برمجة تطبيقات الشبكات باستخدام Visual C# .NET

Part 1

السلام عليكم ورحمة الله وبركاته

بسم الله الرحمن الرحيم

درس بسيط جدا للتعريف بكيفية إنشاء اتصال و إرسال بيانات بين حاسبين موصولين على شبكة محلية من خلال تطبيق برمجي في نظام التشغيل Windows XP

هذا الدرس إهداء إلى طلاب الأمة الإسلامية و العربية وهو برسم الأمانة لا يحق لأحد وضع اسمه عليه أو القيام بإضافة معلومات عليه

ملاحظة

البرنامج المثال مرفق مع الدرس

معلومات عن المؤلف :

البلد : الجمهورية العربية السورية - مدينة دمشق

المرحلة الدراسية : السنة الثالثة بكالوريوس تكنولوجيا المعلومات

الجامعة الافتراضية السورية Syrian Virtual University

E-Mail : munichbayrn@hotmail.com

رحم الله من أهدى إلي عيوبي

أرجوا ممن لديه ملاحظات على هذا الدرس أن يتكرم علي بملاحظاته و نصيحته و له جزيل الشكر

اللهم علمنا ما ينفعنا و انفعنا بما علمتنا و زدنا علما

مقدمة خاصة بالقارئ

عزيزي القارئ

لكي تكون من الذين وجهت لهم هذا الدرس يجب أن يتحقق فيك أمران لا ثالث لهما و هما :

1. **لديك خبرة لا بأس بها في برمجة تطبيقات Windows Application من خلال Visual Studio C# .NET**
2. **لديك إلمام بالمبادئ النظرية للشبكات**

حيث لن أقوم بالاستفاضة في شرح مفاهيم الشبكات النظرية أو البرمجة بلغة C#

مقدمة عامة

في الحقيقة إن برمجة تطبيقات الشبكات من خلال أي لغة أمر كبير جدا ومعقد ،و للذي يريد أن يقوم بإنشاء تطبيقات كبيرة في هذا المجال يحتاج إلى خبرة كبيرة في أنظمة تشغيل الشبكات ، ومعرفة عميقة في جميع المفاهيم الخاصة بالشبكات بالإضافة إلى معرفة ممتازة بالنسبة للغة البرمجة التي يريد بناء التطبيق من خلالها .

إن هذا الدرس الذي أقدمه إليكم لا يشكل شيئا بالنسبة إلى هذا العنوان (برمجة تطبيقات الشبكات) ، إنما هو نظرة في طريق هذا النوع من التطبيقات .

إن الغرض الرئيسي من التطبيقات Windows الخاصة بمجال الشبكات هو عملية إنشاء اتصال و تبادل البيانات و إجراء عمليات التحكم و المراقبة و إلى ما هنالك بين أجهزة الحاسوب المربوطة على الشبكة ، وهذا الكلام ينطبق على الحالتين

شبكة الانترنت – شبكة عمل فردية

فمثلا على شبكة الانترنت ، تحتاج أحيانا إلى تحميل أو رفع ملفات (Download - Upload) من أو إلى موقع معين وهو شكل من أشكال تبادل البيانات .

و أيضا مثل آخر البرمجيات الكثيرة التي تؤمن تبادل البيانات بين الأجهزة الحاسوبية على شبكة فردية .

ولذلك وفرت المكتبة FCL (Framework Class Library) الخاصة ببيئة .NET العديد من الصفوف الجاهزة التي تساعد المطور على تطوير تطبيقات خاصة بمجال الشبكات ، و في هذا الدرس سوف أقوم باستخدام هذه الصفوف بإنشاء تطبيق يقوم بإنشاء اتصال و نقل بيانات من جهاز لآخر على الشبكة وذلك من خلال لغة C# ، فهيا بنا .....

فصل الشرح النظري

إن الكلام الذي سوف تقرأه هنا سوف نقوم بشرحه عمليا ، لذلك أرجو أن تقوم بقراءته إلى النهاية دون ملل ، حيث أنني سوف أقوم بالاختصار قدر الإمكان إن شاء الله .

