بسم الله الرحمن الرحيم

الجمهورية اليمنية
جامعة إب
كلية العلوم
قسم : رياضيات حاسوب

نظام إدارة معمل الحاسوب

عن طريق الشبكات المحلية LAN

إعداد الطلاب

إدريس محمد النظاري.
زايد أحمد عبده إسماعيل.
عبد العزيز طاهر اليفرسي.
محمد حزام محمد عبدالله.
محمد قايد غانم الشامي.
يونس عبدالوهاب الصباحي.

إشراف

أ.د/رشاد الجوفي

العام الجامعي ٢٠٠٩ - ٢٠١٠م

﴿اقرأ باسم ربك الذي خلق ۞ خلق الإنسن من علق ۞ اقرأ
وربك الأكرم ۞ الذي علم بالقلم ۞ علم الإنسان ما لم يعلم﴾

صدق الله العظيم

zayedalbadanee@yahoo.com

إهداء

إلى الشمعة التي تحترق لتضيء لي الطريق . . . .

إلى من ذبلت لأزهر . . . . . .

إلى من تعبت لأرتاح . . . . . .

إلى من ظللتني دعواتها . . . . . .

إلى والدتي

إلى من كد ليمهد لي الطريق . . . . . .

إلى الصدر الواسع والقلب المشفق . . . . . .

إلى والدي

إلى الأقمار التي جعلت حياتي بهجة ومتعة . . . . .

إلى من قاسموني الحياة حلوها ومرها . . . . . .

إلى إخواني

zayedalbadanee@yahoo.com

الفهرس

zayedalbadanee@yahoo.com

الفصل الأول

المقدمة 4
المشاكل التي يتناولها المشروع 5
أهداف المشروع 5
الأدوات المستخدمة في المشروع 6
نمذجة النظام 7
الخطة الزمنية 9

الفصل الثاني

تعريف الشبكات 10
أنواع الشبكات من حيث نطاق التوصيل 11
أنواع الشبكات من حيث طرق التوصيل 15
الأسباب التي أدت إلى إنشاء شبكات الكمبيوتر 17
مكونات الشبكات الأساسية 18
الشبكات المحلية 19

الفصل الثالث

البروتوكولات 22
مكونات أصناف بروتوكولات الـ TCP وبروتوكول الـ UDP 23
الأصناف الخاصة ببروتوكول الـ TCP 23
الأصناف الخاصة ببروتوكول الـ UDP 26
طبقات الشبكات والـ Socket المتزامنة 27
استخدام برمجة الـ Socket المتزامنة لإنشاء اتصال عبر بروتوكول الـ TCP 28
استخدام برمجة الـ Socket المتزامنة لإنشاء اتصال عبر بروتوكول الـ UDP 30
مكونا ت أصناف الـ Socket المتزامنة 31
برمجة الـ Socket الغير المتزامنة 33
مكونا ت وأصناف الـ Socket الغير متزامنة 33
تطبيقات عدم التزامن في الدوت نت 34
دوال مكتبة الـ I/O 35
أصناف الـ Stream 36
الدوال التي تستخدم لإدارة الـ Stream 38
مفهوم الـ Threading 40

الـ Threading في الدوت نت 41
الطرق التابعة للصنف Thread 43
الخواص التابعة للصنف Thread 44

الفصل الرابع

عملية إرسال النص 49
عملية استقبال النص 52
عملية إرسال النص لمجموعه 53
عملية استقبال النص المرسل لمجموعه 54
عملية إرسال الملفات 56
عملية استقبال الملفات 58
عملية مراقبة الأجهزة 60

الفصل الخامس

عملية عرض الدرس 64
عملية عرض الدرس باستخدام بروتوكول الـ TCP 64
عملية عرض الدرس باستخدام بروتوكول الـ UDP 65
عملية عرض الدرس باستخدام قنوات الـ TCP 67
عملية استخدام جهاز الطالب 69
عملية تجميد وفك تجميد الماوس ولوحة المفاتيح 74
عملية أبقاف تشغيل جهاز الكمبيوتر 75

الفصل السادس

مميزات النظام 76
الاستنتاجات 76
التوصيات والمقترحات 77
شكر وعرفان 78
المراجع 79

الملاحق

بناء الشبكات المحلية 80
ضبط إعدادات الشبكة 85

zayedalbadanee@yahoo.com

الفصل الأول

تعريف بالمشروع

المقدمة

المشاكل التي يتناولها البحث

أهداف البحث

الأدوات المستخدمة في البحث

نمذجة النظام

الخطة الزمنية للتنفيذ

zayedalbadanee@yahoo.com

مقدمة:-

مع التطوّر العلمي الذي شُهد في العصر الحاضر في مجال التكنولوجيا والمعلوماتية الذي يُلاحظ تأثيره الواضح على حياتنا العلمية والعملية فقد دخل الحاسوب في جميع ميادين الحياة حتى أنه أعتبر في الكثير من المجتمعات الأداة الأساسية للحياة العصرية سواء في خدمة الأعمال التجارية والعلمية المختلفة أو في مجالات الثقافة والترفيه والتسلية...ولا يُعتبر الحاسوب الأداة الأولى والأهم في هذا العصر لو لم تكن هناك الوسائل القادرة على ربط الحواسب ببعضها، حيث بإمكاننا اليوم تبادل كميات كبيرة من المعلومات وجعلها تنتقل عبر مسافات بعيدة بمجرد الضغط على لوحة المفاتيح ، فقد ألغت شبكات الحاسوب مفهوم الحواجز الجغرافية والحدود الإقليمية وأصبحت ضرورة ملحّة لكل شركة أو مؤسسة ناجحة وأيضا لكل فرد من أفراد المجتمع .

وإننا من خلال مشروعنا هذا نسعى إلى توفير الكثير من الجهد على الطالب والمعلم على حداً سوى وجعل عمليه التعليم داخل المعامل أكثر سهولة ومرونة من خلال تقديم طريقة تعليم حديثة والابتعاد عن الطرق التقليدية في التعليم حيث تمكن هذه الطريقة من سهولة الاتصال بين المعلم والطالب داخل المعمل من خلال توفير الخدمات التالية:

١. تمكين المعلم من استخدام باقي الأجهزة من خلال جهازه.
٢. عرض ما يقوم بشرحه المعلم في بقية الأجهزة
٣. التحكم الكامل بجهاز الطالب (إيقاف تشغيل – إعادة تشغيل – تجميد –إغلاق وفتح...).
٤. إرسال الملاحظات والجمل التوضيحية إلى الطلاب من جهاز المعلم .
٥. مراقبة عمل الطلاب من جهاز المعلم .
٦. إمكانية تبادل إرسال الملفات (كتب–محاضرات ....) .

المشاكل التي يتناولها المشروع

من خلال الدراسة في معمل الحاسوب وجدت المشاكل التالية:

١. الجهد: صعوبة متابعة المعلم للطلبة اثنا التدريس (يحتاج إلى مجهود كبير).

٢. الوقت:- ضياع الوقت داخل معمل الحاسوب (يحتاج المعلم إلي متابعة كل طالب).

٣. عدم أمكانية تبادل البيانات(ملاحظات- ملفات- رسائل...) بين المعلم والطالب عبر الأجهزة .

٤. عدم قدرة المعلم على التحكم في أجهزة الطلاب.

ومن هنا نشأت فكرة إنشاء البرنامج الذي من شأنه أن يقوم بأداره معمل الحاسوب .

أهداف المشروع:-

١. تطوير أسلوب التعليم داخل معامل الحاسوب.

٢. أمكانية التواصل المباشر بين المعلم والطلاب عبر ألشبكه.

٣. إرسال واستقبال البيانات(ملاحظات-ملفات-رسائل...) بين المعلم والطالب.

٤. شرح المحاضرات عن طريق الشبكة.

٥. التعديل علي أخطاء الطلاب من خلال جهاز المعلم.

٦. مراقبة المعلم لجميع أجهزة الطلاب من جهازه.

٧. التحكم بجميع أجهزة الطلاب عن طريق الجهاز الخاص بالمعلم وبطريقة سهلة وواضحة.

zayedalbadanee@yahoo.com

الأدوات المستخدمة في المشروع:

الأدوات البرمجية:

١. Microsoft Access: حيث سنقوم من خلاله ببناء قاعدة البيانات الخاصّة بالنظام من ناحية الجداول والعلاقات فيما بينها .

٢. Visual C#.Net : إحدى لغات الدوت نت وتمتاز بسهولة التعامل معها إلى جانب دعمها لجميع أنواع التطبيقات حيث يمكن استخدامها في بناء أنظمة الشبكات .

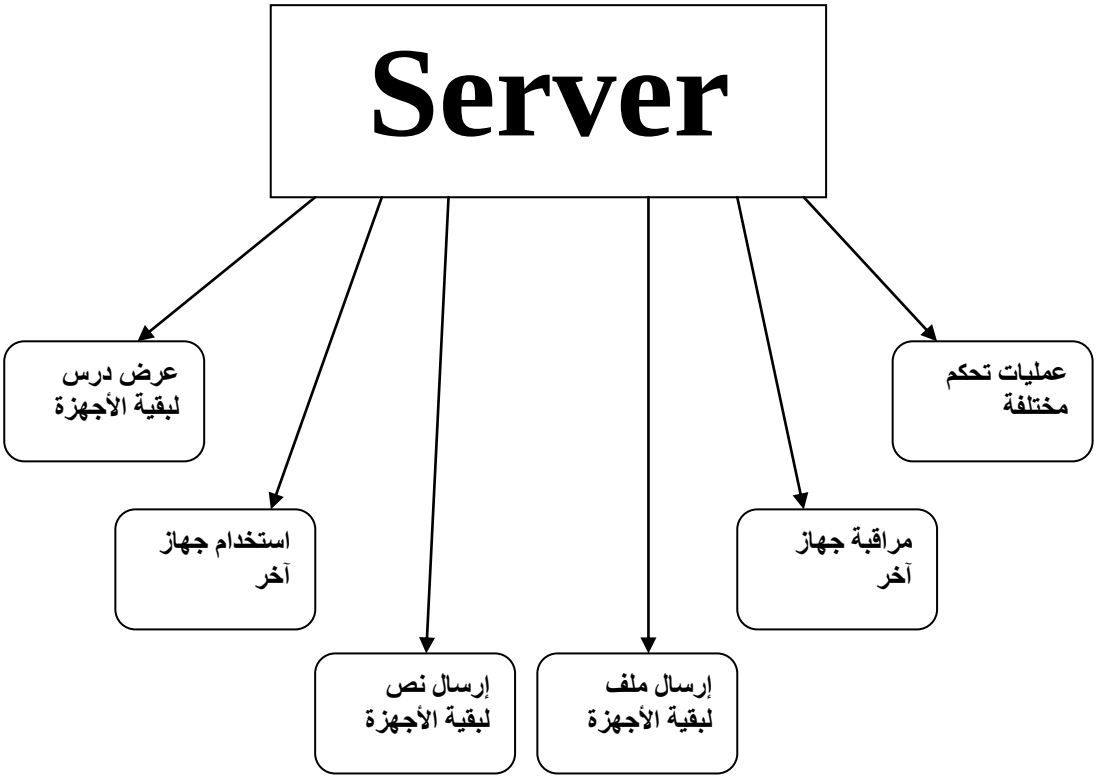
zayedalbadanee@yahoo.com

نمذجة النظام :-

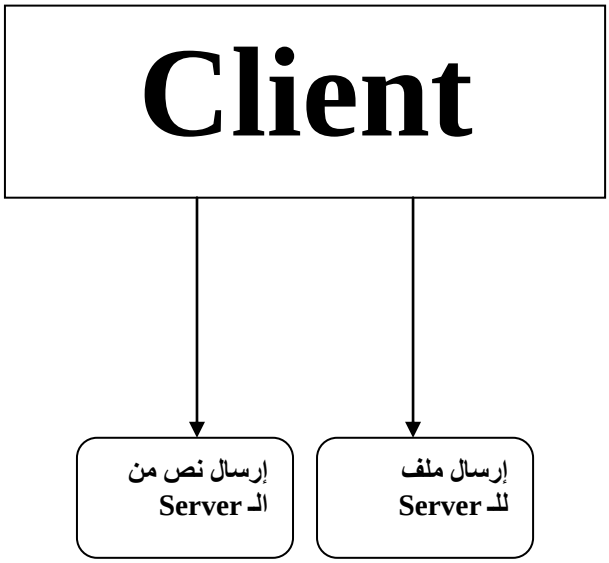
تعتبر العمليات الموضحة في الأشكال التالية تلخيص للخدمات سنحاول توفيرها في النظام

شكل(1-1) النموذج العام للنظام

zayedalbadanee@yahoo.com



الشكل(2-1)العمليات الأساسية للـــ Server



الشكل(3-1)العمليات الأساسية للـــ Client

الخطة الزمنية للمشروع:-

- المرحلة الأولى : تحليل المشكلة

شهر واحد ، شملت دراسة جوانب وأبعاد المشكلة .

- المرحلة الثانية: تصميم الحل

شهر واحد ، تضمن إيجاد حل ملائم لمشكلة إدارة معامل الحاسوب عبر الشبكات.

- المرحلة الثالثة : تنفيذ الحل

3 أشهر ، تضمن ذلك بناء البرنامج وبناء قاعدة البيانات مع الاستفادة من خبرات الدكتور المشرف وخبرات بعض الأساتذة والزملاء . مع محاولة إحاطة البرنامج بأكبر قدر من الحماية والمرونة .

zayedalbadanee@yahoo.com

الفصل الثاني

نظرة عامة عن الشبكات

تعريف الشبكات.

أنواع الشبكات.

أنواع الشبكات من حيث طرق التوصيل.

الأسباب التي أدت إلى إنشاء شبكات الكمبيوتر.

مكونات الشبكات الأساسية.

الشبكات المحلية.

zayedalbadanee@yahoo.com

تعريف الشبكات

١. الشبكة:

هي وسيلة ربط بين أجزاء تتعلق ببعضها، تجمع بينها وظيفة مشتركة، وهي نظام قنوات أو خطوط تقوم بربط عدة نقاط ببعضها، وتدعى هذه النقاط بعقد أوNodes ، كعقد محطات شبكة القطارات، أو المدن المتصلة ببعضها بواسطة الطرق . إننا نستخدم الشبكات كل يوم، مثال على ذلك، شبكة الهاتف.

ينطبق هذا المبدأ على الشبكات الحاسوبية فهي تربط الحواسب ببعضها و بطرفيات أخرى، تماماً كما هو الحال بين المدن والمحطات.

٢. شبكة الكمبيوتر:

هي مجموعة من أجهزة الكمبيوتر والأجهزة المحيطية (Peripherals) التي تتصل ببعضها، وتتيح لمستخدميها أن يتشاركوا الموارد (Resources) والأجهزة المتصلة بالشبكة مثل الطابعة (Printer) وكارت الفاكس (Modem) ومحرك القرص المدمج (CD – ROM Drive)وغيرها. وهذا المفهوم هو الأساس الذي يقوم عليه التشبيك ونظرياته.ويجب أن لا يقلّ الحد الأدنى لمكونات شبكة الكمبيوتر عن:

- جهازي كمبيوتر على الأقل .
- بطاقة شبكية (Network Interface Card NIC) والتي تشكل جسر الاتصال بين الكمبيوتر وأسلاك النقل التي تربط مكونات الشبكة .

- وسط ناقل (Transmission Media) للاتصال بين عناصر الشبكة مثل الكابلات (Cables)والأسلاك (Wires) أو الأمواج القصيرة (Waves Radio)والألياف الضوئية (Fiber Optic).
- بروتوكول اتصال يحدد خوارزمية تخاطب مكونات الشبكة والمواصفات التقنية الواجب توفرها مثل عرض الحزمة المستخدم (Bandwidth) وطريقة ترتيب المعلومات عند إرسالها Formats (Packets) وغيرها من المواصفات التقنية .
- نظام تشغيل شبكي (Network Operating System- NOS) يقدم خدمة تنظيم صلاحيات وحقوق المستخدمين (Rights And Permissions) في الوصول إلى الموارد والأجهزة المشتركة على الشبكة ومن أمثلته (Windows98، Windows 2000، Novell Netware)

أنواع الشبكات من حيث نطاق التوصيل :

١. الشبكات المحلية LAN

تدعى الحواسب المرتبطة ببعضها في منطقة جغرافية واحدة صغيرة ، أو شبكة محلية اختصاراً أو LAN كما هو شائع، وهو اختصار يرمز إلى Local Area Networkتعود ملكية هذه الشبكات عادةً إلى شركة واحدة، تكون مكاتبها في نفس الموقع أو البناء أو ضمن حرم واحد.

يوضح المخطط أدناه خارطة شبكة محلية خاصة بشركة، وتحوي على شبكات محلية صغيرة في كل قسم من أقسام الشركة، ويوجد في كل قسم مخدم خاص. يتم ربط كل من الشبكات الصغيرة بمجمّع أو محوّل مركزي Hub/Switch يقوم بربطها مع المخدم الرئيسي. في هذا المخطط، يكون لكل من أقسام الشركة مخدم خاص واحتياجات

معلوماتية خاصة. يمنع هذا الترتيب الازدحام على المخدم أو الشبكة، الأمر الذي يحسن من أداء الشبكة ككل. وتعرف هذه التقنية بتقنية تقسيم الشبكة.Subnetting

حيث سوف يتم شرح هذا النوع من الشبكات بشكل مفصل في نهاية هذا الفصل .

٢. الشبكات الإقليميةWANs

الشبكات الإقليمية WANs هي شبكات تربط بين عدة شبكات موجودة في مناطق متباعدة جغرافياً، ويرمز اختصار WAN إلى .Wide Area Networks تمتد الشبكات الإقليمية ضمن مدينة، أو دولة، أو قارّة، أو حتى عبر الكرة الأرضية. تتم عملية وصل الشبكات الصغيرة ببعضها من خلال بنية اتصالات. بإمكان شركات الاتصالات القيام بتزويد خدمة الربط الإقليمي المطلوب لقاء أجر محدد.

٣. شبكة مؤسسة الهاتفPSTN

إن أقل أنظمة الربط الإقليمي كلفة هي تلك التي تتم عن طريق شبكة مؤسسة الهاتف PSTN، وهي نفس الشبكة التي تقوم بتخديم المكالمات الهاتفية الصوتية.

أما فيما يتعلق بالأداء، فهو جيد للاستعمال المنزلي، ولكنه غير كاف بالنسبة للشركات. إن عرض قناة الاتصال على شبكة مؤسسة الهاتف هو 56 كيلوبت/الثانية، ويتم التوصل إلى هذه السرعة باستخدام تقنية ضغط المعطيات. ولكن بسبب العوامل البيئية، كنوعية الخط الهاتفي ومزايا مقاسم الهاتف، فإن هذه السرعة تظل غير فعلية. ويتم الوصل بواسطة موديم يربط الحاسب الشخصي بخط الهاتف، ويقوم بمهمة التحويل من تماثلي إلى رقمي.

