

โปรแกรม Basic Network Admin

Basic Network Admin

นายคทาวัช อ้นตระกูล รหัส 45380191

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ..... 1 / ส.ค. 2553 /

เลขทะเบียน..... 15006700	ป.ศ.
เลขเรียกหนังสือ.....	ค.ม.ร.ป.
มหาวิทยาลัยนเรศวร	2549

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

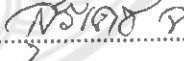
ปีการศึกษา 2549




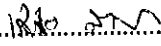
ใบรับรองโครงการวิศวกรรม

หัวข้อโครงการ	โปรแกรม Basic Network Admin		
ผู้ดำเนินโครงการ	นายคทาวัช	ยันตระกุล	รหัส 45380191
อาจารย์ที่ปรึกษา	ดร.สุรเดช จิตประไพศาลกุล		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2550		

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ อนุมัติให้โครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะกรรมการสอบโครงการวิศวกรรม


.....ประธานกรรมการ
(ดร.สุรเดช จิตประไพศาลกุล)


.....กรรมการ
(อาจารย์ จิราพร พุกสุข)


.....กรรมการ
(อาจารย์ แสงชัย มังกรทอง)

หัวข้อโครงการ	โปรแกรม Basic Network Admin
ผู้ดำเนินโครงการ	นายททาวุธ อันตระกูล รหัส 45380191
อาจารย์ที่ปรึกษา	ดร.สุรเดช จิตประไพศาลกุล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2549

บทคัดย่อ

ในปัจจุบันระบบเน็ตเวิร์กภายในองค์กรได้รับความนิยมอย่างแพร่หลาย ผู้ใช้งานคอมพิวเตอร์ต่างก็มีทางเลือกการใช้งานที่หลากหลาย อาจมีผู้ใช้ส่วนหนึ่งที่ใช้งานผิดเป้าหมาย ซึ่งก่อให้เกิดปัญหาตามมามากมาย จึงเป็นปัญหาอย่างยิ่งของผู้ดูแลระบบในการดูแลความเป็นระเบียบของการใช้งานระบบในการควบคุมความเป็นระเบียบของเครือข่ายนั้นๆ ผู้ดูแลระบบต้องรู้ว่าผู้ใช้งานคอมพิวเตอร์อยู่นั้นมีระเบียบการใช้งานขององค์กรหรือไม่ หรือใช้คอมพิวเตอร์ทำอะไรที่มีความเสี่ยงต่อระบบ หรือก่อให้เกิดปัญหาต่อระบบหรือไม่แล้วจึงตัดสินใจดำเนินการต่อไปว่า จะแจ้งเตือนหรือระงับการให้บริการผมจึงพัฒนาโครงการชิ้นนี้เพื่อใช้เป็นเครื่องมือช่วยเหลือของผู้ดูแลระบบเพื่อให้สามารถดูแลระบบเน็ตเวิร์กได้สะดวกมากยิ่งขึ้นพร้อมทั้งเพิ่มเครื่องมืออื่นๆ สำหรับเพิ่มความสะดวกในการใช้งานทำให้การดูแลระบบง่ายขึ้นและมีความสามารถที่จะพัฒนาต่อได้โดยง่าย

Project	Basic Network Admin
Name	Mr.Katawut Huntrakul ID. 45380191
Project Advisor	Dr.Suradet
Major	Computer Engineering
Department	Electrical and Computer Engineering
Academic Year	2006

ABSTRACT

Nowadays intranet system has become more and more popular in many companies. People who use it also have many choices to choose from. As a result, there are some groups of people who might use it for a wrong purpose. This has become a problem for the caretakers because it is their direct duty to see if the system is being used as it is supposed to or not. Moreover, network caretakers are also responsible for abuse of the system which may put a network at risk. So, if this kind of problem happens, it is their responsibility to decide whether to allow the person to continue a use of network or to stop a service. Therefore, I have developed this project to be as a tool to help the network caretakers take care of the system more conveniently. This software is also equipped with other tools to make a use of it easier and can be further developed as well.

