

Лабораторна робота № 3

Тема: Багатопотоковість у C#.

Мета роботи: навчитися працювати з потоками та процесами у мові C#.

Хід роботи

TASK-01

Вимоги до програми шифрування файлів

1. Графічний інтерфейс користувача з можливістю вводу ключа для шифрування та діалогу вибору довільного файлу, який буде шифруватись або розшифровуватись.
2. Здатність працювати з файлами довільного розміру та формату.
3. В процесі шифрування, повинен відображатись індикатор прогресу (0-100%) та час, який пройшов від запуску шифрування.
4. По завершенні шифрування повинне бути відображене вікно з інформацією про розмір зашифрованого файлу, його назву та час, затрачений на шифрування.
5. Необхідно запобігти «підвисанню» вікна при здійсненні операції шифрування (шифрування здійснюватись в окремому обчислювальному потоці за допомогою класу BackgroundWorker) .
6. Коректна обробка виключень, що можуть виникати під час роботи.
7. Для успішного виконання роботи необхідно ознайомитися з наступними темами: асинхронні операції (клас BackgroundWorker), класи File, FileStream, Timer, DateTime, OpenFileDialog, SaveFileDialog, обробка виключень. реалізованих методів розширення.

					ДУ «Житомирська політехніка».23.121.17.000–Лр3			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Леус В.О.			Звіт з лабораторної роботи		Літ.	Арк.
Перевір.		Чижмотря О.В.						1
Керівник							Аркушів	
Н. контр.							10	
Зав. каф.							ФІКТ Гр. ІПЗ-22-3	

Coder-Uncoder:

```
using System;
using System.ComponentModel;
using System.Diagnostics;
using System.IO;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Windows.Forms;
namespace Coder_Uncoder
{
    public partial class TASK : Form
    {
        private BackgroundWorker encryptionWorker;
        private BackgroundWorker decryptionWorker;
        private long encryptedFileSize;
        private long decryptionStartTime;
        private Stopwatch stopwatch = new Stopwatch();
        public TASK()
        {
            InitializeComponent();

            encryptionWorker = new BackgroundWorker();
            encryptionWorker.DoWork += EncryptionWorker_DoWork;
            encryptionWorker.ProgressChanged += EncryptionWorker_ProgressChanged;
            encryptionWorker.RunWorkerCompleted += EncryptionWorker_RunWorkerCompleted;
            encryptionWorker.WorkerReportsProgress = true;

            decryptionWorker = new BackgroundWorker();
            decryptionWorker.DoWork += DecryptionWorker_DoWork;
            decryptionWorker.RunWorkerCompleted += DecryptionWorker_RunWorkerCompleted;
            decryptionWorker.WorkerReportsProgress = true;
        }
        private void FileCodBTN_Click(object sender, EventArgs e)
        {
            OpenFileDialog openFileDialog = new OpenFileDialog();
            if (openFileDialog.ShowDialog() == DialogResult.OK)
            {
                FileCodTB.Text = openFileDialog.FileName;
            }
        }
        private void KatCodBTN_Click(object sender, EventArgs e)
        {
            SaveFileDialog saveFileDialog = new SaveFileDialog();
            if (saveFileDialog.ShowDialog() == DialogResult.OK)
            {
                KatCodTB.Text = saveFileDialog.FileName;
            }
        }
        private void CodBNT_Click(object sender, EventArgs e)
        {
            if (string.IsNullOrEmpty(FileCodTB.Text) ||
                string.IsNullOrEmpty(KatCodTB.Text) || string.IsNullOrEmpty(KeyCodTB.Text))
            {
                MessageBox.Show("Будь ласка, заповніть всі поля.");
                return;
            }

            string inputFile = FileCodTB.Text;
            string outputFile = KatCodTB.Text;
            string key = KeyCodTB.Text;

            if (inputFile == outputFile)
```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 – ЛрЗ	Арк.
		Чижмоторя О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