٤. نظامISDN

يوفر نظام شبكة ISDN أو Integrated Services Digital Network

"مجموعة متكاملة من الخدمات الرقمية للشبكات". شبكة ISDN هي أسرع وأكثر وثوقية من شبكة مؤسسة الهاتفPSTN ، وتستخدم الألياف الضوئية لنقل الإشارة.

أهم مزايا ISDN :

- سرعات نقل عالية: سرعة كل قناة ISDN 64 كيلوبت/الثانية.
- الشبكة رقمية بالكامل، ولا داع لاستخدام المحولات الرقمية/التماثلية (الموديمات) .
- زمن إجراء الطلب الهاتفي قصير جداً بالمقارنة مع الموديم الذي يستغرق حوالي 30 ثانية.
- يتم نقل الصوت رقمياً بنفس طريقة نقل المعلومات الرقمية.

هناك نوعين من تقنيات ISDN :

- ISDN2 يوفر هذا النوع قناتي اتصال سعة كل منهما 64 كيلوبت/الثانية (قناةB)، وقناة واحدة لإجراءات التحكم بالشبكة سعتها 16 كيلوبت/الثانية. ومن الممكن استخدام قناة خاصة تدعى قناة D ضمن ظروف خاصة لنقل المعطيات بسرعة 144 كيلوبت/الثانية .
- ISDN30 يوفر 30 قناة B ، ويعطي سعة خط إجمالية بحدود 2ميغابت/الثانية، وتعتمد هذه السرعة على عدد القنوات المستخدمة في آن واحد. إن إضافة قنوات يزيد من السعة، ولكنه ايضاً يزيد الكلفة.

٥. خدمةxDSL :

آخر تقنيات الاتصال هي خدمة xDSL أوx Digital Subscriber Line ، وتسمى

بخدمة خط المشترك الرقمي، وهي الحل الأفضل للاستخدام في الشركات، ولها تكلفة أكبر .إن خدمة DSL توفر اتصالاً دائماً مع الشبكة، كما هو الحال مع خدمةISDN ، الأمر الذي يلغي الحاجة لإجراء الطلب الهاتفي الذي يقوم به المودیم. هناك عدة أنواع لتقنياتxDSL

٦. الخدمة السلكيةCable :

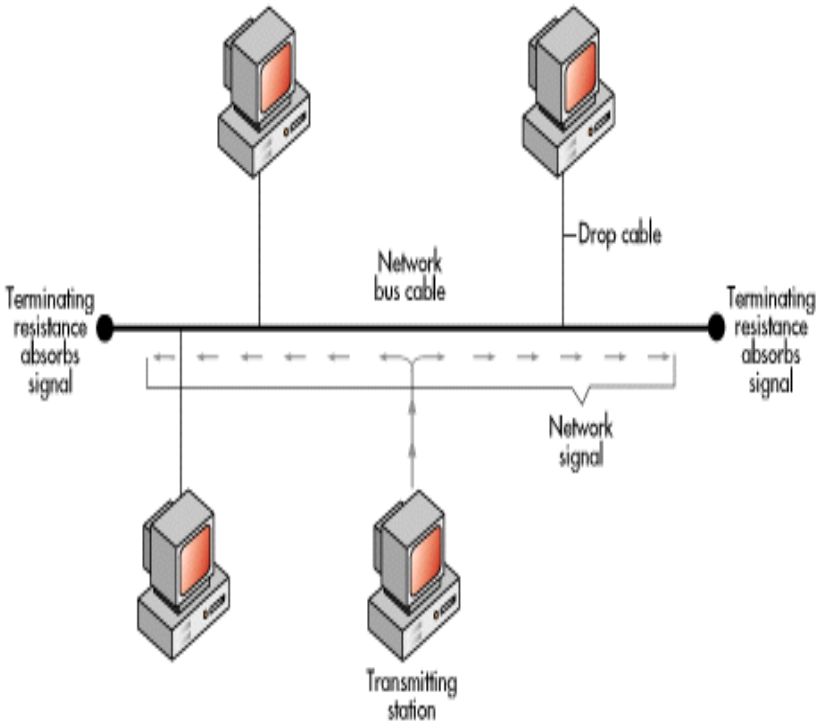
تستخدم الخدمة السلكية Cable شبكات التلفزيون السلكية الموجودة في بعض الدول، ويتم نقل المعطيات بسرعة 512 كيلوبت/الثانية، وتعد هذه الخدمة أفضل من بعض تقنيات .xDSL تستخدم هذه التقنية نفس نوع الكبل المحوري Coax Cable الذي ينقل الإشارة التلفزيونية للمنازل. يتم ربط الحاسب الشخصي بعلبة خدمة الاشتراك التلفزيوني عن طريق كرت الشبكة الموجود في الحاسب.

من المهم أن تدركوا أنه على الرغم من كون خط الاتصال هذا يبدو وكأنه خط اتصال خاص بين جهتين، إلا أنه ليس كذلك، فالمعطيات تنتقل عبر عدة موجهّات ومقاسم قبل أن تصل إلى هدفها.

zayedalbadanee@yahoo.com

أنواع الشبكات من حيث طرق التوصيل

١. الشبكة الخطية (Bus Topology)



الشكل(1-2) الشبكة الخطية

zayedalbadanee@yahoo.com

التوصيل بهذه الطريقة يكون توصيل خطى و تعتبر ابسط طرق التركيب:

- عدة أجهزة كمبيوتر .
- كابل رئيسي يمر بجميع الأجهزة و يتفرع إلى عدة فروع لتوصيلة بأجهزة الشبكة.

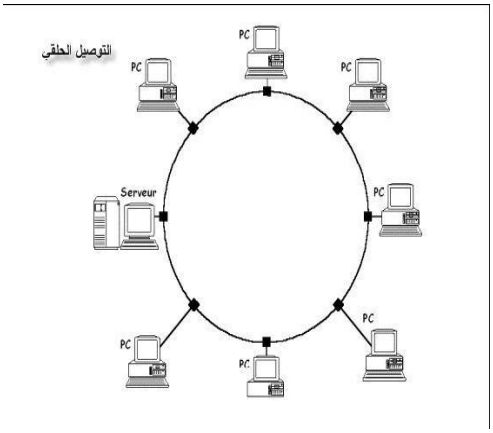
عيب هذه الطريقة:

- إذا حدث عطل بالكابل الرئيسي يؤدى إلى عطل الشبكة بالكامل .
- على كل جهاز انتظار دوره في إرسال البيانات و إلا سوف يحدث تصادم .
- كلما زاد عدد الأجهزة طال وقت الانتظار و زاد بطء الشبكة .
- في حالة عطل احد الأجهزة تدور البيانات في الكابل و يسبب عطل الشبكة .

و للتغلب على هذا العيب و إيقاف الإشارة و منع ارتدادها تستخدم وصلة خاصة تسمى

نهاية طرفية Terminator وتوضع في طرفي الكابل .

٢. الشبكة الحلقية



الشكل(2-2) الشبكة الحلقية

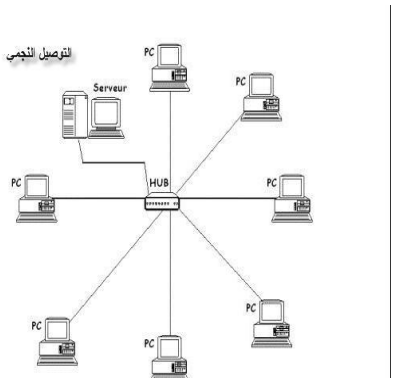
zayedalbadanee@yahoo.com

- يتم ربط الأجهزة على شكل حلقة أو دائرة بدون نهايات .
- تنتقل المعلومات كما في التوصيل الخطى .
- من مميزاتها مرور البيانات في اتجاهين .
- كل كمبيوتر يعتبر مكرر للإشارة و يقويها .

عيب هذه الطريقة:

في حالة توقف احد الأجهزة تتوقف الشبكة بالكامل.

٣. الشبكة النجمية



الشكل(2-3) الشبكة النجمية

- يتم توصيل كل جهاز من أجهزة الشبكة بجهاز يسمى (مجمع)Hub أو جهاز يسمى (مبدل)Switch.
- اى كل الأجهزة توصل إلى نقطة واحدة .
- إذا توقف اى جهاز أو انقطع السلك الذي يربطه بالمجمع لن تتأثر باقي الشبكة .

عيب هذه الطريقة:

إذا حدث عطل في Hub تتوقف الشبكة كليا عن العمل .

الأسباب التي أدت إلى إنشاء شبكات الكمبيوتر

١. **مشاركة المعطيات:**

تسمح لمجموعة من المستخدمين بتبادل المعلومات بشكل منتظم و سريع فقد تكون هذه المعطيات عبارة عن تقرير قام بإعداده موظف في دولة و استفاد منه موظف آخر في دولة أخرى .

٢. **مشاركة التطبيقات:**

توفر المشاركة في البرمجيات التي تم تنصيبها على المخدم (Server).

٣. **مشاركة الأجهزة :**

توفر إمكانية الاستفادة من الطرفيات الموجودة بالشبكة (كالطابعات – الماسح الضوئي- أجهزة الفاكس ....) لتوفير تكاليف شراء العديد من هذه الأجهزة .

٤. **الاتصالات :**

تسهل الاتصالات بين مستخدمي الشبكة من خلال البريد الالكتروني و الرسائل الفورية

٥. **التوافقية :**

لتسهيل عملية صيانة البرمجيات و التطبيقات حيث يتم تحديث البرمجيات مركزيا – أي يقوم مدير نظم المعلومات يعمل على تعديل البرمجيات الموجودة على المخدم Server فقط .

٦. **الأمن :**

يعتبر امن الشبكة غاية في الأهمية – اذ يحتاج المستخدم إلى حساب خاص للدخول إلى الشبكة و يجب لاستخدام أي مورد من الشبكة استخدام كلمة مرور و اسم الحساب .كما يمكن منع بعض المستخدمين من الدخول إلى بعض الأجهزة في أوقات محددة .

٧. **الدخول إلى الانترنت :**

بعد توفير البرمجيات و الأجهزة يمكن للمستخدمين الدخول إلى الانترنت من الشبكة الداخلية للوصول إلى كم ضخم من المعلومات و البرمجيات الإضافية .

مكونات الشبكات الأساسية

١. مكونات مادية Hardware :

٢. الخادم Server: هو الحاسب الخاص بتشغيل الشبكة ويقدم خدمات لكل الأجهزة الموجودة على الشبكة مثل

- تخزين الملفات
- إدارة المستخدمين
- التأمين
- إدارة النظام

٣. محطات العمل Work Stations

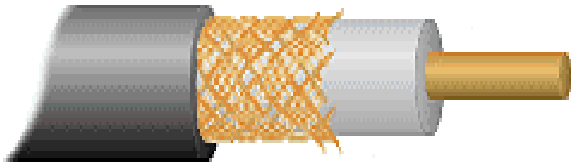
هي احد أجهزة الكمبيوتر المتصلة بالشبكة وتعمل إما بنظام Windows او Macintosh.

٤. كروت الاتصال Network Interface Card

هي وسيلة ربط جهاز الكمبيوتر بالشبكة و ه ي إما أن يكون كروت منفصلة عن الجهاز أو مثبتة على M.B (اللوحة الأم) .

٥. الكابلات

هي الأسلاك المستخدمة لربط الخادم مع محطات العمل



الشكل(4-2) الكابلات

٦. مكونات غير مادية Software

أنظمة Win3.1 و Dos لم تكن تدعم تطبيقات الشبكات إما إصدارات الويندوز التالية تعم نظم الشبكات.

الشبكات المحلية:

١. ما هي الشبكة المحلية:

الشبكة المحلية هي شبكة كمبيوتر (computer network) تنقل المعلومات بسرعة عالية ضمن مساحة جغرافية محدودة (مثلا: بناية واحدة أو عدةبنايات). وتربط هذه الشبكة مجموعة من محطات العمل (workstations) مع بعضها، وذلك بما يُتيح لهذه المحطات تشارُك موارد الشبكة من عتاد (hardware) وبرمجيات(software) ، إضافةً إلى تمكين مستخدِمي الشبكة من تبادُل الملفات والاتصال فيما بينهم عبر البريد الإلكتروني (Email) والجلسات الحوارية.(chat) .

٢. طُرُق الولوج إلى الشبكة المحلية:

كي تتمكَّن الأجهزة الموجودة في الشبكة المحلية من تبادُل المعلومات فيما بينها؛ لا بد لها من مجموعة من قواعد الاتصال المعيارية المتفَّق عليها مُسبَقاً، وتدعى هذه القواعد بروتوكولات(protocol) ، فمن أجل إرسال رسالة من جهاز إلى آخر عبر الشبكة، تُجزَّأ الرسالة في الطرف المرسِل إلى وحدات بيانات تُدعى الحُزَم (packets)، وتُرسَل هذه الحُزَم عبر خطوط الاتصال لِيُعاد تجميعُها في الطرف المستقبِل .

وهناك عدة بروتوكولات تُستَخدَم لِحلِّ مشكلة تشارُك وسط النقل transmission) (mediumفي الشبكات المحلية. وتعتمد هذه البروتوكولات إحدى الطريقتين التاليتين للوصول إلى الشبكة:

- التنافُس :(contetation) تطرأ الحاجة إلى التنافُس عند محاولة أكثر من جهاز كمبيوتر استخدام وسط النقل في الوقت نفسه، مما يؤدي إلى حدوث تصادم.(collision) أما آليات تخفيف ذلك التصادم فهي عديدة، ومنها :
  - تحسُّس وسط النقل :(carrier sensing) آلية تعتمد على تأكد أجهزة الكمبيوتر من خُلوّ وسط النقل قبل استخدامه .
  - تحرّي وسط النقل :(carrier detection) في هذه الآلية، تبقى أجهزة الكمبيوتر تراقب وسط النقل حتى أثناء استخدامها له .

ويُدعى البروتوكول الذي يستخدم كلا هاتين الآليتين بروتوكول CSMA/CD اختصاراً للمصطلح الأجنبي(carrier sense multiple access collision detect) ، وهذا البروتوكول مستخدَم في جميع أنواع شبكات إيثرنت .(Ethernet)

- تمرير الشارة :(token passing) في هذه الطريقة، ينتظر جهاز الكمبيوتر الذي يريد استخدام الشبكة مرور شارة (token) تدور في الشبكة، وتخبره عند وصولها إليه متى يُسمَح له باستخدام الشبكة. ويُدعى البروتوكول الذي يستخدم هذه الطريقة بروتوكول توكن رينغ.(token ring protocol).

وتُعدّ طريقة تمرير الشارة (token passing) أفضل وأكثر معوَّلية من طريقة التنافس(contention) ، ولكنها– بالمقابل -أكثر كلفة

٣. طُرُق الإرسال في الشبكات المحلية:

تُرسَل المعلومات في الشبكات المحلية إلى العُقَد الأخرى بإحدى ثلاث طُرُق، وفي كل طريقة منها تُرسَل حزمة واحدة من المعلومات إلى عقدة أو أكثر، ففي الإرسال

الأحادي (unicast) يتم الإرسال إلى عقدة واحدة، أما في الإرسال المتزامن المتعدِّد الوُجهات (multicasting) فيتم الإرسال إلى أكثر من عقدة، بينما في النوع الأخير المُسمى الإرسال العام أو البث (broadcasting) فتُرسَل حزمة المعلومات إلى جميع العُقَد في الشبكة .

الفصل الثالث

أهم الأصناف المستخدمة في برمجة الشبكات

مصطلحات الفصل

مكونات أصناف بروتوكول الـ TCP وبروتوكول الـ UDP

طبقات الشبكات والـ Socket غير المتزامنة

برمجة الـ Socket الغير متزامنة

دوال مكتبة الـ I /O

مفهوم Threading

أهم المصطلحات المستخدمة في المشروع

١. الـــ Socket :هي أداة توفرها الدوت نت يتم من خلالها نقل البيانات من جهاز إلى آخر .

٢. الـــ Thread :هي تقنية تقوم بعمل توازي في تنفيذ المعالجات داخل CPU أي انه يمكن القيام بأكثر من معالجة في نفس الوقت .

٣. Server :هو الحاسب الخاص بالتحكم بالشبكة ويقدم خدمات لكل الأجهزة المتصلة بالشبكة.

٤. Client :وهو أحد الاجهزة المتصلة بالشبكة والمستفيدة من خدمات الـــ Server.

٥. Stream : هي مكتبة في الدوت نت تسهل عملية نقل البيانات من مكان الى آخر سواء عبر الشبكة أو على نفس الجهاز من خلال حجز مناطق في الذاكرة الظاهرية RAM.

٦. **البروتوكولات (protocol)**

- **البروتوكول :** هو عبارة عن مجموعة من القوانين و الإجراءات التي تنظم تسلسل إجراء اتصال ما(مثل الدبلوماسية).
- **البروتوكولات في شبكات الكمبيوتر** : هي القواعد والإجراءات التقنية التي تسمح لمجموعة من الحواسب بالتعامل مع بعضها البعض عند اتحادها على الشبكة.

zayedalbadanee@yahoo.com

- **عمل البروتوكولات**

العملية الكاملة لنقل البيانات على الشبكة تمر بمجموعة من الخطوات، و في كل خطوة معينة تنفذ مهام محددة لا يمكن تنفيذها في خطوة أخرى، و لكل خطوة

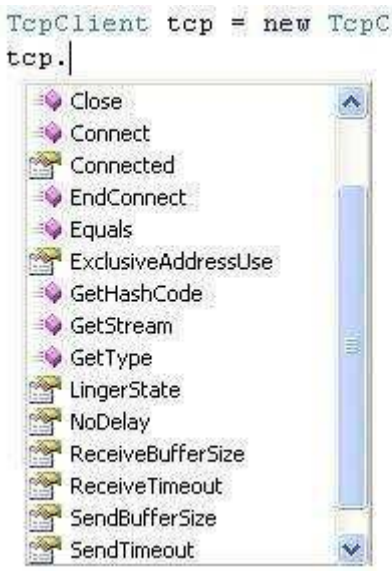
بروتوكول محدد أو مجموعة بروتوكولات تحدد كيفية تنفيذ المهام المتعلقة بهذه الخطوة، كما أن هذه الخطوات تكون متشابهة لكل جهاز على الشبكة، كما يجب ملاحظة أن الجهاز المرسل يقوم بإتباع هذه الخطوات من الأعلى إلى الأسفل بينما يقوم الجهاز المستقبل بإتباع هذه الخطوات بشكل معكوس من الأسفل إلى الأعلى.