مبادئ هذا النوع من التطبيقات البرمجية

إن من أهم المبادئ وأوضحها في عملية إنشاء اتصال بين جهازين مربوطين معلى شبكة معينة هي العنونة ، ماذا يعني ذلك ؟

عندما أريد أن أقوم بالاتصال بحاسب آخر على الشبكة المحلية من خلال الحاسب الخاص بي و المربوط على الشبكة أيضا ، يجب أن أقوم بتحديد عنوان هذا الحاسب الذي أريد الاتصال به و الذي يميزه عن جميع الحواسب الأخرى المربوطة على الشبكة ، وكما نعلم أن هذا العنوان على الشبكة المحلية يدعى

IP Address

أيضا يوجد مفهوم آخر مرتبط بمفهوم IP Address وهو ال Port ، ماذا يعني هذا ؟

كما قلنا فان عنوان الحاسب مفيد في عملية إنشاء الاتصال ، لكن هل إنشاء الاتصال هو فقط غايتنا ، نريد أيضا تبادل البيانات مع الحاسب الآخر لذلك نحن بحاجة إلى عنوان التطبيق البرمجي الذي نريد تبادل البيانات من خلاله ، حيث أنه على نظام التشغيل Windows يتم تحديد ما يسمى بالمنفذ لكل تطبيق يعمل على نظام التشغيل وهو ما يدعى بالانكليزية Port

هذا المنفذ هو رقم يميز التطبيق البرمجي عن كافة التطبيقات البرمجية الأخرى العاملة على نظام التشغيل .

وهكذا ترى أن العنوان الكامل يتألف من جزأين (IP Address , Port)

بالإضافة إلى IP Address و Port هناك مفهوم آخر لا يقل أهمية عن مفهوم العنوان وهو البروتوكول ،إن من أشهر البروتوكولات المستخدمة في تهيئة الاتصالات و تعريف كيفية نقل البيانات بين الحواسب

UDP – TCP

تفيد هذه البروتوكولات في تحديد الصيغة التي سيتم من خلالها تبادل البيانات بين الحواسب ، حيث أنه أنواع البيانات التي يمكن أن ننقلها مختلفة ، و هي بالتالي تحتاج إلى كيفية نقل عبر الشبكة مختلفة حسب طبيعة هذه البيانات ، إن الذي يحدد هذا الأمر هو البروتوكول المستخدم

إن عملية نقل البيانات من جهاز حاسب إلى آخر على شبكة محلية تمثل نقل جميع أنواع البيانات والتي تشمل

نقل ملفات (Files) - نقل نصوص (Text) - نقل صور (images) - نقل صوت (Voice)

ملاحظة

يوجد أمر شديد الأهمية يتوقف عليه بناء البرنامج يجب أخذه بعين الاعتبار

عندما نفكر في إنشاء برنامج يقوم بنقل نوع معين من البيانات على الشبكة يجب الأخذ بعين الاعتبار نوع البروتوكول الذي يجب أن نستخدمه حيث أن بروتوكولات الشبكة تختلف من ناحية (السرعة و الأداء)

ولذلك سوف أضرب هذا المثال البسيط لتوضيح الأمر:

فمثلا عندما أريد نقل الصوت (كما في برمجيات الدردشة الصوتية) وهي عملية حرجة تمثل تحديا كبيرا للمبرمج ، فان نوع البروتوكول المستخدم يجب أن يحقق عدة أمور مثل :

السرعة الكبيرة للنقل - الأداء المتميز للنقل

حيث أن الصوت الحي المباشر من أشد أنواع البيانات التي يجب أن يتم إيصالها بالدقة المناسبة

فعندما أريد إنشاء برنامج دردشة صوتية فان البروتوكولان TCP , UDP لا يحققان الغرض على السواء و ذلك لأن :

- TCP : يحقق وثوقية عالية في إيصال البيانات بالدقة اللازمة لكنه لا يحقق شرط السرعة في النقل
- UDP : يحقق السرعة لكنه لا يحقق الوثوقية المطلوبة