        {
            MessageBox.Show("Введення та виведення файлу не можуть бути однаковими.  
Будь ласка, оберіть інший файл для виведення.");
            return;
        }

        encryptedFileSize = new FileInfo(inputFile).Length;

        decryptionStartTime = DateTime.Now.Ticks;
        encryptionWorker.RunWorkerAsync(new EncryptionData(inputFile, outputFile,
key));
    }
    private void EncryptionWorker_DoWork(object sender, DoWorkEventArgs e)
    {
        EncryptionData data = (EncryptionData)e.Argument;
        using (FileStream inputFileStream = new FileStream(data.InputFile,
        FileMode.Open))
        using (FileStream outputFileStream = new FileStream(data.OutputFile,
        FileMode.Create))
        using (Aes aesAlg = Aes.Create())
        {
            aesAlg.Key = Encoding.UTF8.GetBytes(data.Key.PadRight(32,
'0')).Take(32).ToArray();
            aesAlg.Mode = CipherMode.CFB;

            aesAlg.GenerateIV();
            outputFileStream.Write(aesAlg.IV, 0, aesAlg.IV.Length);

            using (CryptoStream csEncrypt = new CryptoStream(outputFileStream,
aesAlg.CreateEncryptor(), CryptoStreamMode.Write))
            {
                byte[] buffer = new byte[1024];
                int bytesRead;
                long totalBytes = 0;
                while ((bytesRead = inputFileStream.Read(buffer, 0, buffer.Length)) >
0)
                {
                    csEncrypt.Write(buffer, 0, bytesRead);
                    totalBytes += bytesRead;
                    int progressPercentage = (int)((totalBytes * 100) / input-
?ileStream.Length);
                    encryptionWorker.ReportProgress(progressPercentage);
                }
            }
        }
        private void EncryptionWorker_ProgressChanged(object sender, Progr-
?ssChangedEventArgs e)
        {
            progressBarCod.Value = e.ProgressPercentage;
        }
        private void EncryptionWorker_RunWorkerCompleted(object sender, RunWo-
?kerCompletedEventArgs e)
        {
            if (e.Error != null)
            {
                MessageBox.Show($"Помилка: {e.Error.Message}");
            }
            else
            {
                string encryptedFileName = Path.GetFileName(FileCodTB.Text);
                string encryptedText = ReadFirst10WordsFromFile(KatCodTB.Text, 25);

                long encryptionEndTime = DateTime.Now.Ticks;

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 – ПрЗ	Арк.
		Чижемотря О.В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        TimeSpan encryptionTime = new TimeSpan(encryptionEndTime - decryptionStartTime);

        MessageBox.Show($"Файл '{encryptedFileName}' успішно зашифровано.\nПерші 10 слів тексту: {encryptedText}\nЧас шифрування: {encryptionTime.TotalMilliseconds} мс\nРозмір файлу: {encryptedFileSize} байт");
    }

    private void FileUncodBTN_Click(object sender, EventArgs e)
    {
        OpenFileDialog openFileDialog = new OpenFileDialog();
        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            FileUncodTB.Text = openFileDialog.FileName;
        }
    }

    private void KatUncodBTN_Click(object sender, EventArgs e)
    {
        SaveFileDialog saveFileDialog = new SaveFileDialog();
        if (saveFileDialog.ShowDialog() == DialogResult.OK)
        {
            KatUncodTB.Text = saveFileDialog.FileName;
        }
    }

    private void UncodBTN_Click(object sender, EventArgs e)
    {
        if (string.IsNullOrEmpty(FileUncodTB.Text) || string.IsNullOrEmpty(KatUncodTB.Text) || string.IsNullOrEmpty(KeyUncodTB.Text))
        {
            MessageBox.Show("Будь ласка, заповніть всі поля.");
            return;
        }

        string inputFile = FileUncodTB.Text;
        string outputFile = KatUncodTB.Text;
        string key = KeyUncodTB.Text;
        if (inputFile == outputFile)
        {
            MessageBox.Show("Введення та виведення файлу не можуть бути однаковими. Будь ласка, оберіть інший файл для виведення.");
            return;
        }

