Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра кібербезпеки та програмного забезпечення

**Звіт**

з виконаної лабораторної роботи № 4

Дисципліна: Системне програмування

на тему

«Взаємодія між процесами IPC (Частина 2)»

Виконав :

студент академічної групи КІ-15

Аннаєв А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перевірив :

Викладач

КонстантиноваЛ.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кропивницкий- 2018

**Мета:** Отримати практичні навики в використанні функцій Win API. Навчитися організовувати коректну взаємодію між процесами застосовуючи механізми:

– Іменований канал (named pipe);

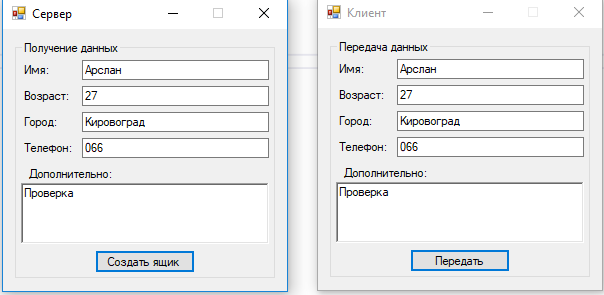
– Поштові скриньки (mailslots).

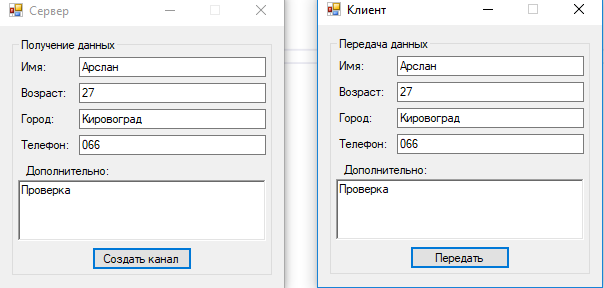
**Завдання:**

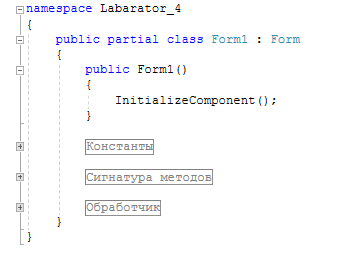
Використовуючи наявну електронну документацію повністю повторити функціонал програм наведених в прикладах на мові С++ чи С#.

Тобто провести перетворення програм (трансляцію), представлених на мові програмування Object Pascal на мову С++ чи С#.

IPC\_MAILSLOT



ICP\_PIPES

IPC\_MAILSLOT\_MASTER

#region Константы

const int MAILSLOT\_WAIT\_FOREVER = -1;

#endregion

#region Сигнатура методов

/// <summary>

/// Создание

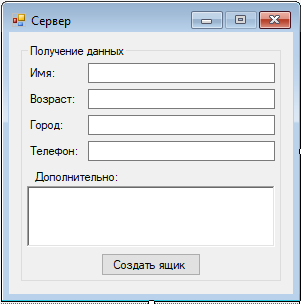
/// </summary>

[DllImport("Kernel32.dll")]

static extern IntPtr CreateMailslot(string name,uint nMax,int time, IntPtr lpSec);

/// <summary>

/// Считывание файла

 /// </summary>

[DllImport("Kernel32.dll")]

private static extern bool ReadFile(IntPtr handle,byte[] buffer,uint toRead,ref uint read,IntPtr lpOver);

/// <summary>

/// Закрытие дескриптора

/// </summary>

[DllImport("kernel32.dll")]

private static extern bool CloseHandle(IntPtr handle);

#endregion

#region Обработчик

/// <summary>

/// Обработчик нажатия кнопки

/// </summary>

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

{ // создание ящика и чтение сообщения

uint size = 1024;

IntPtr hMail = CreateMailslot(@"\\.\mailslot\MyMailSlot", 0,

MAILSLOT\_WAIT\_FOREVER, IntPtr.Zero);

if (hMail == (IntPtr)(-1)){

MessageBox.Show("Ошибка при создании ящика.");

return;

}

byte[] buffer = new byte[size];

ReadFile(hMail, buffer, size, ref size, IntPtr.Zero);

string data = Encoding.Default.GetString(buffer);

txtName.Text = data.Split('|')[0];

txtAge.Text = data.Split('|')[1];

txtCity.Text = data.Split('|')[2];