سوف يتم الحديث في هذا الفصل عن أهم البروتوكولات المستخدمة في هذا النظام وهما بروتوكولي الـــ TCP و الـــ UDP ،حيث يتكون هذين البروتوكولين في بيئة الدوت نيت من العديد من الأصناف التي تسهل من استخدامهما وسوف نتطرق بالحديث عن أصناف هذين البروتوكولين

zayedalbadanee@yahoo.com

❖ مكونات أصناف بروتوكول الـــ TCP وبروتوكول الـــ UDP :

اولاً : الأصناف الخاصة ببروتوكول الـــ TCP :



الشكل(1-3) الدوال التابعة للصنف TcpClient

١) **الصنف TcpClient :**

٢) يستخدم هذا الصنف لتعريف Socket خاصة ببرتوكول الـ **TCP** اذ يمرر له عنوان الجهاز (IPAddress) والـ port للجهة التي نريد الاتصال ويحتوي هذا الصنف علي مجموعه من الدوال و الخصائص ونلخصها بالتالي :

- **دوال TCPClient :**
  - Connect : وتستخدم لأجراء عمليه الاتصال مع الـ server حيث نمرر فيها عنوان الIP الخاص بالserver ورقم الـ Port كما يلي:

```
TcpClient tcp = new TcpClient();
tcp.Connect(IPAddress.Parse("192.168.1.1"),5020);
```

  - Close :لإنهاء الاتصال مع الـ Socket الخاصة ببرتوكول الـ **TCP**.
  - GetStream ويستخدم لقراءة الـ Stream من الـ Socket في عمليةالإرسال والاستقبال وتستخدم عاده لربط الـ Socket مع صنف الـ NetworkStream

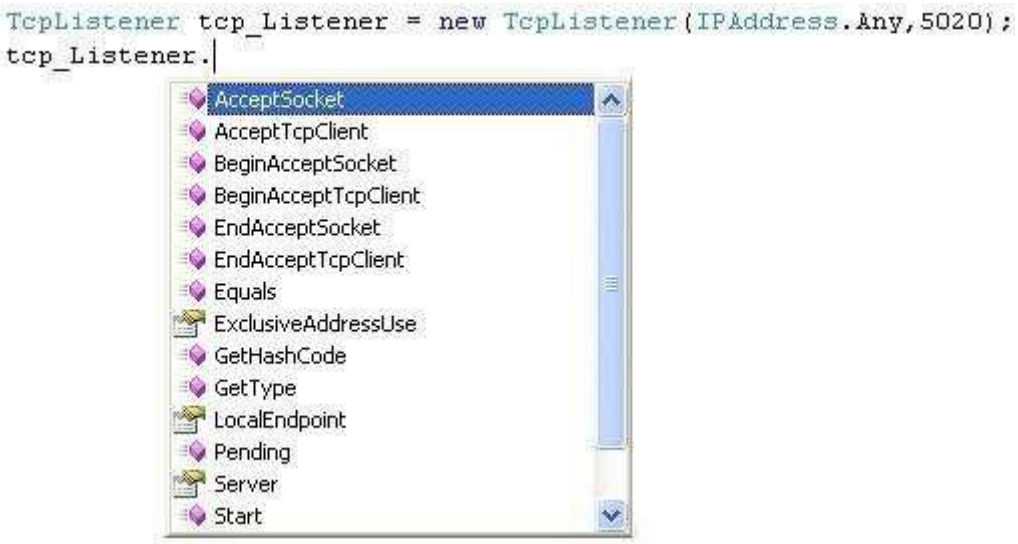
- **أهم خصائص TCPClient :**
  - Linger State : وتأخذ get أو Set لتحديد أو معرفة التأخر أو التباطؤ .
  - NoDelay : وتأخذ get أو Set لتحديد أو معرفة ما إذا كان هناك وقت معين للتأخير أم لا.
  - ReceiveBufferSize و SendBufferSize : وتأخذ get أو Set لتحديد أو معرفة حجم الـ Buffer المستخدم في الstream والمعرف

في TCP Client Object.

- ReceiveBufferTimeOut و SendBufferTimeOut : وتأخذ get أو Set لتحديد أو معرفة الوقت المتاح لعملية الإرسال أو الاستقبال حيث يعطي (Time Out) في حالة أنة لم يجد الطرف الأخر خلال فترة منية معينة .

٣) الصنف TcpListener :

حيث يحتوي على مجموعة من الدوال والخواص و هي كما يلي :



الشكل(2-3) الدوال التابعة للصنف TcpListener

zayedalbadanee@yahoo.com

- **أهم دول TcpListener:**

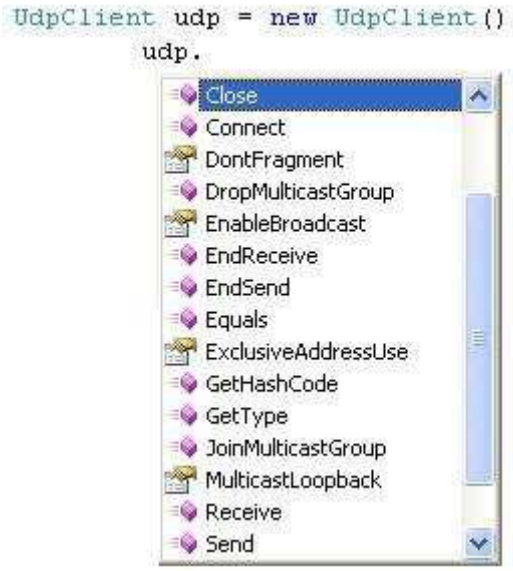
- Accept Socket : وتستخدم لقبول عملية الاتصال مع الـــ Client.
- Start : وهي طريقة تحميل (Overloaded Method) حيث انه في حالة تمرير رقم إليها يتم تحديد عدد الأجهزة التي يسمح بالاتصال معها ، وفي حالة عدم تمرير رقم معين يصبح عدد الأجهزة غير محدد.
- Stop : وتستخدم لإغلاق عملية التصنت ويفضل وضعها في نهاية الـــ Try

والـــ Catch وحتي يتم إنهاء عملية التصنت حتى لو حدث أي خطاء.

- **أهم الخصائص في TcpListener:**

- LocalEndpoint : حيث يرجع الـــ IP ورقم الـــ Port المستخدم في الطرف المراد الاتصال معه.
- Server : ومن خلالها نستطيع الوصول إلى كل الخصائص والدوال في الـــ Server الخاص ببروتوكول TCP مثل ال Accept وال Sendto وReceive والـــ Listen وغيرها.

ثانياً الأصناف الخاصة ببروتوكول الـــ UDP :



zayedalbadanee@yahoo.com

الشكل(3-3) الدوال التابعة للصنف UDPClient

ومن أهم الدوال والخصائص الخاصة بها :

- JoinMulticastGroup و DropMulticastGroup : لضم أو إلغاء

عنوان أو مجموعة .

- MulticastOption : ويستخدم في المجموعات حيث يتم تخزين قائمة عناوين IP لتعامل معها في المجموعات لعمل انضمام أو إلغاء انضمام لأي مجموعة.

❖ طبقات الشبكات والـــ Socket المتزامنة:

في هذا الجزء سوف يتم تقديم برمجة طبقات الـــNetwork Layer بشكل أكثر تفصيلا والـــ Socket هي التي يتم التعامل معها لإرسال واستقبال البيانات بعد تحويلها من والى Stream عبر الشبكة .

▪ مقدمة عن برمجة الـــ Socket :

من المعروف أن الـــ Socket هي الأداة التي يتم نقل البيانات من خلالها من جهاز إلى آخر ولاستخدامها يلزم في البداية تضمين المكتبة(System.Net.Socket) الموجودة في بيئة الدوت نت حيث تحتوي هذه المكتبة على عدد كبير من الأصناف والتي يتم استخدامها في برمجيات الشبكة وسوف يتم الحديث عن أهمها وهو الصنف Socket إذ يمكننا من التعامل مع الـــTCP أوالـــUDP أو مع أي نوع آخر من البرتوكولات بشكل مباشر ويتكون كائن الـــ Socket من ثلاثة باروميترات كما يلي :

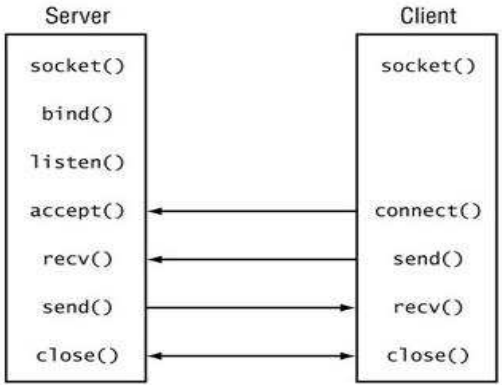
```
Socket MySocket = new Socket (AddressFamily,SocketType.,
ProtocolType);
```

حيث يتم في البارومتر الأول تحديد نوعية الـــ IP Address ، وفي البارومتر الثاني يتم تحديد نوع الـــ Socket أي هل سوف نستخدم Stream أو Dgram لإرسال البيانات وبشكل دائم يتم استخدام الـــ Stream في حالة استخدام الـــ TCP والـــ Dgram في

حالة استخدام الـــ UDP وأخيرا يحدد نوع البروتوكول المستخدم للاتصال هل هو من النوع UDP أو TCP .

- **استخدام برمجة الـــ Socket المتزامنة لإنشاء اتصال عبر بروتوكول الـــ TCP :**

تمر عملية الاتصال باستخدام الاتصال عبر الـــ TCP Socket بجموعه من المراحل كما في الشكل التالي:



الشكل(4-3) المراحل التي تمر بها الـــ TCP Socket

إذ تبدأ العملية في الـــ Client والـــ server بإنشاء الـــ Socket كما يلي :

```
Socket MySocket = new Socket(AddressFamily.InterNetwork,
SocketType. Dgram, ProtocolType.Udp);
```

ثم ربط الـــ Socket مع الكمبيوتر الحالي باستخدام الدالة Bind وتستخدم فقط عند الاستقبال كما يلي:

```
IPEndPoint ip = new IPEndPoint(IPAddress.Any, 5020);
MySocket.Bind(ip)
```

ثم القيام بعملية التصنت على Port المحدد باستخدام الدالة listener ويمكنك تحديد عدد الأجهزة التي سيتم قبولها ولوضع عدد غير محدد نمرر له الرقم -1 ثم نقوم بالموافقة على الاتصال باستخدام الدالة accept كما يلي:

```
MySocket.Listen(-1);
MySocket.Accept();
```

ويتم استقبال البيانات من خلال الدالة Receive حيث تعبئ البيانات في مصفوفة من النوع كما يلي :

```
byte[]Received=new byte[1024];
MySocket.Receive(Received);
```

وهنا تم إنشاء اتصال من النوع TCP وبتعريفها على الـــ Port(5020 كمثال) حيث يتم ربطها بالـــ Socket باستخدام الدالة Bind حيث تم تعريف Listen لا نهائي العدد-1 ..

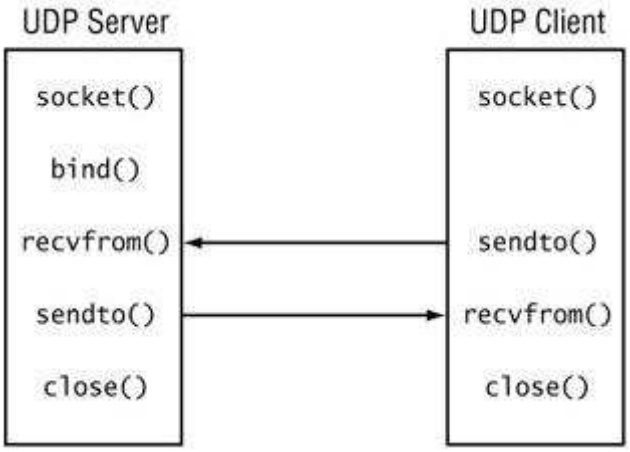
ولتعريف برنامج الإرسال TCP Client باستخدام الـــ Socket لا بد من تعريف الـــ Socket وإسناد عنوان الـــServer ورقم الـــ Port ثم إرسال البيانات باستخدام الدالة Send وتتم عملية الإرسال بما تم تعريفه في الـــsocket حيث سيستخدم Stream Socket كما يلي :

zayedalbadanee@yahoo.com

```
String str = Console.ReadLine();
ASCIIEncoding amen = new ASCIIEncoding();
byte[] msg = asen.GetBytes(str);
Socket MySocket = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
IPEndPoint remote = new
IPEndPoint(IPAddress.Parse("192.168.1.101"), 5020);
MySocket.Connect(remote);
MySocket.Send(msg);
MySocket.Close();
```

- **استخدام برمجة الـ Socket المتزامنة لإنشاء اتصال عبر بروتوكول الـ UDP :**

تمر عملية الاتصال باستخدام الاتصال عبر الـ UDP Socket بمجموعة من المراحل كما في الشكل التالي :



الشكل(5-3) المراحل التي تمر بها الـ UDP Socket

وتشبه عملية استخدام الـ UDP الـ TCP إلى حد كبير إذ تبدأ العملية في الـ Client والـ Server بإنشاء الـ Socket كما يلي :

```
Socket MySocket = new Socket(AddressFamily.InterNetwork,
SocketType. Dgram, ProtocolType.Udp);
```

ثم ربط الـ Socket مع الكمبيوتر الحالي باستخدام الدالة Bind وتستخدم فقط عند الاستقبال وكما يلي :

```
IPEndPoint sender = new IPEndPoint(IPAddress.Any,5020);
MySocket.Bind(sender);
```

ولاستقبال البيانات نستخدم الدالة Receive From حيث نعرف في البداية(Endpoint) الجهاز المراد الاتصال به بناء على ما تم تعريفه في السابق ونمرره مع مصفوفة الـــ Byte إلى الدالة Receive From ومن ثم نستطيع تحميل المصفوفة إلى السلسة النصية من خلال الدالة Get String الموجودة ضمن الصنف ASCII كما يلي :

```
int recv;
byte[] data = new byte[1024];
Endpoint Remote = (Endpoint) (sender);
recv = newsock.ReceiveFrom(data, ref Remote);
Console.WriteLine(Encoding.ASCII.GetString(data,0,recv));
```

ويتم في الإرسال استخدام الدالة Send To حيث نمرر لها البيانات بعد تحويلها من نص إلى مصفوفة من نوع Byte وحجم البيانات المرسلة إذ يمكننا معرفته من خلال الخاصية Length وأخيرا نمرر له كائن الـــ Endpointوالذي يعرف الـــ IP Address والـــ Port لـــ Socket كما يلي كمثال :

```
string welcome = "Hello All";
data = Encoding.ASCII.GetBytes(welcome);
newsock.SendTo(data, data.Length, SocketFlags.Broadcast, Remote);
```

- **مكونات أصناف الـــ Socket المتزامنة :**

١) الصنف IPAddress : ويستخدم لتعريف الـــ IPAddress حيث يمكن إسناده الى الـــ IPEndPoint كمثال ويمكن أن نكون من خلاله مصفوفة من العناوين في حالة إرسال رسالة ما إلى مجموعة أما الصيغة العامة له فهي كما يلي :

```
IPAddress newaddress = IPAddress.Parse("192.168.1.1") ;
```

ويمكن الاختيار بين أ ربعة خيارات في تحديد العنوان وهي كما يلي :

- Any ويستخدم لتمثيل أي عنوان متاح على الشبكة (عند الاستقبال ويعني الاستقبال من الكل)
- Broadcast ويستخدم لتمثيل البث الإذاعي لجميع الأجهزة على الشبكة (حيث يقوم بعملية حساب إلـــIP Broadcast دون تدخل المستخدم، لمعرفة كيفية حساب الـــ Broadcas)
- Loopback ويستخدم لتمثيل العنوان المعروف للـــ loopback هو127.0.0.1

كما يدعم هذا الصنف مجموعة من الدوال وأهمها :

- Equals تستخدم هذه الدالة بشكل عام للمقارنة بين كائنين وهنا ستستخدم للمقارنة بين عنوانين وترجع True إذا كانا متشابهين وFalse إذا كانا مختلفين.
- الصنف IPEndPoint : حيث استخدم لتحديد العنوان والـــ Port للـــ Host والذي نريد الاتصال به والصيغة العامة له كما يلي:

```
IPEndPoint end = new IPEndPoint(IPAddress.Parse("192.168.1.1"), 5020);
```

مجموعة الخواص التي تدعم في الصنف Socket كما يلي :

- AddressFamily ويرجع مجموعة العناوين المعرفة على الـــ Socket
- Available ويرجع حجم البيانات الجاهزة للقراءة من الـــ Socket .
- Connected وتستخدم هذه الخاصية بكثرة لمعرفة إذا كان الـــ Socket متصل مع الـــ Remote Host.
- ProtocolType ويستخدم لمعرفة البروتوكول الذي يستخدم في الـــ Socket.

❖ برمجة الــ Socket الغير متزامنة :

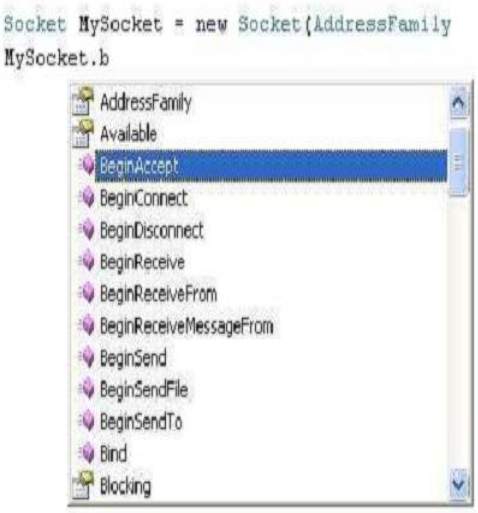
سوف يتم التحدث في هذا الجزء عن استخدام الــ Socket الغير متزامنة بشكل أكثر تفصيلاً عما تم شرحه سابقاً وسوف يتم تطبيق مجموعة من الأمثلة العملية على استخدام الاتصال الغير متزامن في برمجية الشبكات ..

من المعروف أن الاتصال المتزامن مهم جداً في البرمجيات التي تحتاج إلى العمل في الزمن الحقيقي حيث لا يسمح باستخدام الاتصال لأمر أخر إلا بعد انتهاء العملية الجارية واستخدامه مهم جداً في العمليات التي تتطلب مثل هذه الأمور لكن لا ينصح أبدا استخدامه في حالة اذا كان هنالك مجموعة كبيرة من المستخدمين تستخدم الــ Server حيث يمنع الأسلوب المتزامن بقية المستخدمين على الشبكة من إجراء عملية الإرسال في حالة كون الــ Server يستقبل بيانات من جهاز أخر وفي هذه الحالة ينصح باستخدام الاتصال الغير المتزامن إذ يعتبر مهم جداً في حالة إذا أردنا من البرنامج القيام بعدة مهام وعلى نفس ال Thread وباستخدام نفس الاتصال .