        encryptedFileSize = new FileInfo(inputFile).Length;
        decryptionStartTime = DateTime.Now.Ticks;
        decryptionWorker.RunWorkerAsync(new DecryptionData(inputFile, outputFile, key));
    }

    private void DecryptionWorker_DoWork(object sender, DoWorkEventArgs e)
    {
        DecryptionData data = (DecryptionData)e.Argument;
        using (FileStream inputStream = new FileStream(data.InputFile, FileMode.Open))
        using (FileStream outputStream = new FileStream(data.OutputFile, FileMode.Create))
        using (Aes aesAlg = Aes.Create())
        {
            aesAlg.Key = Encoding.UTF8.GetBytes(data.Key.PadRight(32, '0')).Take(32).ToArray();
            aesAlg.Mode = CipherMode.CFB;
            byte[] iv = new byte[aesAlg.BlockSize / 8];
            inputStream.Read(iv, 0, iv.Length);
            aesAlg.IV = iv;

            using (CryptoStream csDecrypt = new CryptoStream(outputFileStream, aesAlg.CreateDecryptor(), CryptoStreamMode.Write))
            {

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 – ПрЗ	Арк.
		Чижемотря О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

        byte[] buffer = new byte[1024];
        int bytesRead;
        while ((bytesRead = inputFileStream.Read(buffer, 0, buffer.Length)) >
0)
        {
            csDecrypt.Write(buffer, 0, bytesRead);
        }
    }
}
private void DecryptionWorker_RunWorkerCompleted(object sender, RunWo-
?kerCompletedEventArgs e)
{
    if (e.Error != null)
    {
        MessageBox.Show($"Помилка: {e.Error.Message}");
    }
    else
    {
        string decryptedFileName = Path.GetFileName(FileUncodTB.Text);
        string decryptedText = ReadFirst10WordsFromFile(KatUncodTB.Text, 25);

        long decryptionEndTime = DateTime.Now.Ticks;
        TimeSpan decryptionTime = new TimeSpan(decryptionEndTime - decry-
?tionStartTime);

        FileInfo decryptedFile = new FileInfo(FileUncodTB.Text);

        MessageBox.Show($"Файл '{decryptedFileName}' успішно розшифровано.\nПерші
10 слів тексту: {decryptedText}\nЧас розшифровки: {decryptionTime.TotalMilliseconds}
мс\nРозмір файлу: {decryptedFile.Length} байт");
    }
}
private string ReadFirst10WordsFromFile(string filePath, int maxLength)
{
    string text = string.Empty;
    try
    {
        using (StreamReader reader = new StreamReader(filePath))
        {
            text = reader.ReadToEnd();
        }
    }
    catch (IOException ex)
    {
        Console.WriteLine($"Помилка читання файлу: {ex.Message}");
    }
    if (text.Length > maxLength)
    {
        text = text.Substring(0, maxLength) + "...";
    }
    return text;
}
private class EncryptionData
{
    public string InputFile { get; }
    public string OutputFile { get; }
    public string Key { get; }
    public EncryptionData(string inputFile, string outputFile, string key)
    {
        InputFile = inputFile;
        OutputFile = outputFile;
        Key = key;
    }
}

```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 – ЛрЗ	Арк.
		Чижмоторя О.В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

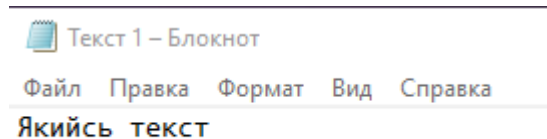
```

private class DecryptionData
{
    public string InputFile { get; }
    public string OutputFile { get; }
    public string Key { get; }
    public DecryptionData(string inputFile, string outputFile, string key)
    {
        InputFile = inputFile;
        OutputFile = outputFile;
        Key = key;
    }
}