و لذلك تم حل هذه المشكلة من خلال إيجاد بروتوكول يؤمن ميزات كلا البروتوكولين السابقين و يدعى RTP

خلاصة الحديث

لكي أقوم بإنشاء اتصال مع حاسب آخر على الشبكة و تبادل البيانات معه يجب أن أملك ما يلي :

1. عنوان الحاسب IP Address للاتصال به
2. عنوان التطبيق البرمجي على الحاسب Port
3. تحديد نوع البروتوكول المستخدم لنقل البيانات

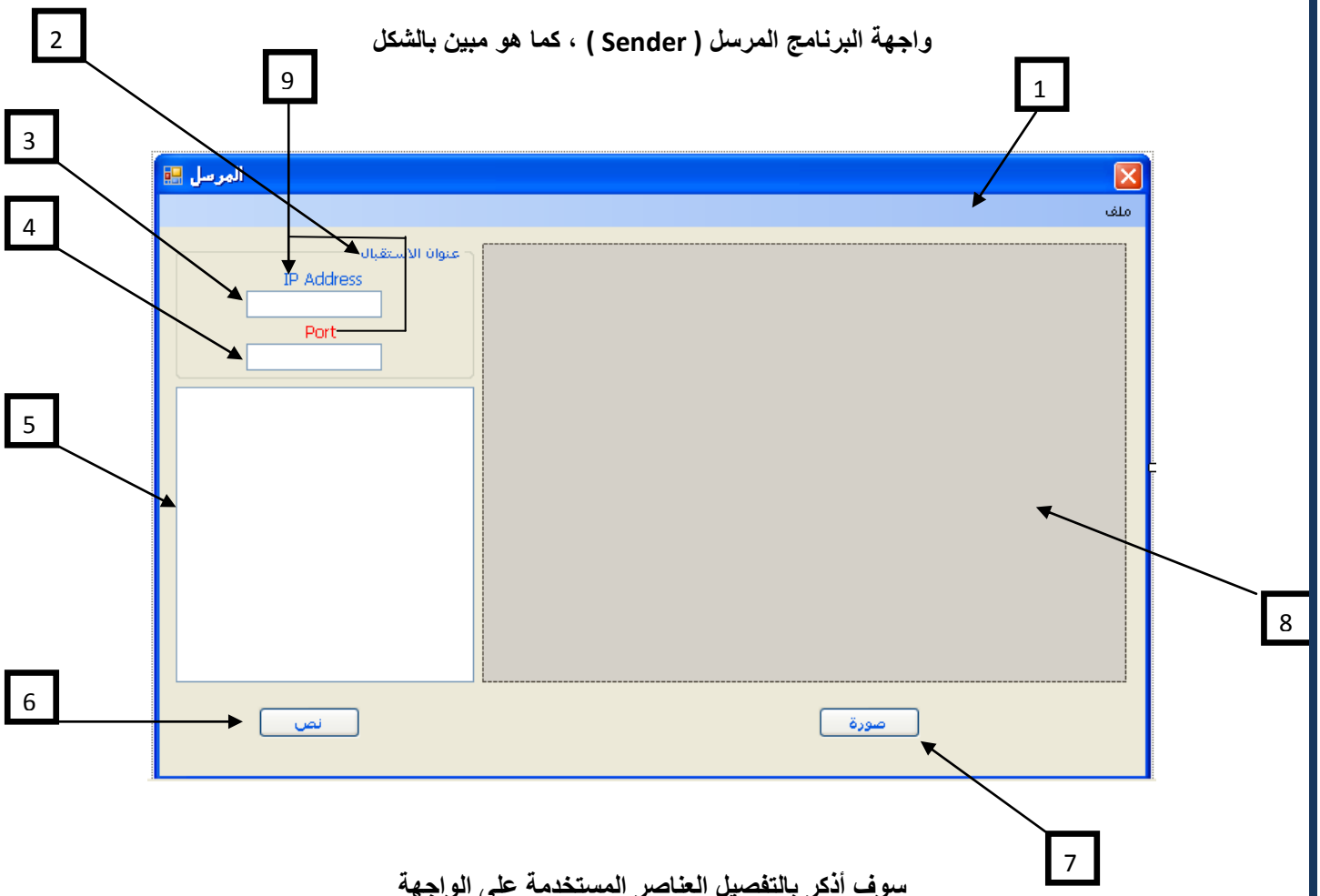
فصل الشرح العملي

في هذا الجزء الأول من الدرس سوف أقوم بشرح إرسال نص فقط ، وفي الجزء الثاني سوف أقوم بتكملة الدرس من خلال شرح إرسال الصور

1. سوف أقوم ببناء هذا التطبيق من خلال Visual Studio C# 2008، يمكنك إتباع نفس الخطوات بالضبط إذا كنت تملك Visual Studio C# 2005
2. هذا التطبيق مؤلف من برنامجين :
   - الأول هو برنامج المرسل (يقوم بإرسال بيانات من نوع " نص " , " صور ")
   - الثاني هو البرنامج المستقبل و الذي سوف يكون على حاسب آخر (يقوم بعرض البيانات المرسلة)

برنامج الجهاز المرسل

واجهة البرنامج المرسل (Sender) ، كما هو مبين بالشكل



سوف أذكر بالتفصيل العناصر المستخدمة على الواجهة

قم بإنشاء مشروع جديد من نمط Windows Application وسمه Sender (المرسل)