- **مكونات وأصناف الــ Socket غير المتزامنة :**

تدعم الدوت نيت الاتصال غير المتزامن بمجموعة من الطرق الموجودة ضمن أصناف الــ socket والتي يتم استدعائها من المكتبة(system.Net.Socket) وقد ميزت الدوت نيت هذه الطرق بوجود ال Begin في بداية اسم الدالة ولكل بداية طريقة يوجد نهاية طريقة مقابله لها وهي كما يلي:

zayedalbadanee@yahoo.com



الشكل(6-3) دوال الـــ Socket

BeginAccept : وتستخدم لقبول طلب الجهاز الأخر وباستخدام هذه الطريقة سوف يتمكن الـــ Server من استقبال عدد من عدد من الطلبات في نفس الوقت وبدون الحاجة لانتظار الانتهاء من العملية الجارية ويتم انهاء كائن الـــ client عن طريق الدالة End Accept .

- **<u>تطبيقات عدم التزامن في الدوت نيت :</u>**

تمر عملية الاتصال الغير متزامن بمجموعة من المراحل تبداء بإنشاء كائن الـــ Socketبعد ذلك يتم تعريف بداية الاتصال لبدء الاتصال غير المتزامن على الـــ Socket حيث يتم إسناد كائن الـــ IPEndPoint وحالة الكائن لها و تبداء في هذه الحالة عملية الاتصال بالـــ Socket وبعد ذلك تمرر إلى الـــ BeginAccept لقبول الطلب حيث يتم قبول الطلب ويرسل إشارة إلى الجهاز الأخر ليعلمه فيها بقبول الجلسة وإمكانية البدء للإرسال ويستطيع الجهاز الأخر بعد الموافقة على الجلسة البدء بالإرسال باستخدام الدالة BeginSend ويستقبل الـــ Server الرسالة من الجهاز الأخر باستخدام الدالة BeginReceive وكما ذكر سابقاً فان لكل عملية Begin تقابلها الدالة End للاستعداد

لإجراء عملية أخرى على نفس الـــ Thread في البرنامج وهو ما ميز الاتصال غير المتزامن عن الاتصال المتزامن .

zayedalbadanee@yahoo.com

❖ **دوال مكتبة الـــ I /O :**

إن الهدف الرئيسي من إنشاء مكتبات الـــ Stream لتسهيل عملية نقل البيانات من مكان إلى أخر سواءً عبر الشبكة أو ضمن أو على نفس الجهاز كما هو الحال بالتعامل مع الملفات أو التعامل مع الطابعة أو أي طرفية أو أي جهاز أخر موصول بالكمبيوتر حيث تسهل علينا عملية تحويلها إلى مصفوفة من نوع Byte وإرسالها وهو ما حل الكثير من المشاكل التي كانت تواجه المبرمجين في التعامل مع البيانات الثنائية(Binary Data) .

يمكن التعامل مع الـــ Stream بأسلوبين المتزامن (Synchronous) والغير متزامن(Asynchronous) وبشكل افتراضي تعمل جميع الـــ IO Streams بالأسلوب المتزامن لكن العيب فيه هو تأثيره الشديد على أدائية النظام اذ يقوم بإغلاق وحدة تنفيذ الأوامر في المعالج المخصصة للبرنامج بحيث لا يسمح بتنفيذ أي أمر أخر إلا بعد الانتهاء من العملية الجارية ولا ينصح أبدا من استخدام الأسلوب المتزامن في حالة إذا كنت تتعامل مع أجهزة قراءة وكتابة بطيئة نسبياً مثل الـــ ÜFloppy Disk او Magnetic Tape لكنها مهمة جداً بالبرمجيات التي تعتمد على أنظمة الزمن الحقيقي (Real Time Systems) حيث أنها تعتمد الأسلوب المتزامن في عملية إرسال و استقبال البيانات وهو ما يمنع القيام بأي عملية أخرى إلى حين الانتهاء من تنفيذ الأمر ومن الأمثلة عليها أنظمة السحب والإيداع في الرصيد البنكي أو أنظمة حجز التذاكر أوشحن بطاقة الهاتف وغيرها .. طبعاً في حالة إذا كان برنامجك لا يحتاج إلى وجود الخواص السابقة عندها ينصح باستخدام الأسلوب الغير

متزامن (Asynchronous) حيث تستطيع من خلاله تنفيذ عمليات أخرى في وحدة المعالجة وبدون الحاجة لانتظار إنهاء العملية الجارية اذ يتم إنشاء thread منفصل لكل عملية طلب إدخال او إخراج مما لا يؤثر على أدائية النظام وينصح في استخدامه إذا كانت عملية القراءة أو الكتابة تجري من خلال أجهزة بطيئة نسبياً ويمكن تميز الدوال المتزامنة عن الغير متزامنة في الدوت نيت بوجود كلمة Begin أو End في بداية اسم الدالة الغير متزامن وكمثال عليها BeginWrite و BeginRead و EndWrite و EndRead .

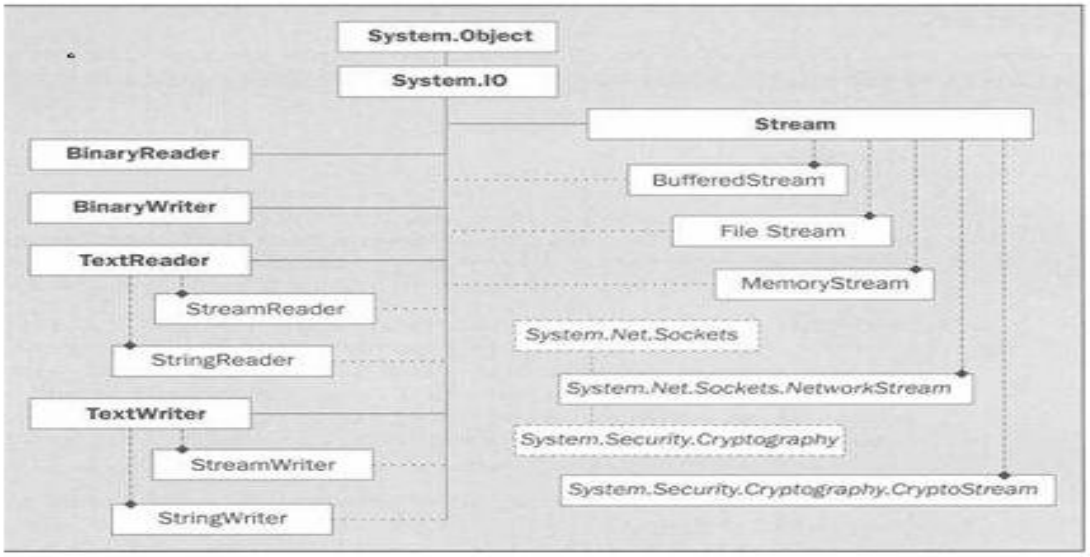
اولاً : أصناف الـــ Stream :

تدعم الدوت نت عمليات الـــ Streams بمجموعة من الأصناف والمندرجة تحت مكتبة الـــ System.IO والتي تستخدم لعمليات الإدخال و الإخراج لنقل البيانات .

ومن الأمثلة عليها الـــ FileStream و الـــ Buffered tream و الـــ MemoryStream و الـــ NetworkStream والتي تستخدم لنقل الـــ Stream عبر الشبكة ، وتقسم أصناف الـــ Stream في الدوت نيت

كما في الشكل التالي:

zayedalbadanee@yahoo.com



الشكل(7-3) أهمية أصناف الــ Streaming في الدوت نت

١) الصنف BufferedStream :

ويستخدم بشكل أساسي لحجز مقدار معين من الذاكرة بشكل مؤقت لتنفيذ عملية معينة كما تستخدم بعض البرمجيات الــ Buffering لتحسين الأدائية حيث تكون كذاكرة وسيطة بين المعالجة والإرسال أو الاستقبال وكمثال عليها برمجيات الطباعة حيث تستخدم الطابعة ذاكرة وسيطة لتخزين البيانات المراد طباعتها بشكل مؤقت ،ويكمن الهدف الأساسي من استخدام الــ Buffering في العمليات التي يكون فيها المعالج أسرع من عمليات الإدخال والإخراج حيث يتم معالجة البيانات ووضعها في الــ Buffer في انتظار إرسالها وهو ما يساهم في تحسين الأدائية بشكل كبير ،ويستخدم الــ BufferedStream عادتاً في برمجيات الشبكات مع ال NetworkStream لتخزين البيانات المراد إرسالها عبر الشبكة في الذاكرة بشكل افتراضي يتم حجز 4096 bytes عند استخدام الــ BufferedStream ويمكن زياداتها أو تقليلها حسب الحاجة ..

٢) الصنف MemoryStream :

وهو شبيه بعملية الــ Buffring السابقة اذ يعتبر كحل جيد لتخزين البيانات بشكل مؤقت في الذاكرة كــ Stream Data قبل الإرسال أو الاستقبال حيث يغنيك عن تخزينها على شكل ملف مما يسرع العملية بشكل كبير .

٣) الصنف NetworkStream:

ويستخدم لتعامل مع الــ Stream لإرساله عبر الشبكة باستخدام الــ Socket ويتم استدعها من المكتبة System.Net.Socket ويفضل استخدام الصنف BufferedStream معه لتحسين الأداء .

٤) الصنف FileStream :

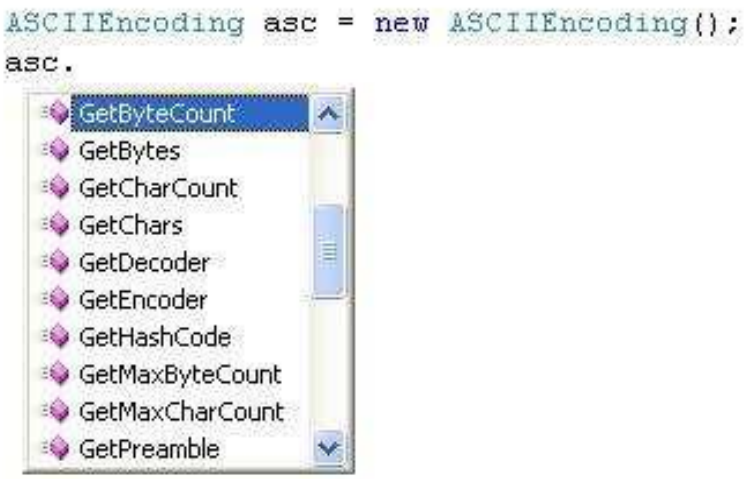
يتم استدعائه باستخدام المكتبة System.IO ويستخدم بشكل أساسي في التعامل مع الملفات سواءً للكتابة إلى ملف أو القراءة من ملف ويستخدم أيضا لتحويل الملف إلى مصفوفة بيانات ثنائيه و يسهل نقلة عبر الشبكة و من الأمور الهامة فيه تحديد مسار الملف المراد القراءة منة أو الكتابة علية .

وهناك بعض الدوال والتي تستخدم لإدارة الــ Stream وهي:

- Flush : وتستخدم لتفريغ محتويات الــ Buffer بعد إتمام العملية المحددة .
- Close : وتستخدم لإغلاق الــ Stream وتحرير المصادر المحجوزة من قبل الــ Stream Object
- zayedalbadanee@yahoo.com

ثانياً : داول Stream :

يمكن استخدام مكتبات الـــ Stream لنقل البيانات من نوع byte أو النصوص وفي العادة يتم استخدام الـــ BinaryReader والـــ BinaryWriter للتعامل مع البيانات الثنائية ويتم استخدام الـــ StreamReader و الـــ StreamWriter لتعامل مع النصوص ويتم استخدام الـــ ASCIIEncoding او الـــ UnicodeEncoding لتحويل من الـــ Stream إلى النصوص عند الاستقبال ومن النصوص إلى الـــ Stream عند الإرسال حيث تستخدم مجموعة من الدوال وهي كما في الشكل التالي :



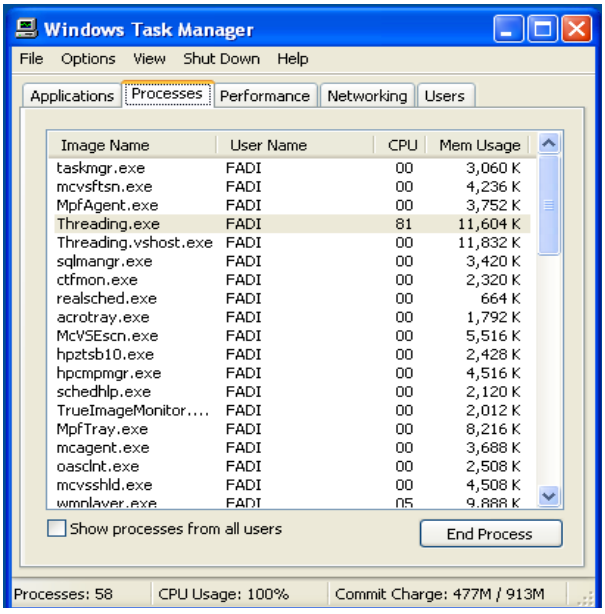
الشكل(8-3) دوال الـــ ASCIIEncoding

- GetByteCount : وهي طريقة تحميل زائد (Overloaded Method) حيث تأخذ مصفوفة رموز أو سلسلة رمزية وترجع عدد البايتات التي سوف نحتاجها لنقل نص معين .
- GetBytes : لتحويل السلاسل نصية إلى مصفوفة من نوع Byte حتى نستطيع إرسالها باستخدام الـــ Stream .

- GetCharCount: حيث تأخذ مصفوفة من نوع Byte وترجع عدد الأحرف التي سوف تكون في السلسلة الرمزية أو في مصفوفة رمزية .
- GetChars: وتستخدم لتحويل مصفوفة البيانات الثنائية الى سلسلة نصية وتستخدم عند استقبال البيانات من الـــ Stream حيث تحولها إلى نص مرة أخرى .

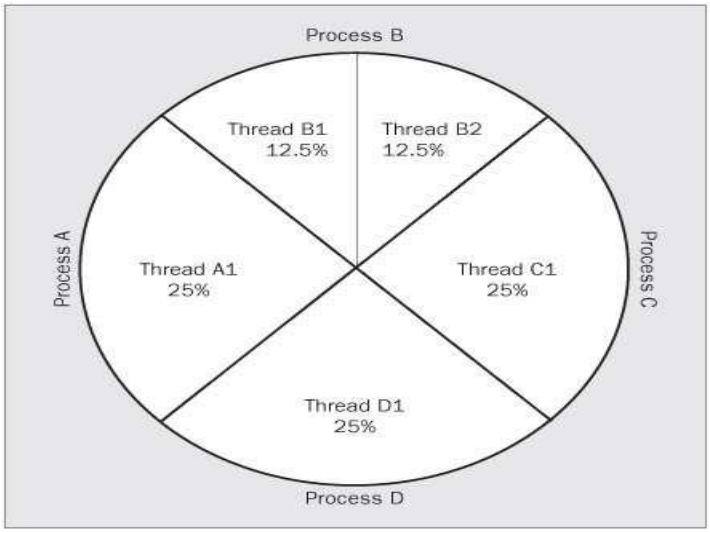
❖ مفهوم Threading:

عزز مفهوم الـــ Multithreading مفهوم الـــ Multitasking في أنظمة التشغيل بحيث أن أي برنامج يمكنه أن يعمل على Thread منفصل عن الآخر بالإضافة إلى إمكانية أن يعمل نفس البرنامج على أكثر من Thread أتى هذا المفهوم بعد تطوير فكرة الـــ Multitasking في أنظمة التشغيل ويتم إدارة هذه العمليات من قبل نظام التشغيل الذي يقوم بتقسيم المهام على المعالج وفق الأولويات لكل Thread وعلى فترات زمنية،ويمكن مشاهدة هذه العمليات في نظام التشغيل من خلال الـــ Task Manager كما في الشكل التالي :



الشكل(3-9) الـــ Task Manager

لاحظ أن كل برنامج يحجز مقدار معين من المعالج بناء على حاجته ويقوم نظام التشغيل بتقسيم المهام على المعالج وفق الحاجة والأولويات لاحظ الشكل التالي :



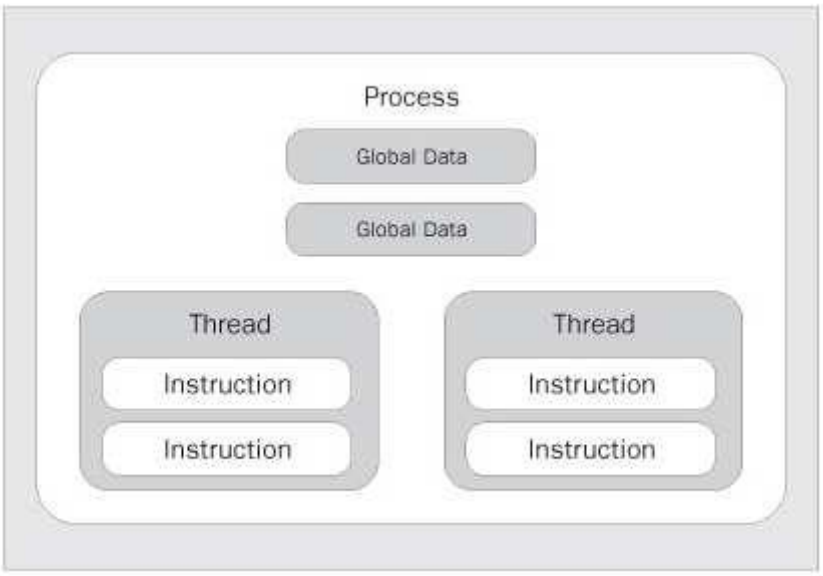
الشكل(10-3) أولويات التنفيذ في الــ CPU

وبالتأكيد يمكن التحكم بأي Thread في نظام التشغيل اذ يمكننا من عمل مقاطعة له كما يمكننا إيقافه بشكل مؤقت وأيضا إلغائه بعمل إزالة له .كل هذه العمليات متاحة لدى المستخدم (User) للقيام بها وأيضا متاح استخدامها برمجيا.