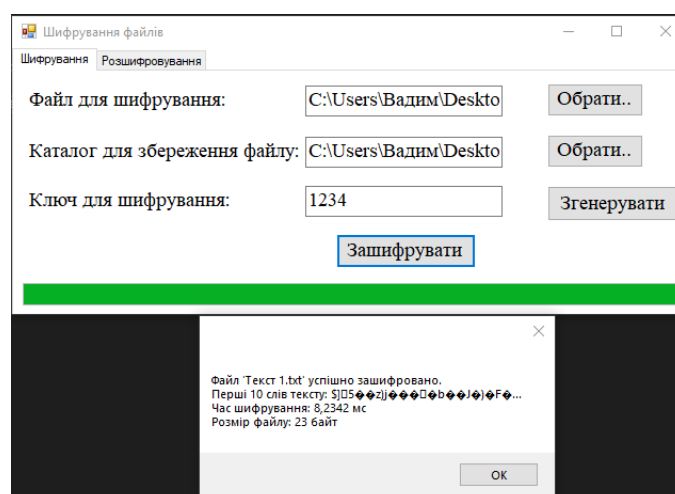
private void GenerationCodBTN_Click(object sender, EventArgs e)
{
    using (RNGCryptoServiceProvider rng = new RNGCryptoServiceProvider())
    {
        byte[] keyBytes = new byte[32];
        rng.GetBytes(keyBytes);
        KeyCodTB.Text = Convert.ToBase64String(keyBytes);
    }
}

private void KeyUncodBTN_Click(object sender, EventArgs e)
{
    using (RNGCryptoServiceProvider rng = new RNGCryptoServiceProvider())
    {
        byte[] keyBytes = new byte[32];
        rng.GetBytes(keyBytes);
        KeyUncodTB.Text = Convert.ToBase64String(keyBytes);
    }
}
}

```

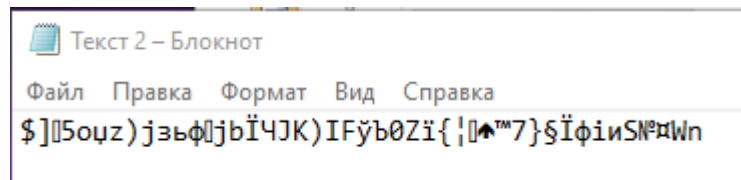


Мал.2. Вміст 1 файлу

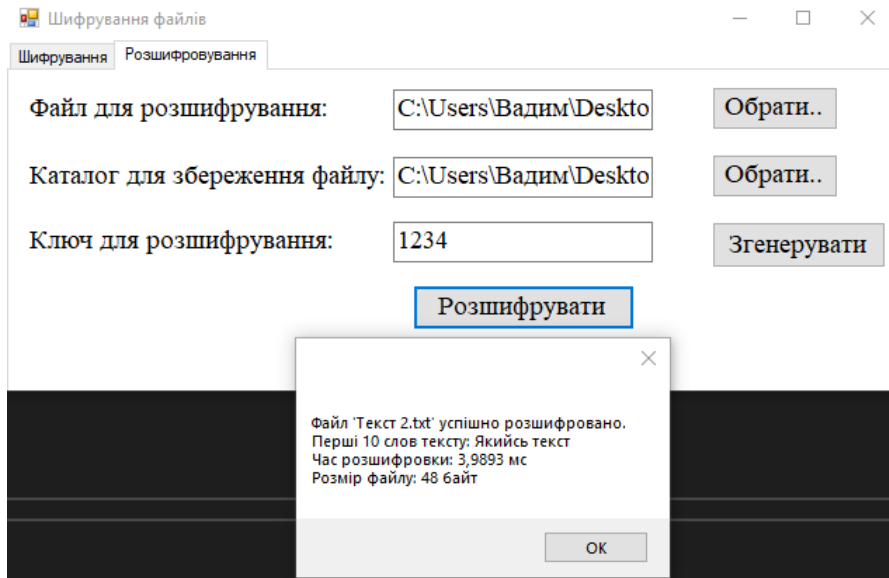


Мал.2. Шифрування файлу

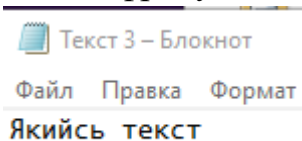
		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 – Лр3	Арк.
		Чижмоторя О.В.				6
Змн.	Арк.	№ докум.	Підпис	Дата		



Мал.3. Вміст 2 файлу



Мал.4. Розшифровування файлу



Мал.5. Вміст 3 файлу

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 – Лр3	Арк.
		Чижмотря О.В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

TASK-02

Вимоги до менеджера процесів

1. Можливість запуску програм: Калькулятор, Microsoft Word та ще трьох на ваш вибір.
2. Перегляд список запущених процесів з інформацією про обсяг оперативної пам'яті, виділений під процес, часу, коли був запущений процес, пріоритет процесу, кількість потоків, які запущені процесом. Відобразити у вигляді таблиці.
3. Можливість зупинки вибраного процесу.
4. Можливість зміни пріоритету вибраного процесу.

ProcessManager:

```
using System;
using System.Diagnostics;
using System.Windows.Forms;
namespace ProcessManager
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            InitializeDataGridView();
        }
        private void InitializeDataGridView()
        {
            dataGridViewProcesses.ColumnCount = 5;
            dataGridViewProcesses.Columns[0].Name = "ProcessName";
            dataGridViewProcesses.Columns[1].Name = "MemoryUsage";
            dataGridViewProcesses.Columns[2].Name = "StartTime";
            dataGridViewProcesses.Columns[3].Name = "Priority";
            dataGridViewProcesses.Columns[4].Name = "ThreadCount";
        }
        private void NotebookBTN_Click(object sender, EventArgs e)
        {
            StartProcess("notepad.exe");
        }
        private void CalculatorBTN_Click(object sender, EventArgs e)
        {
            StartProcess("calc.exe");
        }
        private void ConductorBTN_Click(object sender, EventArgs e)
        {
            StartProcess("explorer.exe");
        }
        private void WordBTN_Click(object sender, EventArgs e)
        {
            StartProcess("winword.exe");
        }
        private void ExcelBTN_Click(object sender, EventArgs e)
        {
            StartProcess("excel.exe");
        }
    }
}
```

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 – Лр3	Арк.
		Чижмоторя О.В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		


```

private void PowerPointBTN_Click(object sender, EventArgs e)
{
    StartProcess("powerpnt.exe");
}
private void RefreshProcessesList()
{
    dataGridViewProcesses.Rows.Clear();
    Process[] processes = Process.GetProcesses();

    foreach (Process process in processes)
    {
        string name = process.ProcessName;
        long memoryUsage = process.WorkingSet64;
        DateTime startTime = process.StartTime;
        int priority = process.BasePriority;
        int threadCount = process.Threads.Count;

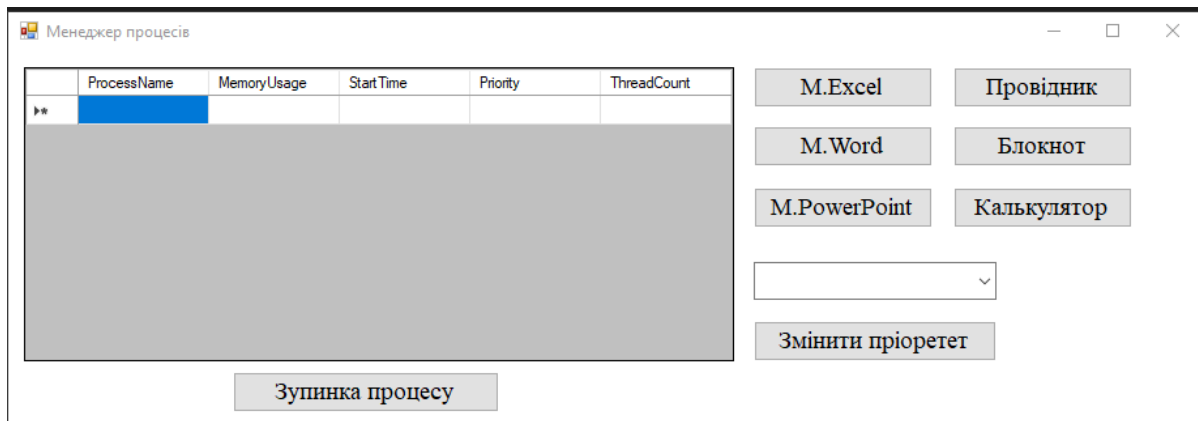
        dataGridViewProcesses.Rows.Add(name, memoryUsage, startTime, priority,
threadCount);
    }
}
private void StartProcess(string processName)
{
    try
    {
        Process.Start(processName);
        RefreshProcessesList();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Помилка запуску процесу: " + ex.Message);
    }
}
private void StopProsBTN_Click(object sender, EventArgs e)
{
    if (dataGridViewProcesses.SelectedRows.Count > 0)
    {
        DataGridViewRow selectedRow = dataGridViewProcesses.SelectedRows[0];
        string processName = selectedRow.Cells[0].Value.ToString();