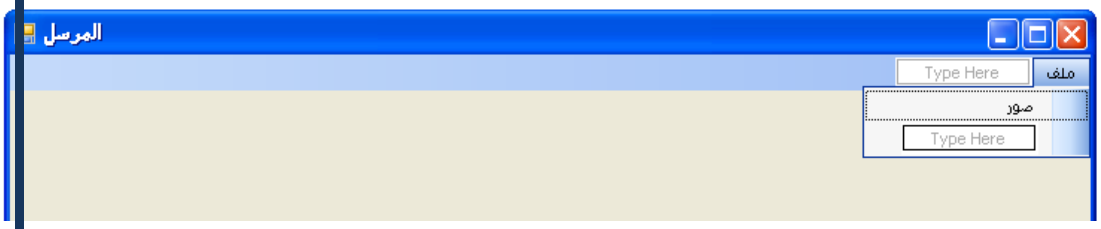
يجب عليك أن تقوم بتنزيل العناصر على الواجهة ، بالترتيب المذكور بالشكل السابق

قبل أن نتكلم عن العناصر الموجودة على الواجهة ، يجب تحديد خصائص الواجهة (Form) ، وهي كما يلي :

الخاصية	القيمة
Text	المرسل
Size(width)	735
Size(height)	463
MaximaizeBox	FALSE
MinizeBox	FALSE

عناصر الواجهة

1. **شريط القوائم (MinuStrip) ، كما في الشكل التالي:**



الخاصية	القيمة
Name	mSt
RightToLeft	yes

2. **صندوق المجموعة (GroupBox)**
3. **مربع نص (TextBox) ، لتحديد العنوان IP Address للجهاز الذي نريد الاتصال به .**

الخاصية	القيمة
Name	ipBox
TextAlign	center

4. **مربع نص (TextBox) ، لتحديد منفذ البرنامج الذي نري تبادل البيانات معه .**

الخاصية	القيمة
Name	portBox
TextAlign	center

5. **مربع نص (TextBox) ، كتابة النص المراد نقله .**

الخاصية	القيمة

txtBox	Name
center	TextAlign

6. زر (Button) ، لإرسال النص إلى الجهاز الآخر .

الخاصية	القيمة
Name	txtButton

7. زر (Button) ، لإرسال صورة إلى الجهاز الآخر .

الخاصية	القيمة
Name	imgButton

8. لوحة (Panel) ، لاستعراض الصورة المراد نقلها .
9. عناوين (Label) .