กิตติกรรมประกาศ

การจัดทำโครงการ Basic Network Admin สำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับการสนับสนุนจากทางภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ ทางผู้จัดทำขอขอบพระคุณอาจารย์ที่ปรึกษา คร.สุรเดช จิตประไพศาลกุล ที่กรุณาตลอดเวลา ความคิด ประสบการณ์และคำปรึกษาอันมีค่า ทำให้คณะผู้จัดทำได้รับประสบการณ์ การทำงานอันมีค่ายิ่ง

ขอกราบขอบพระคุณบิดา มารดา ญาติพี่น้อง ที่ช่วยดูแล เป็นกำลังใจ และให้ความรัก ความอบอุ่น ตลอดเวลา



นายกทาวุธ สันตระกูล

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของโครงการ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ผลที่คาดว่าจะได้รับ	2
1.6 งบประมาณที่ใช้	2

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 องค์ประกอบพื้นฐานของเครือข่าย	3
2.2 โปรโตคอล	4
2.3 สถาปัตยกรรมเครือข่าย	5
2.4 เทคโนโลยีระบบเครือข่าย LAN	9
2.5 โทโปโลยี	10
2.6 อีเทอร์เน็ต	14
2.7 เทคโนโลยี WAN	18
2.8 อินเทอร์เน็ตเลเยอร์	20
2.9 Routing Protocol	24
2.10 Domain Name System (DNS)	25
2.11 การออกแบบระบบเครือข่าย	30
2.12 การเขียนโปรแกรมเน็ตเวิร์กโดยใช้ วินซ็อก (Winsock)	34
2.13 วินซ็อก (Winsock)	36

สารบัญ(ต่อ)

	หน้า
บทที่ 3 วิธีการดำเนินงานโครงการวิศวกรรม	
3.1 การออกแบบซอฟต์แวร์	41
3.2 สถานการณ์ทำงานของโปรแกรม	42
3.3 การควบคุมเครื่องลูกข่ายผ่านกราฟิกโหมด	45
3.4 โปรแกรมโอนย้ายข้อมูล	47
บทที่ 4 การทดสอบและวิเคราะห์การทำงาน	
4.1 การทดสอบการทำงาน	49
บทที่ 5 บทวิจารณ์และสรุป	
5.1 วิจารณ์โครงการ	56
5.2 ปัญหาทั่วไปที่เป็นอุปสรรคต่อการพัฒนาโปรแกรม	56
5.3 แนวทางการแก้ไข และการพัฒนาต่อ	56
5.4 สรุปผลโครงการ	57
เอกสารอ้างอิง	58
ภาคผนวก	59
ประวัติผู้เขียนโครงการ	92

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันระบบเครือข่ายคอมพิวเตอร์ภายในองค์กรได้รับความนิยมอย่างแพร่หลายเนื่องจากทำให้การสื่อสารสะดวกรวดเร็วมากขึ้น ระบบเหล่านี้เอื้อการทำงานอย่างมาก แต่ก็มีผู้ใช้ส่วนหนึ่งที่ใช้งานผิดเป้าหมายซึ่งก่อให้เกิดปัญหาตามมามากมาย จึงเป็นปัญหาอย่างยิ่งของผู้ดูแลระบบในการดูแลความเป็นระเบียบของการใช้งานระบบ ในการควบคุมความเป็นระเบียบของเครือข่าย ผู้ดูแลระบบต้องรู้ว่าผู้ใช้งานคอมพิวเตอร์อยู่นั้นมีระเบียบการใช้งานขององค์กรหรือไม่หรือใช้คอมพิวเตอร์ทำอะไรที่มีความเสี่ยงต่อระบบ หรือก่อให้เกิดปัญหาต่อระบบหรือไม่แล้วจึงตัดสินใจดำเนินการต่อไปว่า จะแจ้งเตือนหรือระงับการให้บริการ ทางผู้จัดทำโครงการจึงพัฒนาโครงการชิ้นนี้ เพื่อใช้เป็นเครื่องมือช่วยเหลือของผู้ดูแลระบบให้สามารถดูแลระบบเครือข่ายคอมพิวเตอร์ได้สะดวกมากยิ่งขึ้น พร้อมทั้งเพิ่มเครื่องมืออื่นๆ สำหรับเพิ่มความสะดวกในการใช้งานทำให้การดูแลระบบง่ายขึ้นและมีความสามารถที่จะพัฒนาต่อได้โดยง่าย

1.2 วัตถุประสงค์

1. เพื่อให้ผู้ดูแลระบบใช้เป็นเครื่องมือในการติดตามพฤติกรรมการใช้งานคอมพิวเตอร์ลูกข่าย
2. เพื่อเพิ่มความสะดวกแก่ผู้ดูแลระบบในการจัดการปัญหาของเครื่องลูกข่าย
3. เพื่อลดปัญหาการดูแลเครือข่ายไม่ทั่วถึงของผู้ดูแลระบบ
4. เพื่อเพิ่มความรู้ทางระบบเน็ตเวิร์คให้ผู้ที่อยากศึกษาคือในด้านนี้