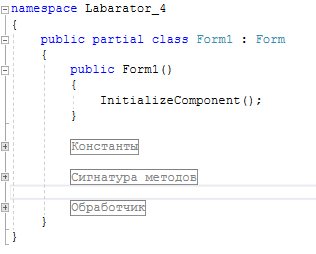
txtPhone.Text = data.Split('|')[3];

txtAuxiliary.Text = data.Split('|')[4];

CloseHandle(hMail);

}

#endregion

IPC\_MAILSLOT\_SLAVE

#region Константы

const int GENERIC\_WRITE = 0x40000000;

const int FILE\_SHARE\_WRITE = 0x00000002;

const int OPEN\_EXISTING = 3;

#endregion

#region Сигнатура методов

/// <summary>

/// Создание файла

/// </summary>

[DllImport("Kernel32.dll")]

static extern IntPtr CreateFile(String lpFN, int dwDdAc, int dwShM,

IntPtr lpSecAt, uint dwCDis, uint dwFlags, IntPtr hTempFile);

/// <summary>

/// Запись в файл

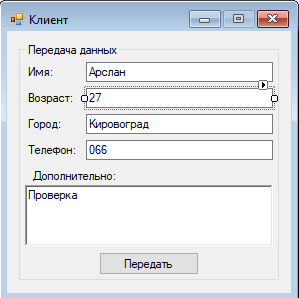
/// </summary>

[DllImport("Kernel32.dll")]

static extern bool WriteFile(IntPtr handle, byte[] buffer, int count,

ref int written, IntPtr lpOverlapped);

/// <summary>

 /// Сброс буффера

/// </summary>

[DllImport("Kernel32.dll")]

static extern bool FlushFileBuffers(IntPtr handle);

/// <summary>

/// Закрытие дескриптора

/// </summary>

[DllImport("Kernel32.dll")]

static extern bool CloseHandle(IntPtr handle);

#endregion

#region Обработчик

/// <summary>

/// Обработчик нажатия кнопки

/// </summary>

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

{

string data = txtName.Text + "|" + txtAge.Text + "|" +

txtCity.Text + "|" + txtPhone.Text + "|" + txtAuxiliary.Text;

int size = data.Length;

IntPtr hMail = CreateFile(@"\\.\mailslot\MyMailSlot",

GENERIC\_WRITE, FILE\_SHARE\_WRITE,

IntPtr.Zero, OPEN\_EXISTING, 0, IntPtr.Zero);

if (hMail == (IntPtr)(-1)) {

MessageBox.Show("Ошибка при подключении.");

return; }

WriteFile(hMail, Encoding.Default.GetBytes(data), size, ref size, IntPtr.Zero);

FlushFileBuffers(hMail);

CloseHandle(hMail);

}

#endregion

IPC\_PIPES\_MASTER

#region Константы

const uint PIPE\_ACCESS\_DUPLEX = 0x00000003;

#endregion

#region Сигнатура методов

/// <summary>

/// Создание

/// </summary>

[DllImport("Kernel32.dll")]

private static extern IntPtr CreateNamedPipe(String т, uint dw, uint dwP, uint nMax,uint nOut, uint nIn, uint nDef, IntPtr pSec);

/// <summary>

/// Подключение

/// </summary>

[DllImport("Kernel32.dll")]

private static extern bool ConnectNamedPipe(IntPtr hHandle, IntPtr lpOverlapped);

/// <summary>

/// ОТключение

/// </summary>

[DllImport("Kernel32.dll")]

private static extern bool DisconnectNamedPipe(IntPtr hHandle);

/// <summary>

/// Считывание файла

/// </summary>

[DllImport("Kernel32.dll")]

private static extern bool ReadFile(IntPtr handle, byte[] buffer, uint toRead, ref uint read, IntPtr lpOverLapped);

/// <summary>

/// Обновление буффера

/// </summary>

[DllImport("Kernel32.dll")]

private static extern bool FlushFileBuffers(IntPtr handle);

/// <summary>

/// Закрытие дескриптора

/// </summary>

[DllImport("kernel32.dll")]

private static extern bool CloseHandle(IntPtr handle);

#endregion

#region Обработчик

/// <summary>

/// Обработчик нажатия кнопки

/// </summary>

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

{

uint size = 1024;