الكود البرمجي

سوف أقوم من خلال ما يلي بشرح كيفية إنشاء الاتصال بين الحاسبي و ذلك من خلال طريقتين مختلفتين :

أولا - لنقل النص (طريقة اتصال أولى)

ثانيا - لنقل الصورة (طريقة اتصال ثانية)

في البداية يجب تضمين مجالات التسمية التالية

```
Using System.Net;

Using System.Net.Sockets;
```

هذان المجالان من خلالهما تقدم لغة C# العديد من الصفوف الجاهزة و التي تمّكن المطور من تطوير برامج شبكات ،

في الصفحة التالية :

أولا - لاحظ مجالات التسمية الموجودة و قم بإضافتها جميعها

ثانيا - لاحظ المتحولات العامة التي قمت بتعريفها

(سوف نأتي على شرح كل متحول عرفناه في وقته ، ولماذا نحتاجه)

```
using System;
using System.IO;
using System.Net;
using System.Data;
using System.Text;
using System.Drawing;
using System.Net.Sockets;
using System.Windows.Forms;

namespace Sender
{
    public partial class Form1 : Form
    {
        //المتحولات العامة
        Bitmap bitmap;
        BinaryWriter bw;
        NetworkStream ns;
        TcpClient peer;
        MemoryStream stream;
        byte[] imgBuf;
        byte[] txtBuf = new byte[1024];

        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

و الآن من خلال حدث Click الخاص بالزر txtButton ، أكتب الكود الآتي :

```
private void txtButton_Click(object sender, EventArgs e)
        {
            try
            {
                UnicodeEncoding coding = new UnicodeEncoding();
                Socket peer1 = new Socket(AddressFamily.InterNetwork,
SocketType.Dgram, ProtocolType.Udp);
                IPEndPoint receiverIp = new IPEndPoint(IPAddress.Parse(ipBox.Text),
int.Parse(portBox.Text));
                string sr = txtBox.Text;
                txtBuf = coding.GetBytes(sr);
                peer1.SendTo(txtBuf, receiverIp);
                peer1.Close();
            }
            catch (Exception ex) { MessageBox.Show(ex.Message); }
        }
```

من خلال هذا الكود يتم إنشاء اتصال مع الحاسب الآخر و إرسال النص المكتوب في مربع النص إلى التطبيق

أولا قمنا بتعريف كائن من الصف UnicodeEncoding و ذلك لكي نتمكن من دعم ترميز اللغة العربية في الجهة المرسلة و المستقبلة حيث يتم عرضه بشكل صحيح ، فهذا الصف لا يدعم فقط ترميز ASCII بل العديد و العديد من تراميز اللغات .

ثانيا تعريف كائن من نمط Socket و ما أدراك ما هي ال Socket :

إنها الواجهة البرمجية التي نتمكن من خلالها من تحقيق الاتصال بين الأجهزة المربوطة على الشبكة ، و هي من أهم أساسيات برمجة الشبكات .

لاحظ الوسطاء التي ممرتها إلى التابع البناء للصف Socket

1. **الوسيط الأول : يمثل عائلة IP Address التي نستخدمها وهي هنا IP Version4 (`AddressFamily.InterNetwork`)**
2. **الوسيط الثاني : يحدد أسلوب نقل البيانات (حددناه هنا Data Gram) ،وذلك لأنني اخترت البروتوكول Udp للنقل و هو يدعم الأسلوب Dgram (أسلوب الرزم)**
3. **الوسيط الثالث : تحديد نوع البروتوكول المستخدم وهو كما ذكرت UDP.**

ثالثا الصف IPEndPoint و الذي يحدد العنوان الذي نريد الاتصال به و تبادل البيانات معه ،ويأخذ تابع البناء لهذا الصف وسيطين :

الأول - هو عنوان IP Address - الثاني - هو منفذ التطبيق Port

إن عملية الاتصال ونقل البيانات تبدأ عند السطر