- **الــ Threading في .Net**

لقد تم استخدام الــ Threading في برمجيات الشبكات ولوحظ انه لا يمكن أن يستقبل البرنامج أي شيء في حالة كونه يعمل على نفس الــ Thread الخاص بال Form مما يمنع تنفيذ أي عملية أخرى إلا بعد انتهاء العملية الجارية وهو ما لن يحدث أبدا اذ انه تم استخدام حلقة لانهائية لبرنامج الاستقبال مما سيؤدي إلى عدم تنفيذ أي أمر آخر وسوف يتوقف البرنامج عن الاستجابة .و الحل لهذه المشكلة هو أحد خيارين :

- الابتعاد عن البرمجة المتزامنة (Synchronous Programming) والتوجه نحو البرمجة غير المتزامنة(Asynchronous Programming) .
- فصل الحلقة اللانهائية عن الـــ Thread الخاص بالبرنامج وجعله يعمل على Thread آخر منفصل وهذا ما يسمى بالـــ Multithreading حيث سيتم تقسيم المعالجة على الـــ Thread الخاص بالبرنامج والـــ Thread الذي تم إنشائه لاحظ الشكل التالي :



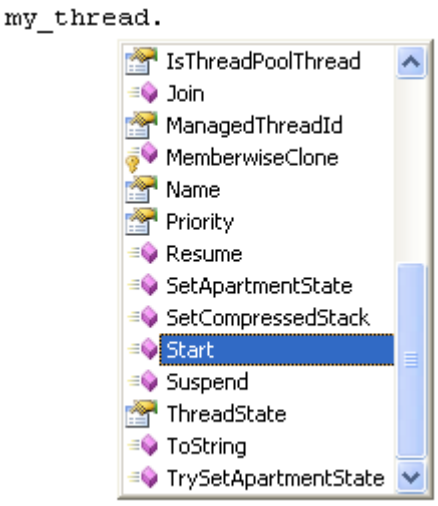
الشكل(11-3) تنفيذ المعالجات بشكل متوازي

zayedalbadanee@yahoo.com

وسيعمل في هذه الحالة بشكل متوازي من البرنامج وليس بشكل تسلسلي كما في السابق ولاستخدام الـــ Threading في الدوت نت يجب أولا تعريف مكتبة الـــ System.Threading ولتعريف Thread جديد نقوم باشتقاق كائن جديد من

الصنف Thread ثم نمرر اسم الدالة التي نريد عمل الـــ Thread لها يمكن عمل كل ذلك مباشرة وبخطوة واحدة كما يلي :

```
Thread my_thread = new (new ThreadStart (my_method));
```



الشكل(12-3) الدوال التابعة للصنف Thread

zayedalbadanee@yahoo.com

لاحظ أنه يمكن التحكم بجميع العمليات التي تخص الـــ Threading من خلال كائن Threadالذي تم اشتقاقه ومن أهم هذه العمليات الـــ Start والـــ Stop والتي تستخدم لتفعيل وإيقاف Thread وكما نستخدم الـــ Abort لإلغاء الـــ Thread نهائيا من المعالجة .

أولا:الطرق التابعة للصنف Thread :

- Start وتستخدم لبدء الـــ Thread في كل مرة يتم استدعائها يتم تنفيذ Thread آخر لنفس الطريقة التي تم تمريرها .
- Join وتستخدم للتحكم بالفترة التي سيتم فيها إنهاء الـــ Thread عند عمل إزالة له .

- Suspend وتستخدم لإيقاف الــ Thread الحالي بشكل مؤقت ولا نحصل على أي خطأ في حالة كان الــ Thread في وضع التوقف .
- Abort وتستخدم لإيقاف الــ Thread بشكل نهائي وإلغائه من الذاكرة ويفضل وضع هذه الطريقة في حدث الإغلاق الخاص بالبرنامج لكي لا يبقى الــ Thread في الذاكرة .
- Resume وتستخدم لإرجاع تنفيذ الــ Thread بعد ايقافه بشكل مؤقت باستخدام الــ Suspend .
- Sleep وتستخدم لإيقاف الــ Thread فترة زمنية معينة .

ثانيا : أهم الخواص التابعة للصنف Thread

- Name تأخذ get أو set لتحديد أو معرفة اسم الــ Thread الذي تم إنشاؤه والهدف منها تسهيل التعامل مع الــ Multithreads في البرنامج .
- CurrentCulture تأخذ get أو set لتحديد أو معرفة اللغة المعرفة لهذا الــ Thread.
- CurrentUICulture و تأخذ get لمعرفة اللغة المستخدمة في نظام التشغيل .
- IsAlive و تأخذ get لمعرفة إذا كان الــ Thread الحالي قيد التشغيل ام لا .
- ManagedThreadId و تأخذ get لمعرفة الرقم الوحيد الخاص بالــ Thread الحالي .
- Priority تأخذ get أو set لتحديد أو معرفة الأولوية للــThread الحالي وفي الوضع الطبيعي تكون Normal ويمكن تغييرها كما يلي :

الشكل(13-3) تغيير أولوية تنفيذ الـــ Thread في الدوت نت

zayedalbadanee@yahoo.com

حيث يحدد فيها مدى أهميتها بالنسبة لأولوية المعالجة

- ThreadState و تأخذ get لمعرفة حالة الـــ Thread الحالي هل هو يعمل أم لا

وترجع False أو True .

- **استخدام الـــ Multithreading في برمجيات الشبكات :**

ينصح باستخدام الـــ Multithreading في برمجيات الشبكات التي تعتمد البرمجة المتزامنة Synchronous Programming اذ أن طبيعة التنفيذ في هذه البرمجيات متسلسلة وهو ما سوف يمنع من تنفيذ عمليات أخرى لحين الانتهاء من العملية الجارية ، اذ أنه من المفضل استخدام Thread منفصل لكل عملية قد تطرنا إلى الانتظار فترة طويلة أو أن العملية ستبقى في وضع التنفيذ طيلة فترة تشغيل البرنامج .

١. استخدام الـــ Threading في برمجيات الزبون Clients :

- استخدام دالة الربط Connec في الاتصال المتزامن : في حالة استخدام الدالة Connec وفي حالة كان الـــ Server بطيء فان عملية الموافقة ستأخذ بعض الوقت مما سيؤدي إلى توقف البرنامج عن الاستجابة لحين إتمام عملية القبول ويفضل في هذه الحالة استخدام منفصل للقيام بعملية الاتصال .

- استخدام الـــ Client في عملية الإرسال والاستقبال المتزامن ، يفضل في حالة الإرسال المتزامن عدم استخدام منفصل لكل عملية وخاصة إذا كان الهدف من البرنامج إرسال مجموعة من التعليمات المختلفة إلى نفس الـــ Server حيث سيكون منشغل باستقبال التعليمات التي يتم إرسالها ، ويفضل في هذه الحالة استخدام البرمجة غير المتزامنة إذا كانت الإرسال لأكثر من تعليمة ولنفس الجهة أما في حالة الاستقبال المتزامن فلا بد من استخدام الـــ Thread لأن عدم استخدام الـــ Thread سيؤدي إلى توقف البرنامج عن الاستجابة .

٢. استخدام الـــ Threading في برمجيات الخادم Server :

- استخدام دالة القبول AcceptSocket ، لابد من استخدام الـــ Thread منفصل في حالة استخدام دالة القبول AcceptSocket في البرامج المتزامنة إذ انه سيتوقف البرنامج عن الاستجابة في حالة عدم استخدام الـــ Thread مع الـــ AcceptSocket حيث توضع عادتا في حلقة لانهائية .

حالة خاصة :

قد تضطر إلى اللجوء إلى البرمجة غير المتزامنة في حالة إذا كان الـــ Server سيستقبل أكثر من طلب من قبل الـــ Client في نفس الوقت مما سيؤدي إلى قبول أول طلب ورفض البقية وهذا أمر غير فعال وخاصة إذا كان البرنامج سيعمل على الانترنت وان عدد الـــ Clients وأوقات دخولهم غير معروفة ، وفي هذه الحالة لا بد من اللجوء إلى البرمجة غير المتزامنة(asynchronous Programming) لإتمام عملية القبول . ويجب الانتباه إلى أمر هام في الـــ (Threading A port) وهو إغلاق كل ما له

علاقة بالـــ Socket مثل الـــTcplistener قبل إغلاق الـــ Thread في حدث إغلاق البرنامج .

zayedalbadanee@yahoo.com

الفصل الرابع

تصميم النظام(1)

عملية إرسال النص

عملية استقبال النص

عملية إرسال النص لمجموعة عبر بروتوكولUDB

عملية استقبال نص عبر بروتوكولUDB

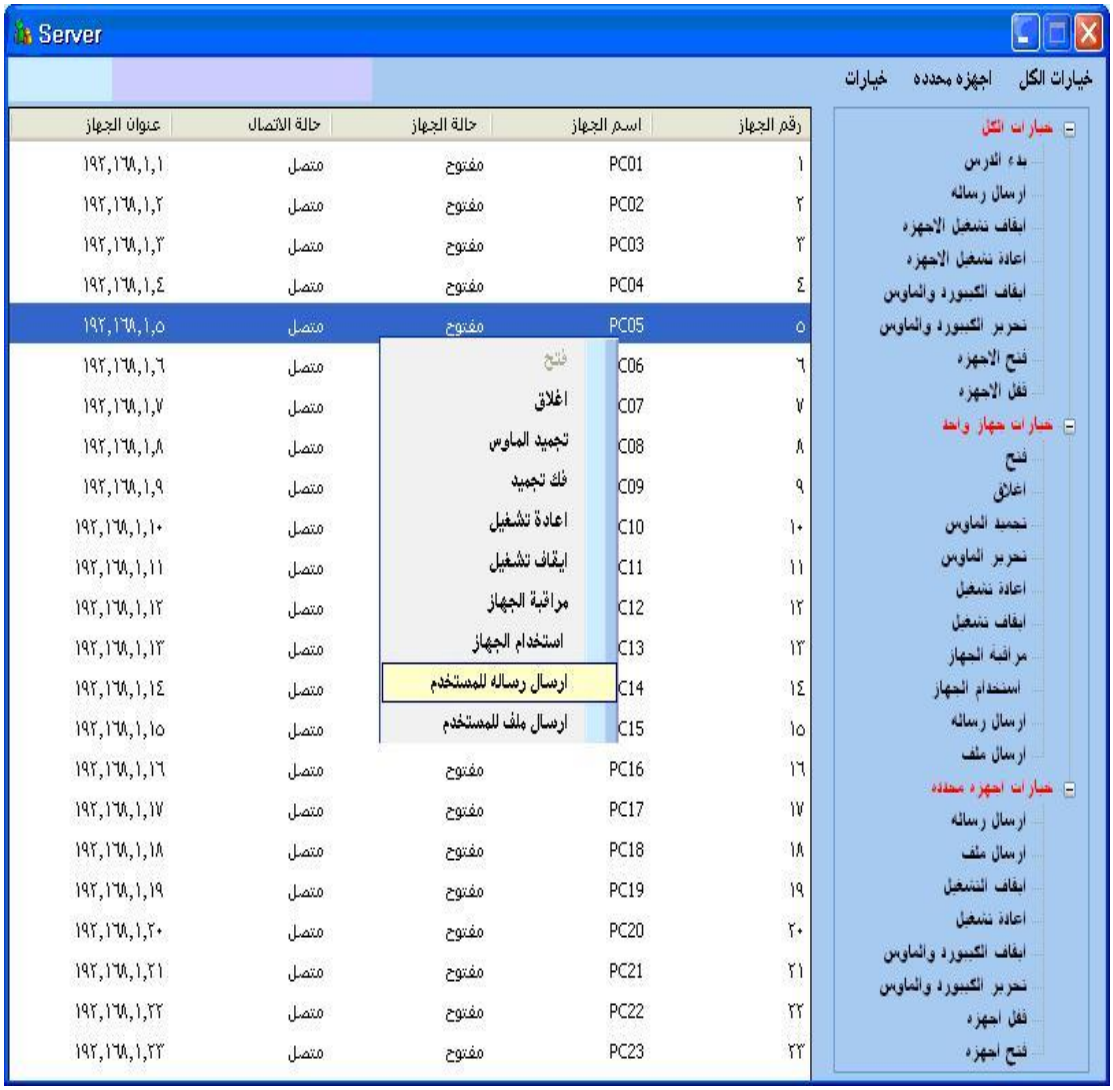
عملية إرسال الملفات

عملية استقبال الملفات

عملية المراقبة

zayedalbadanee@yahoo.com

سيتم في هذا الفصل شرح للعمليات الاساسية التي يؤديها النظام

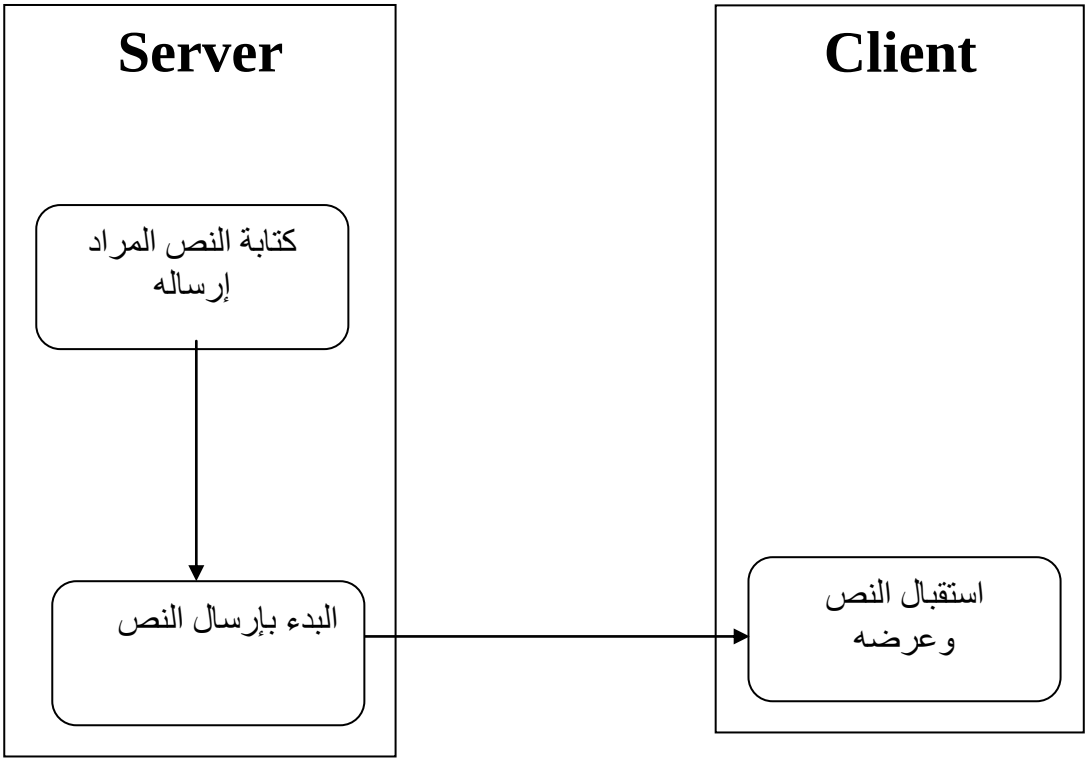


الشكل(4-1) العمليات الاساسية في النظام

zayedalbadanee@yahoo.com

عملية إرسال نص :

من ضمن الاحتياجات التي سيتم توفيرها في هذا المشروع عملية إرسال النصوص من جهاز إلى أخر . قد يحتاج المعلم إلى إرسال بعض النصوص (ملاحظات سؤال توجيه ،...)من جهازه إلى جهاز طالب أو أكثر .



الشكل(1-4) عملية إرسال رسالة إلى المستخدم

فيما يلي سيتم توضيح كيفية إتمام ذلك والكود البرمجي المستخدم الذي سيتم من خلاله إتمام هذه العملية (عملية إرسال النص).

في البداية سيتم تعريف دالة send وهي الدالة التي ستقوم بعملية الإرسال حيث سيتم استدعاء هذه الدالة عند الحاجة لها .

zayedalbadanee@yahoo.com

الكود البرمجي :

```
void send(string msg,string ip) // overloading method
    {
      try
      {
        TcpClient myclient = new TcpClient(ip, 8000);
        NetworkStream myns = myclient.GetStream();
        StreamWriter mysw = new StreamWriter(myns);
        mysw.Write(msg);
        mysw.Close();
        myns.Close();
        myclient.Close();
      }
      catch (Exception ex)
      {
        MessageBox.Show(ex.Message);
      }
```

هنا تم بناء الدالة send التي تستقبل متغيرين من نوع string حيث المتغير msg عبارة عن النص المراد إرساله إما المتغير ip فيقصد به ipaddrees وهو يعني عنوان الجهاز المراد الإرسال إليه (أي الجهاز المستقبل).

والكائن Tcpclient هو الكائن المسئول عن عملية الإرسال. أما الرقم 8000 فهي قيمة الـــ port الذي يتم من خلاله إرسال النص إلى الجهاز الذي يستخدم نفس الـــ port أي أن الـــ port سيكون متساوي في جهاز الإرسال وجهاز الاستقبال .

بعد ذلك تم استخدام الـــ Networkstream باسم myns وهي عبارة عن ذاكره مؤقتة أي أنها ستقوم بخزن النص بشكل مؤقت ثم بعد ذلك سيتم حذفه .

أما الدالة Getstream فأنها ستقوم بتحويل هذا النص إلى النظام الثنائي وذلك لكي يتمكن الجهاز من إرسال هذا النص ويتم بعد ذلك خزنه في myns الذاكرة المؤقتة ، أما الدالة Mysw.writer فهي عبارة عن ذاكرة الكتابة التي ستقوم بالاحتفاظ بالنص–msg – عندما يقوم المستخدم بكتابته .

بعد الانتهاء من العمليات السابقة يتم إغلاق الكائن المسئول عن عملية كتابة النص إلى الذاكرة المؤقتة – StreamWriter – وايضاً يتم إغلاق الذاكرة المؤقتة .

