**ЛАБОРАТОРНА РОБОТА № 1**

**Тема:** «Процеси та потоки. Взаємодія між процесами. Розподіл даних між процесами. Робота з файлами які відображуються у пам'ять»

**Хід роботи:**

**Завдання №1:** Необхідно написати дві програми (три), які будуть мати спільні дані та одночасно до них звертатися;

**Програма №1** **"Генерація даних" (консольна):**

**Лістинг коду:**

using System;

using System.IO;

class Program

{

static void Main()

{

//get file path

var basePath = Directory.GetParent(Directory.GetCurrentDirectory()).Parent.Parent.Parent.FullName;

var dataFilePath = Path.Combine(basePath, "data.dat");

Console.WriteLine("Шлях до файлу: " + dataFilePath);

var byteArray = new byte[25];

var rng = new Random();

Console.ForegroundColor = ConsoleColor.Yellow;

Console.WriteLine("Генерація випадкових чисел:");

Console.ResetColor();

//arr gen

for (int i = 0; i < byteArray.Length; i++)

{

byteArray[i] = (byte)rng.Next(1, 100);

Console.Write($"{byteArray[i]} ");

}

Console.WriteLine();

//write

using (var fileStream = new FileStream(dataFilePath, FileMode.OpenOrCreate))

{

fileStream.Write(byteArray, 0, byteArray.Length);

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine("Робота завершена");

Console.ResetColor();

}

}

}

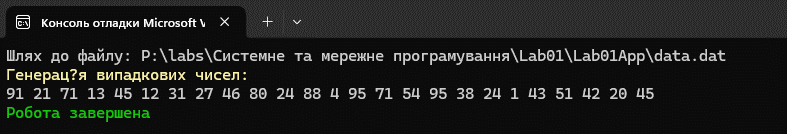


Рис. 1. Результат виконання

**Програма №1** **"Сортування даних" (консольна):**

**Лістинг коду:**

using System;

using System.IO;

using System.IO.MemoryMappedFiles;

using System.Threading;

class Program

{

static void Main()

{

//get file path

var filePath = Path.Combine(Directory.GetParent(Directory.GetCurrentDirectory()).Parent.Parent.Parent.FullName, "data.dat");

//wait for Enter

Console.WriteLine("Натисніть Enter, щоб почати");

while (true)

{

if (Console.ReadKey().Key == ConsoleKey.Enter)

break;

}

using (var mmf = MemoryMappedFile.CreateFromFile(filePath, FileMode.Open, "mappedFile"))

{

using (Mutex mutex = new Mutex())

using (var accessor = mmf.CreateViewAccessor(0, 25))

{

var isSorted = false;

do

{

isSorted = false;

for (int i = 0; i < 24; i++)

{

byte currentByte, nextByte;

accessor.Read(i, out currentByte);

accessor.Read(i + 1, out nextByte);

if (currentByte > nextByte)

{

accessor.Write(i, nextByte);

accessor.Write(i + 1, currentByte);

isSorted = true;

}

}

//pause between iterations

Thread.Sleep(1000);

Console.WriteLine("Пауза між ітераціями..");

} while (!isSorted);

}

}

}

}

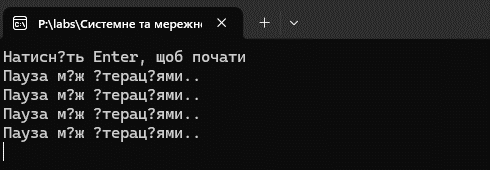


Рис. 2. Результат виконання

**Програма №2** **"Виведення даних" (віконна):**

**Лістинг коду:**

using System;

using System.IO;

using System.IO.MemoryMappedFiles;

using System.Threading;

class Program

{

static void Main()

{

//get file path

var filePath = Path.Combine(Directory.GetParent(Directory.GetCurrentDirectory()).Parent.Parent.Parent.FullName, "data.dat");

//wait for Enter

Console.WriteLine("Натисніть Enter, щоб почати");

while (true)

{

if (Console.ReadKey().Key == ConsoleKey.Enter)

break;

}

using (var mmf = MemoryMappedFile.CreateFromFile(filePath, FileMode.Open, "mappedFile"))

{

using (Mutex mutex = new Mutex())

using (var accessor = mmf.CreateViewAccessor(0, 25))

{

var isSorted = false;

do

{

isSorted = false;

for (int i = 0; i < 24; i++)

{

byte currentByte, nextByte;

accessor.Read(i, out currentByte);

accessor.Read(i + 1, out nextByte);

if (currentByte > nextByte)

{

accessor.Write(i, nextByte);

accessor.Write(i + 1, currentByte);

isSorted = true;

}

}

//pause between iterations

Thread.Sleep(1000);

Console.WriteLine("Пауза між ітераціями..");

} while (!isSorted);