```
peer1.SendTo(txtBuf, receiverIp);
```

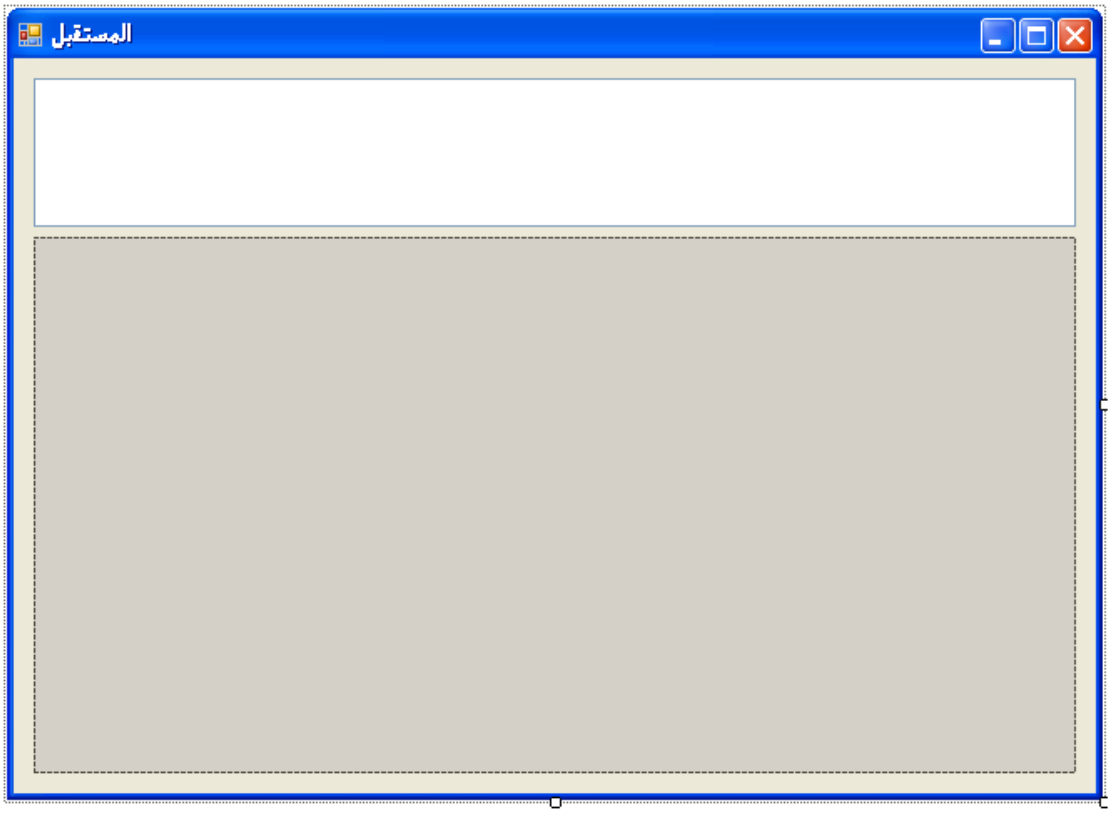
لاحظ استخدام المنهج SendTo من خلال الكائن التابع للصف Socket ، يستخدم للاتصال ونقل البيانات ، و يأخذ وسيطين :

الأول - الوسيط الأول (من نمط Buffer) ، و هو يمثل البفر المستخدم لتخزين البيانات التي نريد نقلها و قد مررت إليه مصفوفة من نمط Byte التي قمت بتخزين النص فيها على شكل Binary

ثانيا - الوسيط الثاني من نمط (IPEndPoint) و الذي يمثل العنوان الذي نريد الاتصال به

برنامج الجهاز المستقبل

وهو برنامج بسيط كما يوضح الشكل التالي :



وهو عبارة عن عنصرين TextBox لاستقبال النص

و Panel لاستقبال الصور المرسلة

أما الكود البرمجي

```
using System;
using System.Net;
using System.Data;
using System.Text;
using System.Drawing;
using System.Threading;
using System.Net.Sockets;
using System.Windows.Forms;

namespace Receiver
{
    public partial class Form1 : Form
    {
        UnicodeEncoding o = new UnicodeEncoding();
        Socket RecPeer;
        Thread thread;
```