عملية استقبال النص :

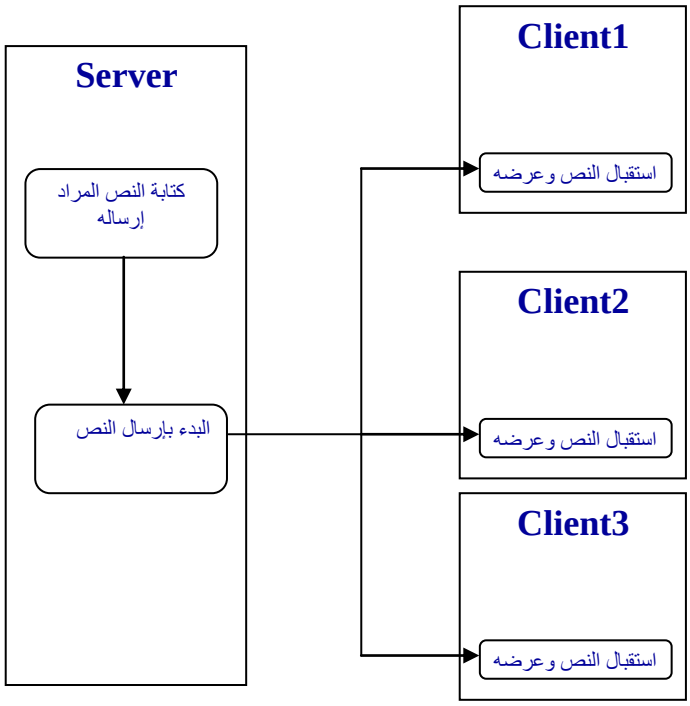
لاستقبال النص المرسل من قبل جهاز أخر لابد من تعريف متغير من نوع TcpListener يقوم هذا المتغير باستقبال البيانات من أي جهاز يقوم بإرسالها عبر الـــ port المحدد يتم خزن هذه البيانات في Network stream على شكل بيانات ثنائية ثم يقوم الـ stream reader بقراءتها من داخل الذاكرة وتحويلها إلى string ويتم خزنة في متغير من نوع string ثم يتم إرسال هذا النص إلى الدالة setText والتي تقوم بنقل المتغير من Thread إلى أخر

```
public void text_Receiver()
    {

        tcp = new TcpListener(IPAddress.Any,8000);
        tcp.Start();
        sock = tcp.AcceptSocket();
        ns = new NetworkStream(sock);
        sr = new StreamReader(ns);
        string x = sr.ReadToEnd();
```

```
        i++;
        this.SetText(x);
        MessageBox.Show(x, "وارده رسالة", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
        tcp.Stop();

        while (true)
        {
           text_Receiver();}


      private void SetText(string text)
    {
       if (this.dataGridView1.InvokeRequired)
       {
          SetTextCallback d = new SetTextCallback(SetText);
          this.Invoke(d, new object[] { text });
       }
       else
       {
          MessageBox.Show(text);
       }
    }
```

zayedalbadanee@yahoo.com

عملية إرسال نص لمجموعة عن طريق بروتوكول الUDP :



الشكل(2-4) عملية إرسال نص عبر المجموعات

في عملية إرسال النص عن طريق بروتوكول الـــ TCP كان لابد من تحديد عنوان الجهاز المرسل إليه النص أي أن بروتوكول الـــ TCP لا يدعم الإرسال إلى أكثر من جهاز في نفس اللحظة ولكننا استخدمنا هذا البروتوكول لما له من مميزات في دعم اللغة العربية ولسهولة استخدامه ولكن نجد أننا بحاجة إلى استخدام بروتوكول الـــ UDP لأننا بحاجة إلى التعامل مع أكثر من جهاز في نفس اللحظة مثل إرسال رسالة إلى كل المتصلين بالشبكة فبدل من إرسال هذه الرسالة إلى كل جهاز على حدة كما يحدث في بروتوكول الـــ TCP نقوم بإرسال هذه الرسالة إلى مجموعة فيستطيع كل جهاز في هذه المجموعة استلام هذه الرسالة بنفس اللحظة .

الكود البرمجي:

```
UnicodeEncoding cod = new UnicodeEncoding();
Socket server = new Socket(AddressFamily.InterNetwork,SocketType.Dgram,
ProtocolType.Udp);
IPEndPoint iep = new IPEndPoint(IPAddress.Parse("224.100.0.1"),5020);
byte[] data = cod.GetBytes(textBox1.Text);
server.SendTo(data, iep);
server.Close();
```

ومن خلال الكود السابق نلاحظ انه تم اولاً دعم ترميز اللغة العربية حيث أن بروتوكول الـــUDP لا يدعم اللغة العربية.

وبعد ذلك تم تعريف المتغير `server` من نوع `Socket` وياخذ ثلاثة بارمترات الأول `AddressFamily` ويمثل مجموعة العناوين المستخدمة في عملية الارسال الثاني `SocketType` يحدد اسلوب نقل البيانات علماً بان ال UDP يدعم الأسلوب `Dgram` .

الثالث `ProtocolType` يمثل نوع البروتوكول المستخدم أي بروتوكول UDP .

ومن ثم يتم تعريف الطرف الذي سيستقبل الرسالة وأيضا البورت المستخدم في عملية النقل ثم يتم تحويل النص المراد نقلة إلى مصفوفة من نوع `Byte` ثم يتم البدء بعملية نقل المصفوفة السابقة إلى المجموعة المحددة وبالبورت المحدد وبعد ذلك يتم إيقاف عملية النقل.

عملية استقبال نص عن طريق بروتوكول الUDP :

إن عملية استقبال النص عن طريق بروتوكول الـــ UDP هي عملية سهلة حيث في بداية هذه العملية يتم أولا تدعيم اللغة العربية ومن ثم يتم استقبال النص من البورت

2050 وبعد ذلك يتم ربط الجهاز مع المجموعة المحددة ثم يتم تعريف الطرف الذي تم استقبال النص منه بأنه يستقبل من أي جهاز ومن ثم يتم تعريف مصفوفة تستقبل البيانات القادمة من الأداة sockt وبعد ذلك يتم تحويل هذه المصفوفة إلى نص (string) وتخزينها في متغير string ومن ثم يتم تفريغ محتويات هذه المصفوفة .

الكود البرمجي:

```
void recive()
    {
      try
      {
        UnicodeEncoding cod = new UnicodeEncoding();
        sock = new UdpClient(5020);
        sock.JoinMulticastGroup(IPAddress.Parse("224.100.0.1"));
        IPEndPoint iep = new IPEndPoint(IPAddress.Any, 0);
        byte[] data = sock.Receive(ref iep);

        string s = Convert.ToString(cod.GetString(data));
        Array.Clear(data, 0, data.Length);
        sock.Close();
        MessageBox.Show(s);
        textBox1.Text = s;
        while (true)
        {
          recive();
        }
      }
      catch { };
}
          Thread myth2;
    private void Form1_Load(object sender, EventArgs e)
    {
      myth2 = new Thread(new System.Threading.ThreadStart(recive));
      myth2.Start();
}
```

أولا يتم دعم ترميز اللغة العربية لان بروتوكول الـــ UDP لا يدعم اللغة العربية .

ثانيا يتم تعريف متغير من نوع socket بحيث يقوم هذا المتغير بالاستقبال من الـــ port 5020 .ومن ثم يتم ربط الجهاز مع المجموعة المحددة بال

. (IPAddress.Parse("224.100.0.1")

ثم يتم تعريف الطرف الذي نستقبل منة الرسائل بحيث أنة يقوم بالاستقبال من أي جهاز.

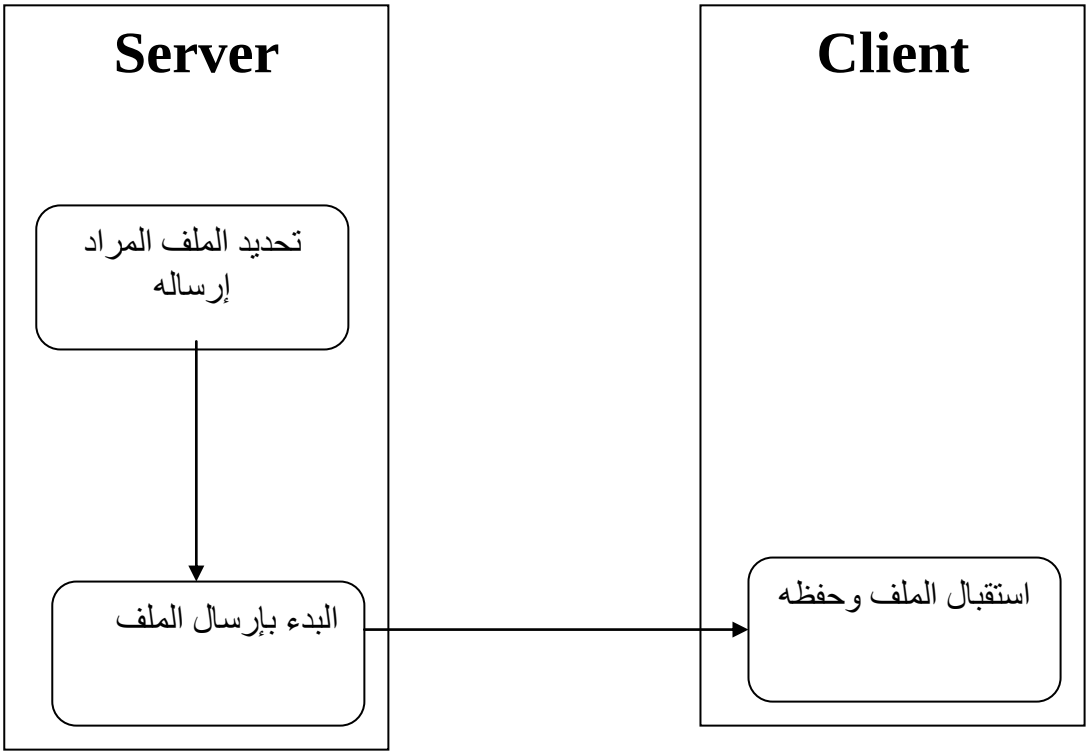
وبعد ذلك يتم تعريف مصفوفة من نوع byte تستقبل البيانات القادمة من الـــ sockt .

ثم يتم تعريف متغير من نوع string بحيث يقوم هذا المتغير بتحويل المصفوفة السابقة إلى string .

وبعد ذلك يتم بعد أن تم خزن المصفوفة داخل الـــ string تفريغ محتويات هذه المصفوفة ومن ثم يتم إيقاف الـــ sockt ،نلاحظ من خلال الكود السابق وجود لوب أو حلقة لانهائية تعمل على استقبال النصوص بشكل متتالي أو مستمر، حيث إن أي عملية استقبال يجب أن تقوم بالاستقبال بشكل لانهائي،وأخيرا يتم تعريف متغير من نوع Thread حيث وان أي عملية استقبال لابد إن تتم بواسطة الـــ thread وهذا الـــ thread هو عبارة عن عملية متزامنة تقوم بعمل تزامن داخل المعالج بحيث يمكن عمل عدة معالجات في وقت واحد.

عملية إرسال الملفات

إن عملية إرسال الملفات مشابه إلى حدٍ ما لعملية إرسال النصوص



الشكل(3-4) عملية إرسال الملف إلى المستخدم

zayedalbadanee@yahoo.com

```
private void send(string ip)
{
  try
  {
    FileStream fs = new FileStream(textBox1.Text, FileMode.Open);
    byte[] buffer = new byte[fs.Length];
    int len = (int)fs.Length;
    fs.Read(buffer, 0, len);
    fs.Close();

    BinaryFormatter br = new BinaryFormatter();
```

```
            TcpClient myclient = new TcpClient(ip, 7000);
            NetworkStream myns = myclient.GetStream();
            br.Serialize(myns, FileName);

            BinaryWriter mysw = new BinaryWriter(myns);
            mysw.Write(buffer);
            mysw.Close();

            myns.Close();
            myclient.Close();
            MessageBox.Show("تـــم ارســـال الـــملـــف...", "SuccessFully",
MessageBoxButtons.OK,          MessageBoxIcon.Exclamation);
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
                }
```

ليتم إرسال ملف عبر الشبكة لابد من تحويل هذا الملف إلى بيانات من نوع binary ليستطيع الجهاز التعامل مع هذه البيانات .

بعد ذلك يتم تعريف ذاكرة مؤقتة FileStream لتخزين بيانات الملف ثم يتم تحويل الملف إلى بيانات من نوع byte وخزنها داخل مصفوفة من نوع byte ثم يتم نقل محتويات هذه المصفوفة إلى الذاكرة المؤقتة NetworkStream .

وبهذا تتم عملية إرسال ملف وعندها يتم إغلاق الـــ NetworkStream و الـــTcpClient لتتم عملية إرسال أخرى .

عملية استقبال الملفات:

إن عملية استقبال الملفات هي عملية مشابه إلى حدٍ كبير لعملية استقبال النص ويوجد بعض الاختلافات كما سنلاحظ من خلال الكود التالي:

كود الاستقبال :

```
NetworkStream myns;
TcpListener mytcpl;
Socket mysocket;
Thread myth;
BinaryReader bb;
int x = 0;
#region File_Recriver()
void File_Receiver()
{
   try{
      mytcpl = new TcpListener(IPAddress.Any,7000);
      mytcpl.Start();
      mysocket = mytcpl.AcceptSocket();
      myns = new NetworkStream(mysocket);
      BinaryFormatter br = new BinaryFormatter();
      object op;
      op = br.Deserialize(myns); // Deserialize the Object from Stream
      x = 0;
      bb = new BinaryReader(myns);
      byte[] buffer = bb.ReadBytes(50000000);
      FileStream fss = new FileStream(@textBox1.Text + (string)op,
FileMode.CreateNew, FileAccess.Write);
      fss.Write(buffer, 0, buffer.Length);
      fss.Close();
      mytcpl.Stop();
      MessageBox.Show((string)op+" الـمـلـف اسـتـلام تـم "'+" '", "وارد ملف",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
```

```
        if (mysocket.Connected == true)
        {
          while (true)
          {
            File_Receiver();
          }
        }
      }
}
```

في عملية استقبال الملفات تبدأ عملية التصنت على نفس الـــ port الموجود في دالة الاستقبال ويتم أيضا تفعيل الـــ socket الذي سيقوم بالاستقبال وستبدأ عملية الاستقبال كالتالي :

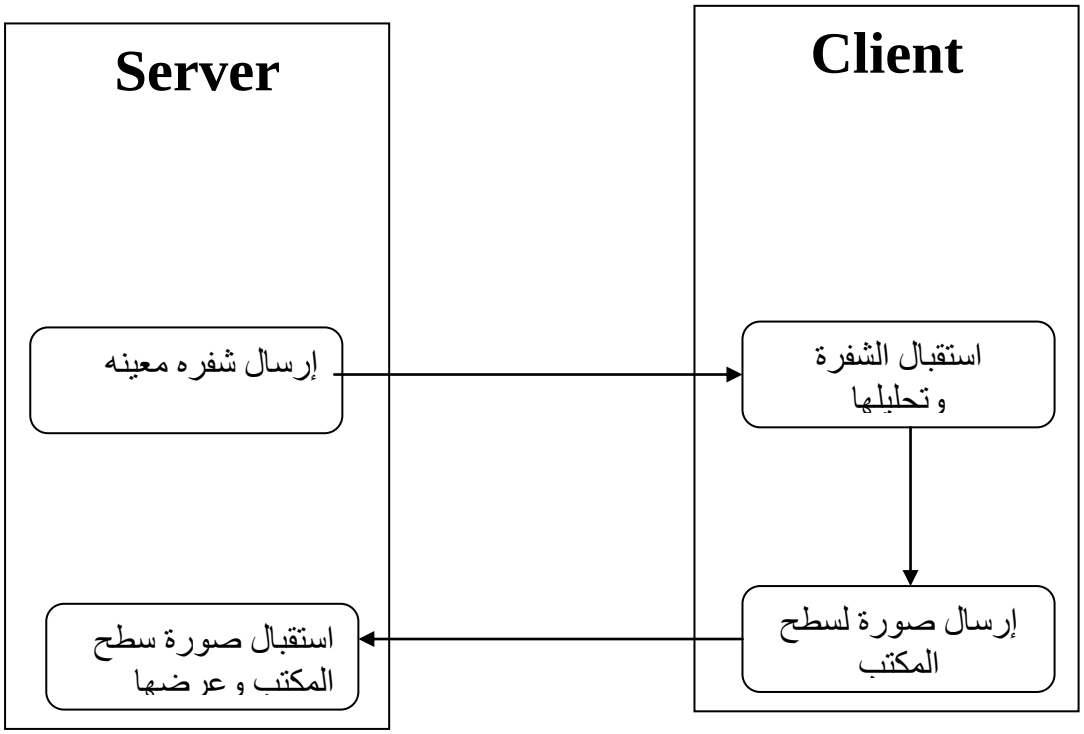
ستقوم الـــNetworkStream باستقبال البيانات المرسلة على شكل بيانات بالنظام الثنائي ثم يتم إنشاء متغير من نوع object يقوم باستقبال محتويات الملف مع أسمه من داخل الـــNetworkStream وتبدأ عملية القراءة من الـــNetworkStream بواسطة الـــ BinaryReader ويتم بعد ذلك تخزين ما داخل الـــBinaryReader إلى مصفوفة من نوع byte وعندها يتم إعطاء الـــFileStream معلمات الحفظ لمسار الملف واسمه وإنشاء ملف جديد لتتم عملية الكتابة من المصفوفة إلى داخل هذا الملف .

وعد الانتهاء من العمليات السابقة يتم إغلاق الFileStream وإيقاف المتصنت ليكون مستعد لعملية استقبال أخرى.

عملية مراقبة الأجهزة :-

عملية مراقبة الأجهزة هي إحدى العمليات الأساسية في هذا النظام حيث ومن خلاله يتم مراقبة الأجهزة التي يعمل بها الطلاب من قبل الجهاز المتحكم

(جهاز المعلم)، حيث أن هذه العملية توفر الكثير من الجهد للمعلم حيث يستطيع مراقبة الأجهزة عن بعد عن طريق جهازه الرئيسي.



الشكل(4-4) عملية مراقبة جهاز المستخدم

أولا الجهاز المتحكم :

الكود البرمجي:

```
void rec()
    {
      try
      {
        tcp1.Start();
        ns1 = new NetworkStream(soc);
        pictureBox1.Image = Image.FromStream(ns1);
        tcp1.Stop();
        while (true)
```

```
        {
            rec();
        }
    }
    catch {};
}
```

حيث وان هذه الدالة تعمل على استقبال الصور التي سيتم إرسالها من الجهاز الذي يتم مراقبته وعرضها .

أولا يتم البدء بعملية التنصت (البدء بعملية المراقبة)ثانيا يتم تعريف متغير من نوع Socket وبعد ذلك يتم اخذ أي شيئ يلتقطة المتنصت.

من ثم يتم تعريف متغير من نوع NetworkStream وتخزن بداخلها ما تم التقاطه من قبل المتنصت Stream.

وتْظهر الصور التي تم تخزينها في الذاكرة المؤقتة الخاصة بالشبكات و التي تم التقاطها من قبل المتنصت والتي تظهر ما يقوم به الجهاز المراقب من عمليات داخل كائن الـــ PictureBox.Imag .وبعد ذلك يتم إيقاف عملية التنصت.

ونلاحظ من خلال الكود السابق لوب لانهائي أو حلقة لانهائية فائدتها أنها تظل تستدعي الدالة `rec();` دائماً وبشكل لانهائي لالتقاط الصور .

ملاحظة :-

لابد أن تتم عملية الاستقبال بواسطة الـــ thread وهذا الـــ thread يقوم بعمل تزامن داخل المعالج بحيث يمكن عمل عدة معالجات في وقت واحد ، وذلك

لان عملية الاستقبال تكون بصورة لانهائية وأيضا هنا في عملية التنصت تكون عملية التنصت بصورة لانهائية.