1.3 ขอบเขตของโครงการ

1. พัฒนาโปรแกรมด้วย Microsoft Visual BASIC 6.0 บนระบบปฏิบัติการ Windows XP
2. ทำการจับหน้าจอของเครื่องลูกข่ายที่ลงโปรแกรมฝังลูกข่ายไว้ได้
3. ทำการควบคุมเครื่องลูกข่ายที่อยู่ในการควบคุมได้
4. ทำการรับส่งไฟล์ระหว่างเครื่องของผู้ดูแลระบบกับเครื่องลูกข่ายได้

1.4 ขั้นตอนการดำเนินงาน

กิจกรรม	พ. ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.
	49	49	50	50	50	50	50	50	50	50	50
1.จัดเก็บรวบรวมข้อมูล	←→										
2.จัดหาเอกสารและหนังสือ		←→									
3.ศึกษาและออกแบบ			←→								
4. เขียน โปรแกรม				←→							
5. ทดสอบ โปรแกรม						←→					
6.ปรับปรุงแก้ไขให้สมบูรณ์							←→				
7. จัดทำรายงานเป็นรูปเล่ม									←→		

1.5 ผลที่คาดว่าจะได้รับ

1. สามารถนำความรู้ที่ได้จากการเขียน โปรแกรมมาพัฒนาต่อในระบบของ network ได้
2. สามารถนำโปรแกรมที่ได้ทำขึ้นมาใช้งานได้จริง

1.6 งบประมาณที่ใช้

ค่าหนังสือหรือเอกสารประกอบการทำโครงการ

1,000 บาท

บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

เครือข่ายคอมพิวเตอร์ (Computer Network) คือระบบที่มีคอมพิวเตอร์อย่างน้อยสองเครื่อง เชื่อมต่อกันโดยใช้สื่อกลาง และก็สามารถสื่อสารข้อมูลกันได้อย่างมีประสิทธิภาพ ซึ่งทำให้ผู้ใช้คอมพิวเตอร์แต่ละเครื่องสามารถแลกเปลี่ยนข้อมูลซึ่งกันและกันได้ นอกจากนี้ยังสามารถใช้ทรัพยากร (Resources) ที่มีอยู่ในเครือข่ายร่วมกันได้ เช่น เครื่องพิมพ์ ซีดีรอม สแกนเนอร์ ฮาร์ดดิสก์ เป็นต้น

แนวคิดในการสร้างเครือข่ายคอมพิวเตอร์นั้น เริ่มมาจากการที่ผู้ใช้ต้องการที่จะแลกเปลี่ยนข้อมูลกันอย่างมีประสิทธิภาพและรวดเร็ว คอมพิวเตอร์เดี่ยวๆ เป็นอุปกรณ์ที่มีความสามารถในการประมวลผลข้อมูลในปริมาณมากอย่างรวดเร็วอยู่แล้ว แต่ข้อเสียคือ ผู้ใช้ไม่สามารถแชร์ข้อมูลนั้นกับคนอื่นอย่างมีประสิทธิภาพได้ก่อนที่จะมีเครือข่ายคอมพิวเตอร์

2.1 องค์ประกอบพื้นฐานของเครือข่าย

การที่คอมพิวเตอร์จะเชื่อมต่อกันเป็นเครือข่ายได้ ต้องมีองค์ประกอบพื้นฐานดังต่อไปนี้

2.1.1 คอมพิวเตอร์ อย่างน้อย 2 เครื่อง

2.1.2 เน็ตเวิร์คการ์ด หรือ NIC (Network Interface Card) เป็นการ์ดที่เสียบเข้ากับช่องที่เมนบอร์ดของคอมพิวเตอร์ ซึ่งเป็นจุดเชื่อมต่อระหว่างคอมพิวเตอร์และเครือข่าย