```
        byte[] txtBuf = new byte[1024];

                        التابع البناء للنموذج
        public Form1()
        {
            InitializeComponent();

            thread = new Thread(new ThreadStart(TxtReceive));
            thread.Start();
        }

                    منهج استقبال الاتصال و النص المرسل

        void TxtReceive()
        {
            try
            {
                IPEndPoint localIp = new IPEndPoint(IPAddress.Any,
5000);

                while (true)
                {

                    RecPeer = new Socket(AddressFamily.InterNetwork,
SocketType.Dgram, ProtocolType.Udp);
                    RecPeer.Bind(localIp);
                    RecPeer.Receive(txtBuf);
                    RecPeer.Close();
                    textBox1.Text =
Convert.ToString(o.GetString(txtBuf));
                    Array.Clear(txtBuf, 0, txtBuf.Length);
                }
            }
            catch { }
        }

        private void Form1_FormClosing(object sender,
FormClosingEventArgs e)
        {
            thread.Abort();
        }
    }
}
```

لاحظ مجالات التسمية في الكود تلاحظ استخدام المجال System.Thread ،وهو أمر ضروري لأننا في هذا البرنامج نحتاج إلى استخدام مسالك تنفيذ على التوازي مع البرنامج في تابع البناء للنموذج .

في منطقة المتحولات العامة تلاحظ بأنني عرفت كائن UnicodeEncoding لدعم اللغة العربية .

كما عرفنا أيضا كائن من نمط Socket ، بالإضافة إلى تعرف كائن من نمط Thread .

و عرفت أيضا مصفوفة Bytes لتمثيل البفر(Buffer) بالاستقبال .

نلاحظ أيضا المنهج TxtReceive ، و الذي أنشأته للقيام بعملية استقبال الاتصال و البيانات المرسلة .

لاحظ أنني عرفت كائن IPEndPoint والذي يمثل هنا العنوان المحلي للجهاز المستقبل ، تتذكر بأننا قمنا بنفس العملية في برنامج المرسل ؟

لاحظ الحلقة اللانهائية التي وضعتها !!!!!!!

الآن تفهم لماذا أحتاج إلى (thread) مسلك منفصل عن مسلك البرنامج الرئيسي .

تتذكر في كود برنامج الإرسال بأنني قمت بإغلاق الاتصال ;()peer1.close

حيث أنني في برنامج الإرسال أقوم بعملية الاتصال و الإرسال و إغلاق الاتصال كل عملية Click على زر الإرسال ، (من دون إغلاق الاتصال لن أعود قادرا على إرسال بيانات مرة أخرى في هذه الطريقة) .

لذلك في برنامج الاستقبال أحتاج دوما إلى استقبال الاتصال ومن ثم استقبال البيانات ومن ثم إغلاق الاتصال وبعد ذلك أعود لفحص عملية اتصال جديدة ، هذا الترتيب واضح من خلال استدعاء المناهج التابعة للصف Socket.

المنهج Bind من الصف Socket أقوم من خلاله باستقبال الاتصال على العنوان المحلي .

المنهج Receive من الصف Socket أقوم من خلاله باستقبال البيانات المرسلة .

لاحظ السطر التالي

```
Array.Clear(txtBuf, 0, txtBuf.Length);
```

من خلال هذه السطر أقوم باستخدام صف جاهز توفره لي C# من خلال System واستفيد من المنهج Clear لكي أقوم بتنظيف البفر (Buffer) ، لاستقبال بيانات من جديد .

لاحظ أيضا حدث closing_Form و الذي من خلاله أقوم بعملية إجهاض لل thread .

لاحظ أنه استدعاء المنهج TxtRceeive يتم من خلال thread منفرد في تابع البناء .

في الفصل القادم إن استطعت و يسر الله لي سوف أقوم بتكملة البرنامج والذي سوف أقوم فيه بشرح عملية اتصال مختلفة و كيفية إرسال بيانات مثل صور عبر الشبكة .