ثانياً الجهاز المراقب :-

وهو الجهاز الذي يراد مراقبته من قبل الجهاز المتحكم لكي تتم عملية مراقبة الجهاز من قبل الجهاز المتحكم لابد أن يتم إرسال صوره من جهاز المراقب إلى جهاز المتحكم بصوره متتالية لهذا نلجأ إلى استخدام timer حيث يقوم هذا الـــ timer بأخذ صوره لسطح المكتب وإرسالها إلى الجهاز المتحكم حيث أن الـــ timer يحجز ذاكرة مؤقتة في الـــ RAM ليتم خزن صورة سطح المكتب قبل تحويلها إلى مصفوفة من نوع byte لكي يتم اخذ صورة لسطح المكتب لابد أن نستخدم الـــ Capture Screen Reference والموجود ضمن دوال الـــ API والتي توجد فية دالة الـــ `GetDesktopImage()` وهي الدالة التي تقوم بأخذ صورة لسطح المكتب وبعد احضار صورة لسطح المكتب يتم خزنها في الذاكرة المؤقتة `MemoryStream()` وتحويلها الى مصفوفة من نوع `byte` ومن ثم ارسالها الى الجهازالمتحكم الذي يستقبلها بشكل `byte` ثم يحولها الى صورة.

zayedalbadanee@yahoo.com

الكود البرمجي:

```
TcpClient myclient;
MemoryStream ms;
NetworkStream myns;
BinaryWriter mysw;

private void timer1_Tick(object sender, EventArgs e)
{
   try
   {
      ms = new MemoryStream();
      CaptureScreen.CaptureScreen.GetDesktopImage().Save(ms,
System.Drawing.Imaging.ImageFormat.Jpeg);
      byte[] arrayimage = ms.GetBuffer();

      myclient = new TcpClient("localhost", 2000);

      myns = myclient.GetStream();
      mysw = new BinaryWriter(myns);
      mysw.Write(arrayimage);
      ms.Flush();
      myns.Flush();
      mysw.Flush();
      ms.Close();
      myns.Close();
      mysw.Close();
   }
   catch { }
```

zayedalbadanee@yahoo.com

الفصل الخامس

تصميم النظام (ج)

عملية عرض الدرس من قبل المعلم :

عملية عرض الدرس باستخدام إرسال صورة عبر بروتوكول الـ TCP.

عملية عرض الدرس باستخدام إرسال صورة عبر بروتوكول الـ UDP.

عملية عرض الدرس باستخدام قنوات الـ TCP.

عملية استخدام جهاز الطالب .

عملية تجميد وفك تجميد الماوس ولوحة المفاتيح.

عملية إيقاف وإعادة تشغيل جهاز الكمبيوتر.

zayedalbadanee@yahoo.com

في هذا الفصل سوف يتم شرح عملية عرض الدرس من قبل المعلم وكيفية استقبال هذا الدرس من قبل جميع أجهزة الطلاب وأيضا سوف يتم شرح آلية استخدام المعلم لجهاز الطالب بفرض تعديل الأخطاء أو غيرها

عملية عرض الدرس من قبل المعلم

يمكن أن تتم هذه العملية بطرق مختلفة سوف يتم مناقشتها في هذا الفصل وتوضيح مميزات وعيوب كل طريقة من هذه الطرق .

أولا عرض الدرس باستخدام إرسال صورة عبر بروتوكول TCP :

حيث يتم في هذه الطريقة إرسال الصورة تبعاً للطريقة التي تم شرحها في الفصل السابق (عملية المراقبة)

حيث يقوم جهاز المعلم باختيار جميع الأجهزة المتصلة بالشبكة ويقوم بإرسال صورة سطح الكتب لكل جهاز على التوالي إلى حين الانتهاء من الإرسال إلى أخر جهاز ثم يقوم بالتقاط صورة أخرى ويقوم بإرسالها إلى الأجهزة المتصلة على التوالي وهكذا...

تمتاز هذه الطريقة بوضوح الصورة المستلمة من جهاز المعلم ولكنها تعتبر في نفس الوقت طريقة غير فعالة نظراً لما تحتاجه من وقت في عملية الإرسال إلى كل جهاز على التوالي ، فلو كان هناك أربعين جهاز متصل بالشبكة سوف يقوم جهاز المعلم بإرسال الصورة أربعين مرة قبل العودة لالتقاط الصورة التالية وأيضا سوف يحدث بطء في جهاز المعلم نظراً للعمل الكبير الذي يقوم به –إرسال الصورة أربعين مرة– هذا الجهاز لذا لابد أن يتم البحث عن طريقة أكثر فاعلية من هذه الطريقة لتوفير الجهد والوقت ومن المناسب في هذه الحالة استخدام طريقة إرسال صورة عبر بروتوكول UDP .

ثانياً عملية عرض الدرس باستخدام إرسال صورة عبر بروتوكول UDP :

حيث يقوم هذا البروتوكول بإرسال الصورة مرة واحدة فقط عبر مجموعة بحيث

يستطيع كل جهاز منظم إلى هذه المجموعة استقبال هذه الصورة بنفس اللحظة .

- **عملية إرسال الصورة :**

الكود البرمجي:

```
byte[] arrImage
Bitmap bt = new Bitmap(CaptureScreen.CaptureScreen.GetDesktopImage());
pictureBox1.Image = ResizeBitmap(bt, 500, 400);
MemoryStream ms = new MemoryStream();
pictureBox1.Image.Save(ms, System.Drawing.Imaging.ImageFormat.Jpeg);
arrImage = ms.GetBuffer();
ms.Close();
Socket server = new Socket(AddressFamily.InterNetwork,
SocketType.Dgram, ProtocolType.Udp);
IPEndPoint iep = new IPEndPoint(IPAddress.Parse("225.100.0.1"), 5020);
server.SendTo(arrImage, iep);
Array.Clear(arrImage, 0, arrImage.Length);
ms.Flush();
ms.Close();
server.Close();
```

شرح الكود البرمجي:

يتم اولاً اخذ صورة من سطح المكتب وخزنها داخل متغير bt من نوع Bitmap ثم يتم ضغط هذه الصورة للتقليل من حجمها بتقليل دقتها ثم يتم بعد ذلك خزن الصورة داخل ذاكرة مؤقتة في الRAM بامتداد معين ،

zayedalbadanee@yahoo.com

ثم يتم أيضا تحويل محتوى الذاكرة المؤقتة إلى مصفوفة من نوع Byte ثم يتم تعريف متغير من نوع soket وبعد ذلك يتم تعريف النقطة التي سوف ترسل إليها الصورة بأنها أي مجموعة مثلاً المجموعة(225,100,01)

ثم يتم إرسال مصفوفة الصورة عبر المجموعة المحددة .

- **عملية استقبال الصورة :**

لاستقبال الصور ة المرسلة من جهاز المعلم يجب أولا الانضمام إلى المجموعة المرسل إليها هذه الصورة .

الكود البرمجي:

```
UdpClient sock = new UdpClient(5020);
sock.JoinMulticastGroup(IPAddress.Parse("225.100.0.1"));
IPEndPoint iep = new IPEndPoint(IPAddress.Any, 0);
data = sock.Receive(ref iep);
MemoryStream ms = new MemoryStream(data);
pictureBox1.Image = Image.FromStream(ms);
sock.Close();
Array.Clear(data, 0, data.Length);
ms.Flush();
ms.Close();
```

يتم اولاً تعريف متغير من نوع Udpclient عبر بروتوكول محدد ثم يتم الانضمام إلى المجموعة المرسل إليها الصورة ،يتم بعد ذلك تعريف الطرف المرسل بأنة أي جهاز ، ثم يتم استقبال البيانات وخزنها داخل مصفوفة من نوع byte ثم يتم خزن هذه المصفوفة في ذاكرة مؤقتة داخل الRAM ثم يتم عرض محتوى هذه الذاكرة في كائن الصورة

تمتاز هذه الطريقة بتوفيرها للجهد والوقت حيث أنه لا يتم إرسال الصورة مرات متعددة.

ولكن من عيوب هذه الطريقة انه لا يمكن إرسال الصورة ذات الحجم الكبير حيث أنه تم استخدام دالة تقوم بضغط هذه الصورة لذلك لا تكون الصورة المستلمة من قبل جهاز الطلاب بالوضوح والدقة المطلوبين لذلك لابد من إيجاد عملية تجمع بين مميزات الطريقتين السابقتين .

عرض الدرس باستخدام قنوات الـTCP :

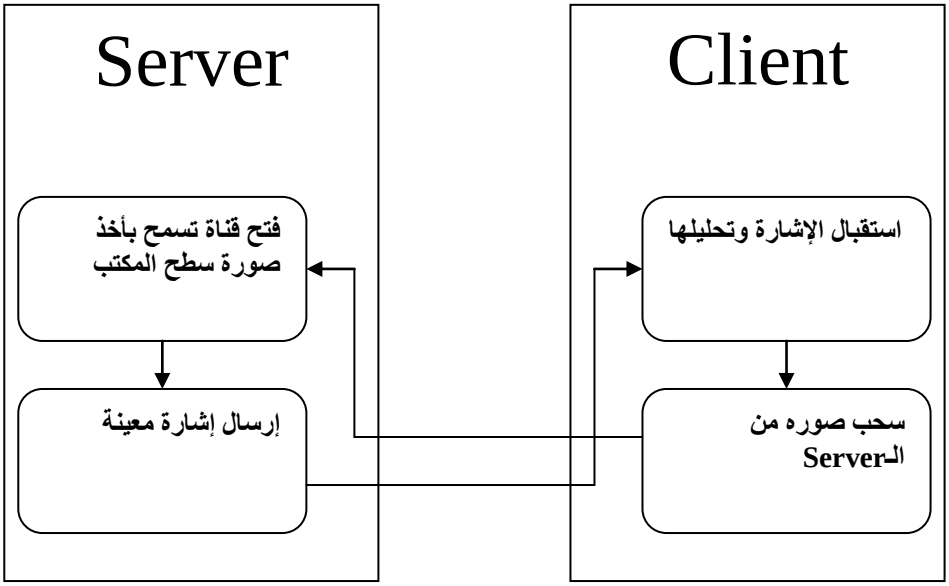
من خلال الطرق السابقة تم ملاحظة أن طريقة إرسال الصورة من جهاز المعلم غير مجدية ،حيث أن العمل في الطرق السابقة سيتركز على جهاز واحد (جهاز المعلم) بينما تظل الأجهزة الأخرى بانتظار وصول الصورة لها وسوف يتم في هذه الطريقة بعكس هذه الآلية–تركيز الجهد على جهاز المعلم– حيث سوف يقوم كل جهاز بإحضار صورة سطح المكتب من جهاز المعلم وبالتالي يتوزع العمل على جميع الأجهزة ولتحقيق هذه الفكرة لابد من استخدام القنوات فكرة هذه الطريقة (طريقة قنوات الـــTCP) بان جهاز المعلم سوف يفتح قناة اتصال تسمح لبقية الأجهزة بأخذ صورة سطح المكتب ، وبالمقابل يقوم جهاز الطالب بإحضار الصورة (صورة سطح الكتب) من جهاز المعلم .

اولاً جهاز المعلم :

الكود البرمجي :

```
chan = new TcpChannel(2500);
ChannelServices.RegisterChannel(chan, false);

RemotingConfiguration.RegisterWellKnownServiceType(Type.GetType("ScreenCaptur
e, ScreenCapture"), "MyCaptureScreenServer", WellKnownObjectMode.Singleton);
```

حيث يقوم الكود السابق اولاً بتعريف متغير من نوع TcpChannel ثم يتم تسجيل هذه القناة لتقوم بأداء عملها.



الشكل (1-5) عملية عرض الدرس على الطلاب

ثانياً جهاز الطلاب :

الكود البرمجي:

```
URI = "Tcp://"+this.Tag+":6600/MyCaptureScreenServer";
 chan1 = new TcpChannel();
 ChannelServices.RegisterChannel(Program.c2,false);
      obj = (              ScreenCapture)Activator.GetObject(typeof(ScreenCapture), URI);
```

يقوم الكود السابق اولاً بتعريف متغير من نوع TcpChannel وبعد ذلك يتم تسجيل هذه القناة وبعد ها يتم تعريف متغير من نوع `ScreenCapture obj;`

أما الكود البرمجي التالي فيعمل على:

```
byte[] buffer = obj.GetDesktopBitmapBytes();
MemoryStream ms = new MemoryStream(buffer);
pictureBox1.Image = Image.FromStream(ms);
```

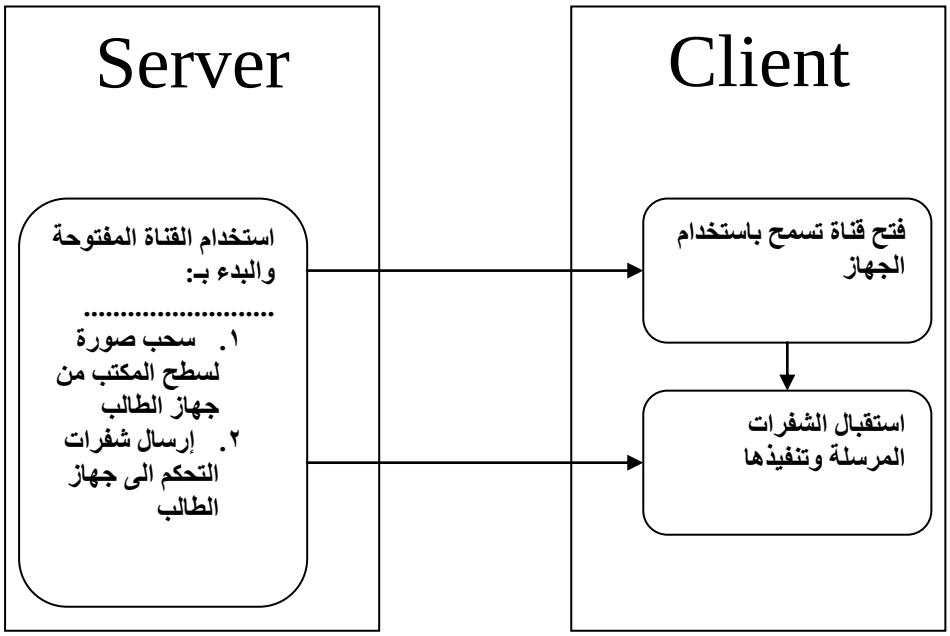
اخذ صورة لسطح المكتب من جهاز المعلم ويقوم بخزنها في مصفوفة من نوع byte

ويتم خزن هذه المصفوفة في ذاكرة مؤقتة ثم يتم بعد ذلك عرض محتويات الذاكرة المؤقتة

بكائن صورة ونلاحظ أن هذه الطريقة أفضل بكثير من الطرق السابقة وذلك لتوفيرها للوقت والجهد وأيضا دقة الصورة.

عملية استخدام الجهاز (جهاز الطالب)

إن عملية استخدام الجهاز تتم وفق مبدأ القنوات حيث يتم في هذه العملية فتح قناة بين جهاز المعلم وجهاز الطالب حيث تمكن هذه القناة جهاز المعلم من القيام بإرسال عدة أوامر يقوم جهاز الطالب بتنفيذها .



الشكل(5-3) إستخدام جهاز الطالب من قبل المعلم

الكود البرمجي:

```
public control()
    {
       InitializeComponent();
    }
    [DllImport("user32.dll")]
    private static extern uint MapVirtualKey(
       uint uCode,     // virtual-key code or scan code
       uint uMapType   // translation to perform
       );
```

```
string URI;
ScreenCapture obj;
TcpChannel chan1;
bool connected;
void start()
{
   try
   {
      URI = "Tcp://"+this.Tag+":6600/MyCaptureScreenServer";
      chan1 = new TcpChannel();
      ChannelServices.RegisterChannel(chan1,false);
      obj = (ScreenCapture)Activator.GetObject(typeof(ScreenCapture), URI);

      connected = true;
      timer1.Enabled = true;
   }
   catch (Exception) { stop(); };
}
void stop()
{
   try
   {
      timer1.Enabled = false;
      connected = false;

      this.FormBorderStyle = FormBorderStyle.Sizable; // Normal Size Mode
      this.WindowState = FormWindowState.Normal;
      ChannelServices.UnregisterChannel(chan1);//to Un Register chan Channel

   }
   catch (Exception) { }
}
private void control_Load(object sender, EventArgs e)
{
   start();
}

private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
   if (connected == true)
```

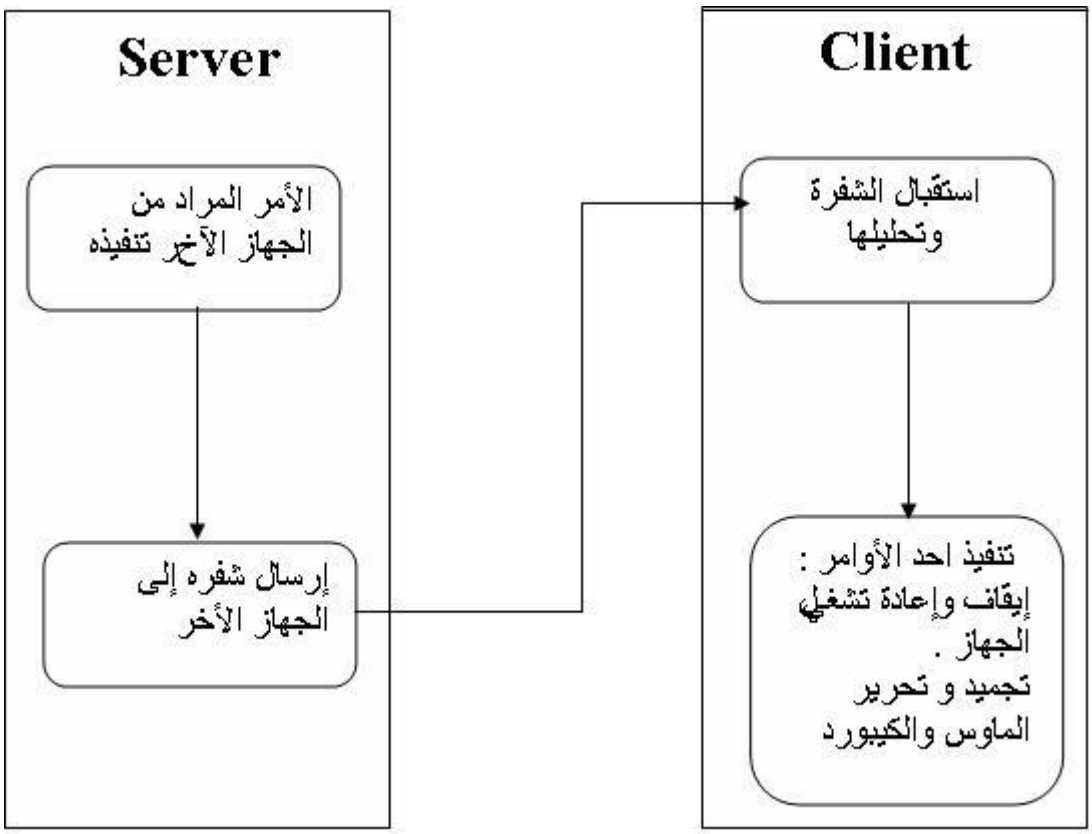
```
    {
        obj.PressOrReleaseMouseButton(true, e.Button == MouseButtons.Left, e.X, e.Y);
    }
}

private void pictureBox1_MouseMove(object sender, MouseEventArgs e)
{
        if (connected == true)
        {
            obj.MoveMouse(e.X, e.Y);
        }
}

private void pictureBox1_MouseUp(object sender, MouseEventArgs e)
{
    if (connected == true)
    {
        obj.PressOrReleaseMouseButton(false, e.Button == MouseButtons.Left, e.X, e.Y);
    }
}

private void control_KeyDown(object sender, KeyEventArgs e)
{
    if (connected == true)
    {
        e.Handled = true;
        obj.SendKeystroke((byte)e.KeyCode, (byte)MapVirtualKey((uint)e.KeyCode, 0), true, false);
    }
}

private void control_KeyUp(object sender, KeyEventArgs e)
{
    if (connected == true)
    {
        e.Handled = true;
        obj.SendKeystroke((byte)e.KeyCode, (byte)MapVirtualKey((uint)e.KeyCode, 0), false, false);
    }
}
Form1 f;
private void control_FormClosing(object sender, FormClosingEventArgs e)
```

```
{
    this.stop();
    try
    {
        if (f.kk == 1)
            chan1 = new TcpChannel(2500);
    }
    catch { };
}

private void timer1_Tick(object sender, EventArgs e)
{
    try
    {
        byte[] buffer = obj.GetDesktopBitmapBytes();
        MemoryStream ms = new MemoryStream(buffer);
        pictureBox1.Image = Image.FromStream(ms);
    }
    catch (Exception ex ) { MessageBox.Show(ex.Message); };
}

private void textBox1_KeyDown(object sender, KeyEventArgs e)
{
    if (connected == true)
    {
        e.Handled = true;
        obj.SendKeystroke((byte)e.KeyCode, (byte)MapVirtualKey((uint)e.KeyCode, 0), true, false);
    }
}

private void textBox1_KeyUp(object sender, KeyEventArgs e)
{
    if (connected == true)
    {
        e.Handled = true;
        obj.SendKeystroke((byte)e.KeyCode, (byte)MapVirtualKey((uint)e.KeyCode, 0), false, false);
    }
```