IntPtr hPipe = CreateNamedPipe(@"\\.\pipe\ThePipe",

PIPE\_ACCESS\_DUPLEX, 0, 1, size,

size, 100, IntPtr.Zero);

if (hPipe == IntPtr.Zero){

MessageBox.Show("Ошибка при создании канала.");

return;

}

if (!ConnectNamedPipe(hPipe, IntPtr.Zero)){

MessageBox.Show("Ошибка при открытии канала.");

return;

}

byte[] buffer = new byte[size];

ReadFile(hPipe, buffer, size, ref size, IntPtr.Zero);

string data = Encoding.Default.GetString(buffer);

txtName.Text = data.Split('|')[0];

txtAge.Text = data.Split('|')[1];

txtCity.Text = data.Split('|')[2];

txtPhone.Text = data.Split('|')[3];

txtAuxiliary.Text = data.Split('|')[4];

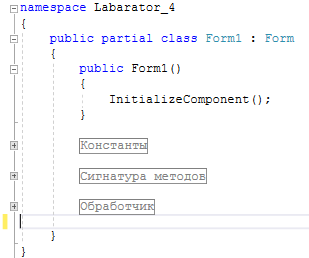
FlushFileBuffers(hPipe);

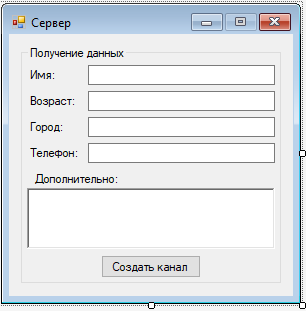
DisconnectNamedPipe(hPipe);

CloseHandle(hPipe);

}

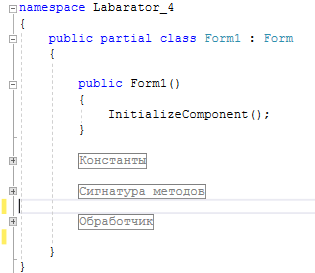
#endregion





IPC\_PIPES\_SLAVE

#region Константы

 const int GENERIC\_WRITE = 0x40000000;

const int FILE\_SHARE\_WRITE = 0x00000002;

const int OPEN\_EXISTING = 3;

#endregion

#region Сигнатура методов

/// <summary>

/// Создание файла

/// </summary>

[DllImport("Kernel32.dll")]

static extern IntPtr CreateFile(String lp,int dwD,int dwS,IntPtr lpS,

uint dwC,uint dwF, IntPtr hT);

/// <summary>

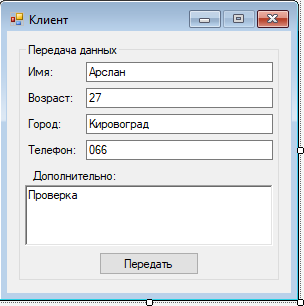
/// Запись в файл

/// </summary>

[DllImport("Kernel32.dll")]

static extern bool WriteFile(IntPtr h, byte[] b, int c, ref int w, IntPtr lpOv);

/// <summary>

 /// Сброс буффера

/// </summary>

[DllImport("Kernel32.dll")]

static extern bool FlushFileBuffers(IntPtr handle);

/// <summary>

/// Закрытие дескриптора

/// </summary>

[DllImport("Kernel32.dll")]

static extern bool CloseHandle(IntPtr handle);

#endregion

#region Обработчик

/// <summary>

/// Обработчик нажатия кнопки

/// </summary>

private void btnSend\_Click(object sender, EventArgs e){

string data = txtName.Text + "|" + txtAge.Text + "|" + txtCity.Text + "|" +

txtPhone.Text + "|" + txtAuxiliary.Text;

int size = data.Length;

IntPtr hPipe = CreateFile(@"\\.\pipe\ThePipe",

GENERIC\_WRITE, FILE\_SHARE\_WRITE,

IntPtr.Zero, OPEN\_EXISTING, 0, IntPtr.Zero);

if (hPipe == (IntPtr)(-1)){

MessageBox.Show("Ошибка при подключении.");

return;

}

WriteFile(hPipe, Encoding.Default.GetBytes(data), size, ref size,

IntPtr.Zero);

FlushFileBuffers(hPipe);

CloseHandle(hPipe);

}

#endregion