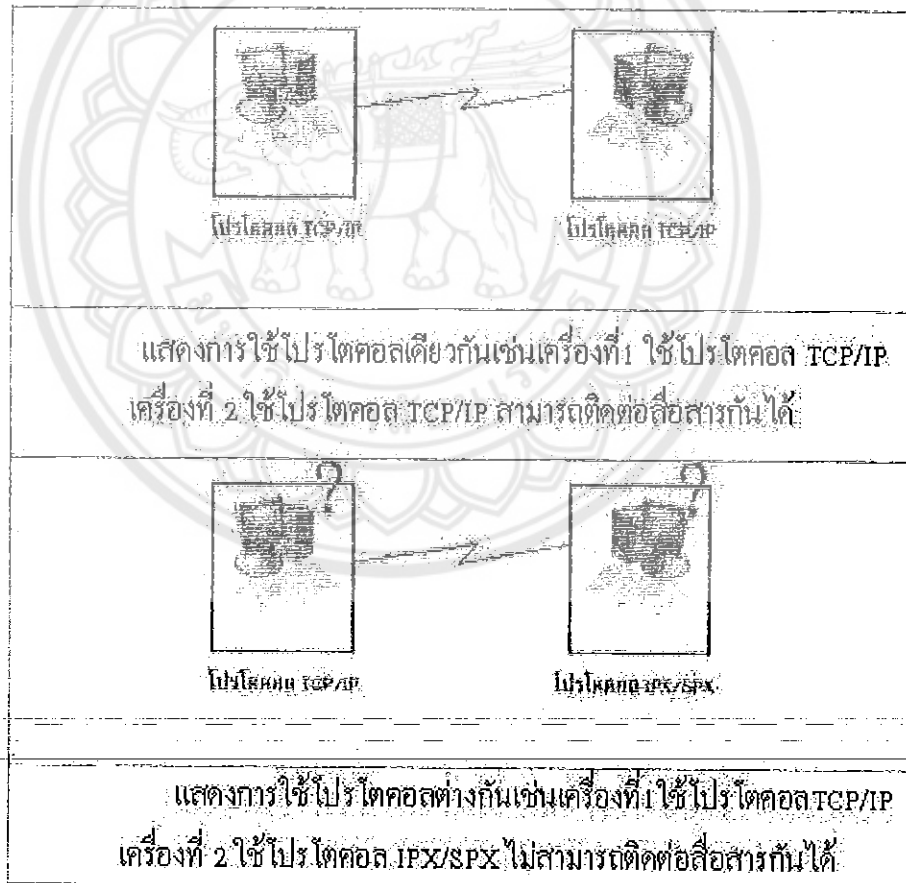
2.1.3 สื่อกลางและอุปกรณ์สำหรับการรับส่งข้อมูล เช่น สายสัญญาณ ส่วนสายสัญญาณที่นิยมที่ใช้กันในเครือข่ายก็เช่น สายโคแอกเชียล สายคู่เกลียวบิด และสายใยแก้วนำแสง เป็นต้น ส่วนอุปกรณ์ เครือข่าย เช่น ฮับ สวิตช์ เราท์เตอร์ เกตเวย์ เป็นต้น

2.1.4 โพรโทคอล (Protocol) โพรโทคอลเป็นภาษาที่คอมพิวเตอร์ใช้ติดต่อสื่อสารกันผ่านเครือข่ายคอมพิวเตอร์ที่สามารถสื่อสารกันได้นั้นจำเป็นที่ต้องใช้ “ภาษา” หรือใช้โปรโทคอลเดียวกัน เช่น OSI, TCP/IP, IPX/SPX เป็นต้น

2.1.5 ระบบปฏิบัติการเครือข่าย หรือ NOS (Network Operating System)ระบบปฏิบัติการเครือข่ายจะเป็นตัวคอยจัดการเกี่ยวกับการใช้งานเครือข่ายของผู้ใช้แต่ละคน

2.2 โพรโทคอล (Protocol)

ในการเชื่อมโยงของเครือข่ายเครื่องคอมพิวเตอร์ ในแต่ละเครื่องอาจก็ต้องมีระบบที่เหมือนกัน หรือแตกต่างกัน เช่นในการใช้งานในเครือข่ายจึงต้องเป็นมาตรฐานหรือระเบียบที่ใช้ในการติดต่อให้แต่ละเครื่องมีวิธีการสื่อสารที่เป็นไปตามแนวทางเดียวกันได้ เพื่อให้เป็นการเชื่อมโยงข้อมูล และในการติดต่อสื่อสารของเครื่องคอมพิวเตอร์ในแต่ละเครื่องต้องมีความเข้าใจถูกต้องตรงกันและสามารถทำงานร่วมกันได้เป็นอย่างดี ไม่เกิดความเสียหายนั้นเกิดขึ้น จึงมีการกำหนดวิธีการมาตรฐานขึ้นเรียกว่า โพรโทคอล ดังนั้นอาจกล่าวได้ว่าโปรโตคอล หมายถึง กฎเกณฑ์ ข้อตกลง ภาษาสื่อสาร รูปแบบ วิธีการเชื่อมต่อของเครื่องคอมพิวเตอร์ในเครือข่าย (ระบบใดๆ ก็ตาม) ให้สามารถติดต่อสื่อสาร มีการใช้งานร่วมกันได้หลากหลาย