نلاحظ من خلال الكود السابق أن جهاز المعلم يقوم بأخذ صورة من سطح المكتب لجهاز الطالب تزامناً مع إرسال الأوامر من جهاز المعلم إلى جهاز الطالب عبر القنوات ،هذه الأوامر ترسل عند الأحداث التالية:

- عند تحريك مؤشر الماوس فوق كائن الصورة يقوم جهاز المعلم بإرسال إحداثيات مؤشر الماوس إلى جهاز الطالب الذي يقوم بدوره بنقل مؤشر الماوس في جهازه إلى الإحداثيات المطلوبة .
- عند ضغط زر الماوس يقوم جهاز المعلم بإرسال إحداثيات الماوس إضافة إلى نوع الضغط (يمين – يسار)فيقوم جهاز الطالب بتنفيذ هذا الأمر عند الإحداثيات المرسلة .
- عند رفع الضغط على زر الماوس يقوم جهاز المعلم بإرسال نفس الأمر مع ملاحظة أن نوع الأمر هو رفع الضغط على زر الماوس.
- عند الضغط على أي زر في لوحة المفاتيح يقوم جهاز المعلم بأخذ شفرة الايسكي لهذا الزر ثم يقوم بتحويله إلى byte من ثم إرسال هذه الشفرة إلى جهاز الطالب والذي يقوم بفهم هذا الأمر وتنفيذه .
- عند رفع الضغط على الزر المضغوط في لوحة المفاتيح يقوم جهاز المعلم بنفس الآلية السابقة ويقوم جهاز الطالب بتنفيذ الأمر المطلوب.نلاحظ أن الكود السابق هو ما يخص جهاز المعلم أما جهاز الطالب فعمله هو أن يقوم بفتح القناة ليتم التواصل بين كلٍ من جهاز المعلم وجهاز الطالب، وهكذا يتمكن جهاز المعلم من استخدام جهاز الطالب.

عملية تجميد وفك تجميد الماوس ولوحة المفاتيح:



الشكل(6-3) عمليات تحكم أخرى يمكن اجرائها في النظام

لتجميد أو لفك تجميد الماوس ولوحة المفاتيح نقوم باضافة الكود التالي

zayedalbadanee@yahoo.com

```
[System.Runtime.InteropServices.DllImport("user32", EntryPoint =
"BlockInput", ExactSpelling = true, CharSet =
System.Runtime.InteropServices.CharSet.Ansi, SetLastError = true)]
```

حيث أن هذا الكود السابق يقوم بتضمين مكتبة داخلuser32 خاصه بالماوس ولوحة المفاتيح

```
private static extern long BlockInput(bool fBlock)
```

الداله السابقه هي التي تقوم بعمليه التجميد والفك فاذا كان المدخل

`BlockInput(true);` فان الداله سوف تقوم بتجميد الماوس ولوحة المفاتيح أما

اذا كان المدخل `BlockInput(false);` فان الداله سوف تقوم بالغاء تجميد

الماوس ولوحة المفاتيح

عملية إيقاف وإعادة تشغيل جهاز الكمبيوتر:

```
System.Diagnostics.Process.Start("shutdown", "/s /f /t 00");
```

هذا الكود يقوم بإيقاف تشغيل جهاز الكمبيوتر

ويعني الرمز **s** أيقاف تشغيل الجهاز

أما الرمز **f** أيقاف جميع التطبيقات ،

الرمز **t** يشير الى المدة التي ستنقضي قبل إيقاف التشغيل.

عملية إعادة تشغيل جهاز الكمبيوتر :

```
System.Diagnostics.Process.Start("Shutdown", "-r /f /t 00");
```

نلاحظ وجود تشابه كبير بين الكوديين السابقين حيث أن الاختلاف الوحيد بينهما هو

الرمز **r** - حيث أن هذا الرمز يعني إعادة التشغيل .

الفصل السادس

الاستنتاجات والتوصيات

مميزات النظام

الاستنتاجات

التوصيات والمقترحات

رسالة شكر

المراجع

zayedalbadanee@yahoo.com

مميزات النظام

١. النظام موجه للسوق العربية حيث تنعدم إمكانية استخدام اللغة العربية في هذه البرمجيات بشكل كلي.

٢. استخدام مصطلحات عربية مفهومة وسلسة للمستخدم حتى مع انعدام الخبرة الكافية للتعامل مع الحاسوب.

الاستنتاجات

ومن خلال عمل المشروع تحققت أهداف المشروع وهي كالتالي:

١. تلبية حاجة المستخدم بأقلّ جهد ممكن .

٢. بناء النظام على خطوات سلسلة قابلة للتعامل مع المستخدم بكل سهولة للوصول إلى ما يُريده (User friendly).

٣. النظام قابل للتعامل مع أي شخص حتى ولو لم يكن ذا تخصص في مجال الكمبيوتر والبرمجة والتصميم .

التوصيات والمقترحات:

١. إدخال نظام التعليم عبر الشبكات إلى معامل الحاسوب للاستفادة من خدمات هذه التقنية.

٢. تضمين قاعدة بيانات للنظام تقوم بخزن بيانات عن الطلاب الذين يستخدمون النظام بحيث تمكن المعلم من قياس مستوى الطالب من خلال هذه البيانات .

٣. إجراء دراسة مماثلة لهذه الدراسة تشمل خدمات أكثر يمكن أن يحققها النظام .

شكر وعرفان

من باب قوله عليه الصلاة والسلام (من لا يشكر الناس لا يشكر الله) ، يسعدنا أن

نشكر من بعد الله وتوفيقه علينا

كلّ من ساهم في مساعدتنا للارتقاء بهذا المشروع

والوصول لهذه النتيجة التي وصلنا إليها الآن،

ونخص بالذكر :

الأستاذ الدكتور / رشاد عبد الله الجوفي

عميد مركز الحاسوب بجامعة إب

الدكتور / اشرف السيسي

الأستاذ / بليغ الهلالي

الأستاذ / أكرم الحداد

الأستاذ / حمزة الصيادي

كما ونشكر كُلّ أستاذ قدم يد العون لنا خلال العمل في هذا المشروع.

zayedalbadanee@yahoo.com

المراجع

(1) د. فادي عبد القادر ."احترف برمجة الشبكات والنظم الموزعة"، (الناشر SocketCoder.Com) ،الإصدار 2006) .

(2) جوان وودكوك ،"خطوة متقدمة في الشبكات"،(الناشر الدار العربية للعلوم–بيروت–لبنان)،الطبعة الأولى 1999م.

(3) سبورتاك ووالترغلين ،"أساسيات شبكات الاتصال" ، (الناشر الدار العربية للعلوم–بيروت– لبنان)،الطبعة الأولى 1998م.

(4) هيثم الهجاء،إياد الشوابكة،محمد بصبوص،"مقدمة في شبكات الحاسوب"،(دار البركة للنشر والتوزيع–الأردن–عمان)الطبعة الاولى2001م .

(5) رامي قطماني،"تعلم C#"،(شعاع للنشر والعلوم–سوريا–حلب)،الطبعة الأولى 2003م

(6) صفحات الويب .

zayedalbadanee@yahoo.com

الملاحق

بناء الشبكة المحلية

ضبط إعدادات الشبكة في نظام التشغيل XP

zayedalbadanee@yahoo.com

بناء الشبكات

قبل البدء بتركيب الشبكة يجب عليك وضع خطة لعملك أي ماهي احتياجاتك من بناء الشبكة وماهي إمكانيتك من العتاد لبناء هذه الشبكة دعنا نتحدث قليلا عن هذا الموضوع لتكتمل لديك فكرة أشمل من خلال الأسئلة التالية:

- ماذا يمكننا الاستفادة من بناء شبكة منزلية؟

١. استخدام اشتراك انترنت واحد يشغل كافة الأجهزة علي الشبكة .

٢. حفظ الملفات وتبادلها علي أي جهاز موجود علي الشبكة .

٣. المشاركة بالطابعات الموجودة .

٤. اللعب بالألعاب الجماعية علي الشبكة .

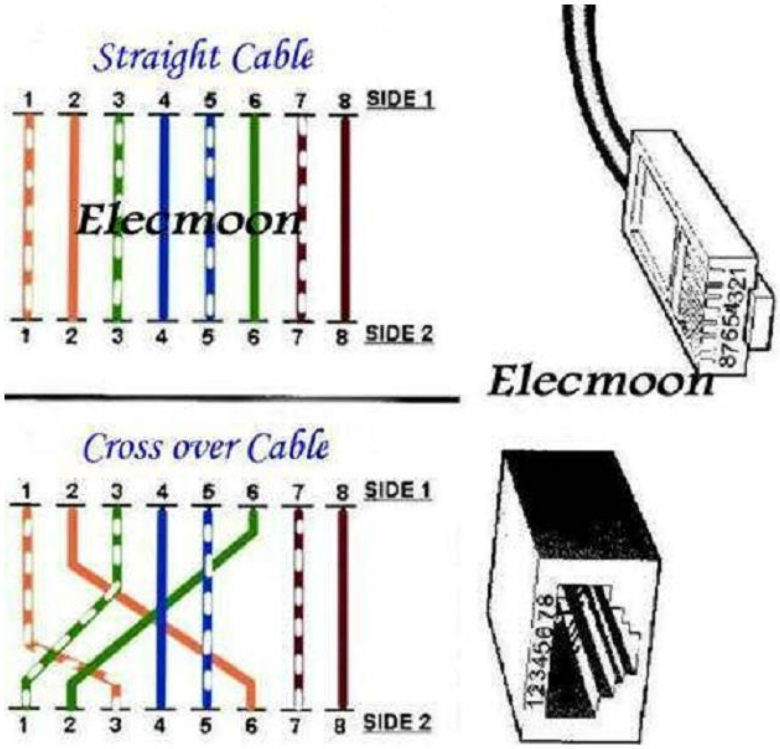
- ماذا نحتاج لبناء شبكة سواء كانت شبكة منزلية أو شبكة مؤسسة صغيرة وحتى كبيرة الحجم؟

١. تجهيز قائمة بالأشياء أو القطع المطلوبة لإنشاء الشبكة:

- NETWORK MEDIA وسائط نقل : وهي عبارة عن كوابل شبكة .
- NETWORK HUB وهو اختياري : هو عبارة عن جهاز يعمل علي توصيل كافة لأجهزة الكمبيوتر مع بعضها البعض من خلال أنشاء مكان مركزي تتجمع فيه كافة الكوابل المربوطة بالأجهزة , ويستخدم عادة في حين وجود أكثر من جهازين كمبيوتر.

٢. تجهيز كوابل الربط :

أذا كنت تريد إنشاء شبكة لجهازين فقط أوصلهم ببعض بكابل من نوع Cross over بدون استخدام هب أو سؤيتش ولكن أذا كان العدد أكثر من أثنين عليك باستخدام هب أو سويتش بأن تقوم بتوصيل الأجهزة بكوابل straight بالهب أو السويتش وهذه الصورة توضح لك كيفية ترتيب الألوان في كلا النوعين من التوصيل

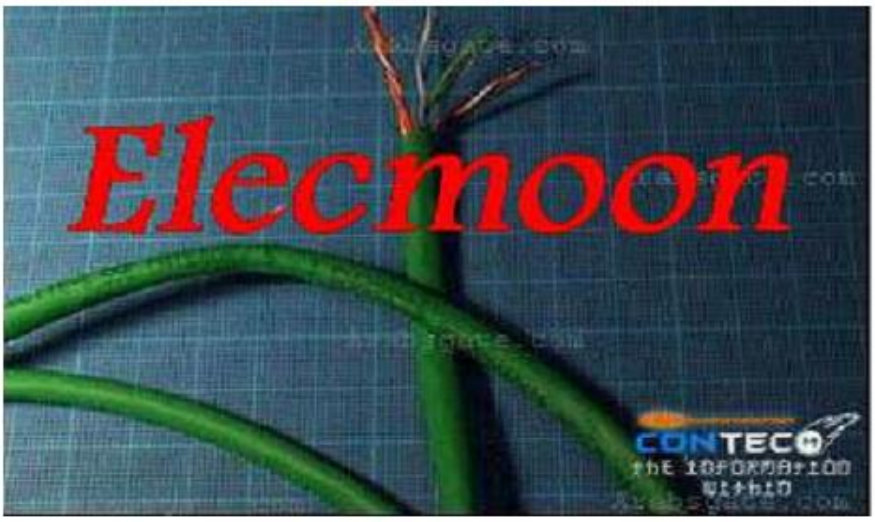


في طريقة straight نقوم بتوصيل الأطراف إلي الهب وتقوم الهب بالداخل بالتوصيل بينهم , cross

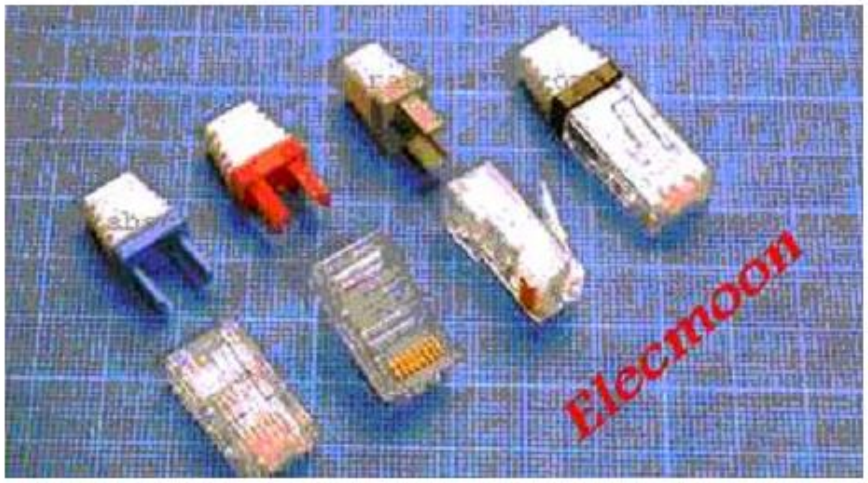
وذلك لتوصيل أطراف الإرسال في كل جهة بأطراف الاستقبال في الجهة الأخرى

والان نقوم بتحضير الكيبل هي التالي :

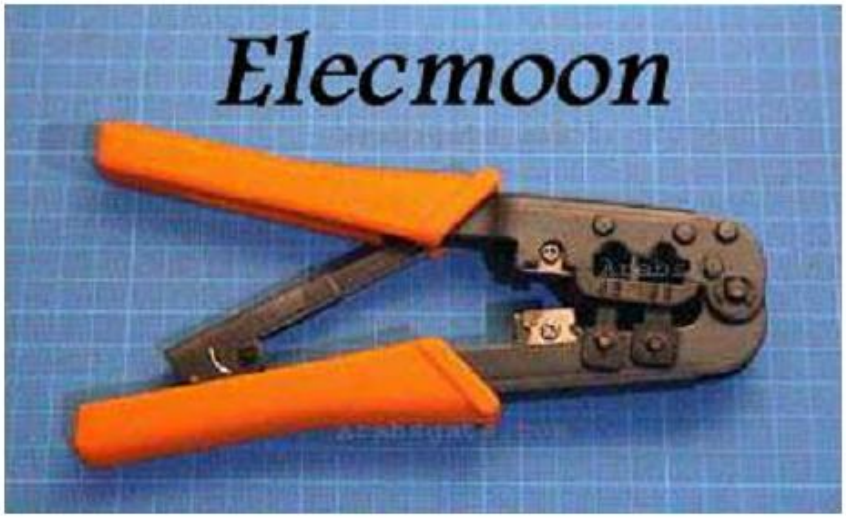
١. الكيبل نفسه بعد أن تكون قد قست الطول اللازم بين الجهازين لعمل شبكة بينهم أو بين الجهاز والهب.



٢. رأسية الكيبل وتسمي (RJ-45) :



٣. المكبس الخاص لذلك ويسمي (UTP crimping tool) :



نقوم ألان بترتيب الأسلاك بأحد الطريقتين straight أوcross over وبإمكانك الآن أن تقوم بكبس رؤوس الكيبل والصور التالية تساعدك علي ذالك:-

zayedalbadanee@yahoo.com

zayedalbadanee@yahoo.com

zayedalbadanee@yahoo.com

إعداد الشبكة

وأول خطوة هي إعداد الايبيهان(IP's number) .

الذهاب إلى لوحة التحكم (control panel) ومنها إلى اتصالات شبكة الاتصال (Network connectoin) .

وهكذا تم ضبط إعدادات الشبكة وأصبحت جاهزة للعمل .

zayedalbadanee@yahoo.com