}

}

}

}

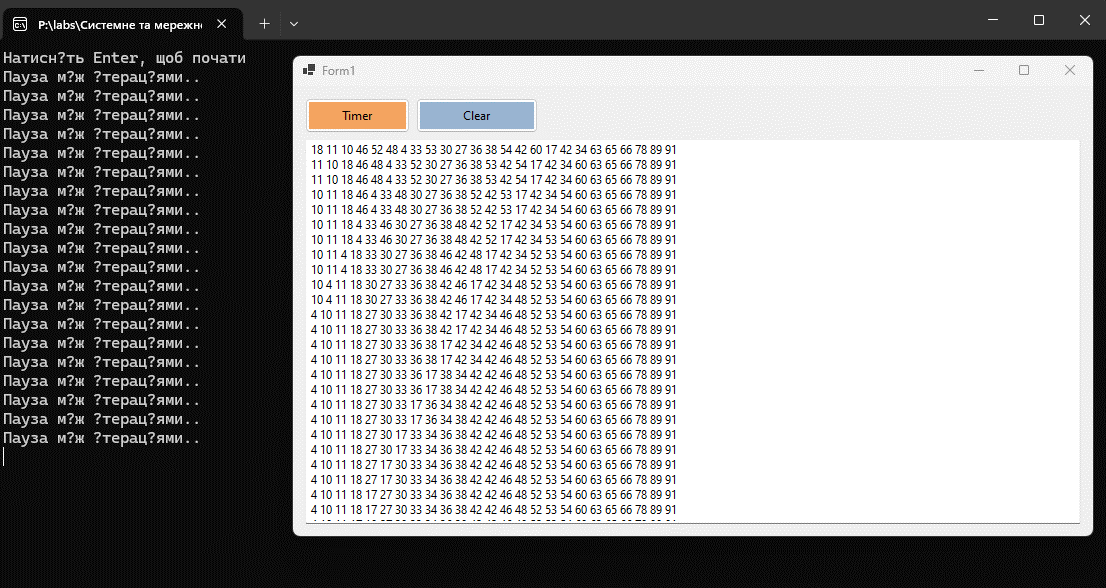


Рис. 3. Результат виконання

**Завдання №2:** Для коректної роботи зі спільними даними у цих двох програмах потрібно додати синхронізацію потоків, які можуть одночасно звертатися до спільних даних.

**Програма №1** **"Сортування даних" (консольна):**

**Лістинг коду:**

class Program

{

static void Main()

{

bool isError = false;

//get file path

var filePath = Path.Combine(Directory.GetParent(Directory.GetCurrentDirectory()).Parent.Parent.Parent.FullName, "data.dat");

//wait for Enter

Console.WriteLine("Натисніть Enter, щоб почати");

while (true)

{

if (Console.ReadKey().Key == ConsoleKey.Enter)

break;

}

using (var mmf = MemoryMappedFile.CreateFromFile(filePath, FileMode.Open, "mappedFile"))

{

using (var accessor = mmf.CreateViewAccessor(0, 25))

using (Mutex mutex = new Mutex())

{

do

{

try

{

var isSorted = false;

do

{

isSorted = false;

for (int i = 0; i < 24; i++)

{

mutex.WaitOne();

byte currentByte, nextByte;

accessor.Read(i, out currentByte);

accessor.Read(i + 1, out nextByte);

if (currentByte > nextByte)

{

accessor.Write(i, nextByte);

accessor.Write(i + 1, currentByte);

isSorted = true;

}

}

//pause between iterations

Thread.Sleep(1500);

Console.WriteLine("Пауза між ітераціями..");

mutex.ReleaseMutex();

} while (!isSorted);

}

catch (Exception e)

{

isError = true;

}

} while (isError);

}

}

}

}

**Програма №2** **"Виведення даних" (віконна):**

using System.IO.MemoryMappedFiles;

using System.Text;

namespace Task1\_WinApp

{

public partial class Form1 : Form

{

bool isOpenedMappedFileOnce;

Mutex mutex;

public Form1()

{

InitializeComponent();

isOpenedMappedFileOnce = false;

mutex = new();

}

private void timer\_btn\_Click(object sender, EventArgs e)

{

timer1.Enabled = !timer1.Enabled;

}

private void clear\_btn\_Click(object sender, EventArgs e)

{

OutputTextBox.Text = "";

mutex.Dispose();

}

private void timer1\_Tick(object sender, EventArgs e)

{

try

{

using (var file = MemoryMappedFile.OpenExisting("mappedFile"))

{

using (var reader = file.CreateViewAccessor(0, 25))

{

mutex.WaitOne();

var bytes = new byte[25];

reader.ReadArray<byte>(0, bytes, 0, bytes.Length);

StringBuilder sb = new StringBuilder();

for (var i = 0; i < bytes.Length; i++)

{

sb.Append($"{bytes[i]} ");

}

OutputTextBox.Text += sb.ToString() + Environment.NewLine;

mutex.ReleaseMutex();

}

}

}

catch

{

if (!isOpenedMappedFileOnce)

{

isOpenedMappedFileOnce = true;

OutputTextBox.Text = "";

OutputTextBox.AppendText("Не отримали дані з файлу. Переконайтесь, що консольний додаток запущений" + Environment.NewLine);

}

}

}

}

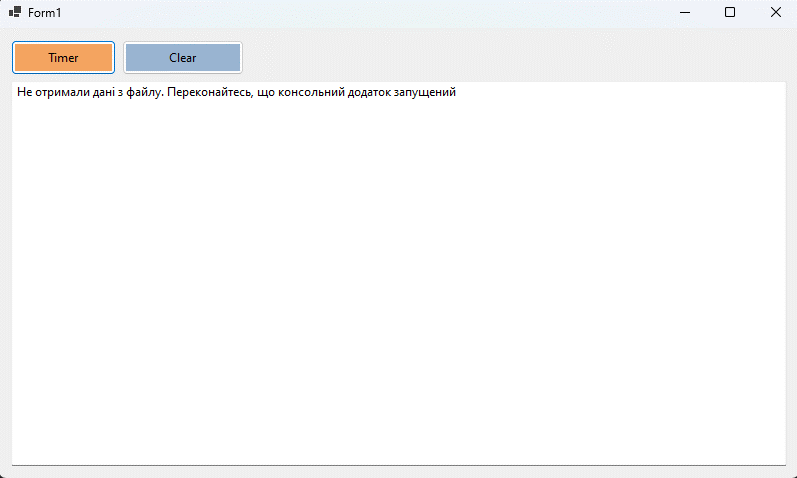
} 

Рис. 4. Повідомлення про помилку