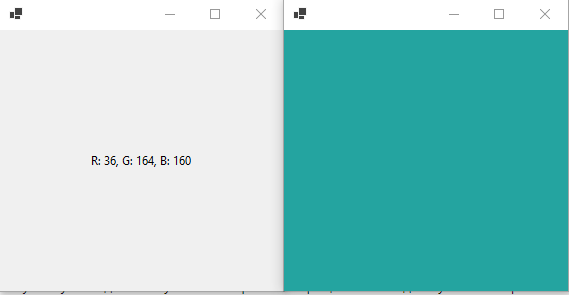


Рис. 4 – Третій результат програми

**Контрольні питання**

1. **В чому полягає різниця між анонімними та іменованими каналами?**

Анонімні канали і іменовані канали використовуються для міжпроцесорного спілкування, але вони відрізняються за декількома параметрами:

* Анонімні канали створюються для одноразового використання і зазвичай використовуються для спілкування між батьківським процесом та його дочірніми процесами.
* Іменовані канали можуть використовуватися багато разів і можуть служити для спілкування між різними процесами на одному комп’ютері або між процесами на різних комп’ютерах в мережі.

1. **Об’єкти яких класів використовуються для організації анонімних каналів?**

Для організації анонімних каналів використовуються об’єкти класів AnonymousPipeServerStream та AnonymousPipeClientStream.

1. **Яким чином організовується запис і читання інформації в анонімному каналі?**

Запис і читання інформації в анонімному каналі організовується за допомогою методів Write та Read відповідно. Батьківський процес створює анонімний канал і передає його дескриптор дочірньому процесу, після чого обидва процеси можуть використовувати канал для обміну даними.

1. **Де і для чого використовуються іменовані канали?**

**Іменовані канали** використовуються для міжпроцесорного спілкування в різних сценаріях. Вони можуть використовуватися для спілкування між процесами на одному комп’ютері або для спілкування між процесами на різних комп’ютерах в мережі. Іменовані канали також можуть використовуватися для реалізації клієнт-серверних додатків.

1. **Яку адресу необхідно вказати при створенні з’єднання за допомогою іменованих каналів?**

При створенні з’єднання за допомогою іменованих каналів необхідно вказати ім’я каналу. Ім’я каналу має бути унікальним в межах комп’ютера і може включати в себе шлях до каналу. Наприклад, ім’я може бути у форматі \\.\pipe\MyNamedPipe.

**Висновок:** Під час виконання лабораторної роботи було розглянуто застосування анонімного та іменованого каналів, розглянуто різницю при їх використанні.