        KillProcessByName(processName);
    }
}
private void KillProcessByName(string processName)
{
    Process[] processes = Process.GetProcessesByName(processName);

    foreach (Process process in processes)
    {
        process.CloseMainWindow();
    }
    RefreshProcessesList();
}
}
}

```

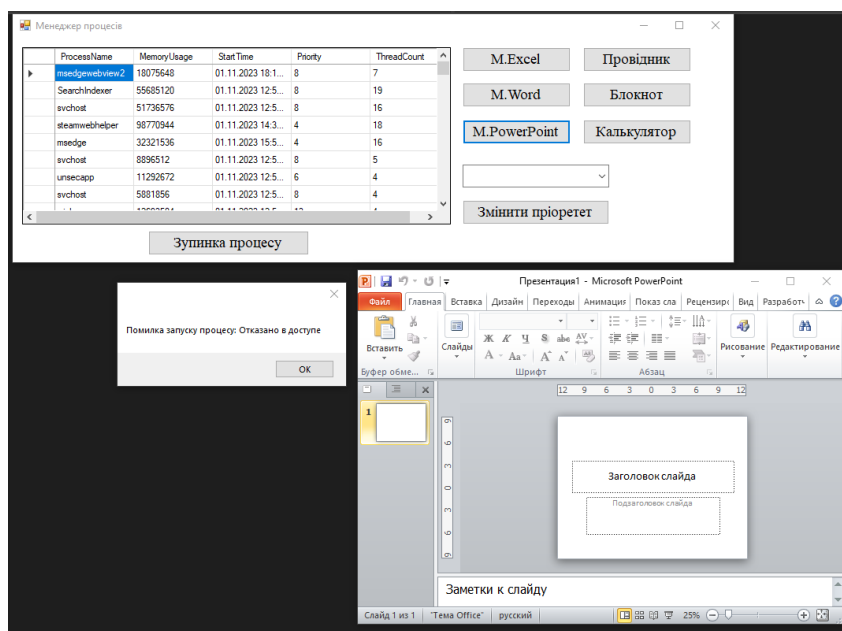
		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 – Лр3	Арк.
		Чижевотря О.В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		



Мал.6. Реалізація програми 2

Після запуску будь-якого додатку з'являється помилка, але додаток запускається, через помилку не можливо зупинити процеси чи змінити пріоритет.

Помилка пов'язана з тим, щоб редагувати процеси потрібні права адміністратора, хоч я і надав їх, але помилка нікуди не ділась, я не знаю як вирішити цю проблему.



Мал.7. Приклад роботи програми 2

Висновки: я навчився використовувати багатопотоковість у C#.

Реалізовано програмний код, який доступний за посиланням:
<https://github.com/VadymLeus/DotNetLab3>

		Леус В.О.			ДУ «Житомирська політехніка».23.121.17.000 – ЛрЗ	Арк.
		Чижмоторя О.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		