รูปที่ 2.1 แสดงการใช้โปรโตคอล

2.3 สถาปัตยกรรมเครือข่าย

การที่มนุษย์สามารถสื่อสารกันได้อย่างมีประสิทธิภาพนั้น เนื่องจากใช้ภาษาเดียวกัน เช่น ภาษาไทย ภาษาอังกฤษ เป็นต้น ถ้าใช้คนละภาษาก็จะสื่อสารกันไม่ได้ ความคอมพิวเตอร์ก็เช่นเดียวกับมนุษย์ การที่คอมพิวเตอร์เครื่องหนึ่งจะสื่อสารกับคอมพิวเตอร์อีกเครื่องหนึ่งได้จำเป็นต้องใช้“ภาษา” เดียวกัน ภาษาที่ว่านี้ศัพท์ทางคอมพิวเตอร์เรียกว่า “โพรโทคอล (Protocol)” ดังนั้นคอมพิวเตอร์ที่สื่อสารกันต้องใช้โพรโทคอลเดียวกัน เช่น คอมพิวเตอร์ที่เชื่อมต่อเข้ากับอินเทอร์เน็ตจะใช้ “ภาษา” หรือโพรโทคอล TCP/IP ส่วนคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการเครือข่ายเน็ตแวร์ (NetWare) ก็จะใช้ “ภาษา” หรือโพรโทคอล IPX/SPX ในการสื่อสารกันเป็นต้น ในบทนี้จะอธิบายถึงหลักการการทำงานของโพรโทคอลประเภทต่าง ๆ ที่ใช้ในระบบคอมพิวเตอร์ในปัจจุบัน

ปัจจุบันฮาร์ดแวร์และซอฟต์แวร์ที่ใช้กับระบบคอมพิวเตอร์มีหลายชนิด บางชนิดก็ใช้งานร่วมกันได้แต่บางชนิดก็ใช้งานด้วยกันไม่ได้เลย ผู้ใช้บางคนอาจมีความจำเป็นต้องสื่อสารกับผู้ใช้ที่จะมีการเชื่อมต่อกับเครือข่ายอื่น เครือข่ายคอมพิวเตอร์ในปัจจุบันส่วนใหญ่จะแตกต่างกันทั้งฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ บางครั้งก็จะสื่อสารกันไม่ได้เนื่องจากเหตุผลที่ว่า คอมพิวเตอร์ที่อยู่ในเครือข่ายต่างประเภทกันจะใช้คนละ “ภาษา” ในการที่จะทำให้การติดต่อสื่อสารเป็นไปได้นั้น คอมพิวเตอร์เหล่านี้จำเป็นต้องใช้ “ภาษา” เดียวกัน ด้วยเหตุนี้จึงได้มีการพัฒนา “ภาษา” หรือโพรโทคอลขึ้นมาเพื่อให้คอมพิวเตอร์ดังกล่าวนี้สามารถสื่อสารกันได้

2.3.1 โพรโทคอลคืออะไร

การเชื่อมต่อคอมพิวเตอร์ให้เป็นเครือข่ายด้วยสายสัญญาณนั้น เป็นขั้นตอนที่ง่ายซึ่งในการสร้างเครือข่ายในแต่ส่วนที่ทำหายก็คือ การพัฒนามาตรฐานเพื่อให้คอมพิวเตอร์และอุปกรณ์เครือข่ายที่ผลิตโดยบริษัทต่างๆ ที่จะสามารถติดต่อสื่อสารกันได้ ซึ่งเป็นมาตรฐานนี้คือ โพรโทคอล (Protocol) หรือสรุปสั้นๆ โพรโทคอลคือ กฎ ขั้นตอน และรูปแบบของข้อมูลที่ใช้ในการสื่อสารระหว่างคอมพิวเตอร์สองเครื่องใดๆ ที่เชื่อมต่อกันเป็นเครือข่ายโพรโทคอลเครือข่ายบางทีก็อาจจะเรียกว่า “สถาปัตยกรรมเครือข่าย (Network Architecture)” เนื่องจากระบบเครือข่ายคอมพิวเตอร์ในปัจจุบันเป็นระบบที่ซับซ้อนมาก ทำให้ยากต่อการออกแบบโดยคนๆ เดียวหรือกลุ่มเดียว

2.3.2 แบบอ้างอิง OSI

องค์การมาตรฐานนานาชาติ หรือเรียกว่า (The International Organization for Standardization) และใช้อักษรย่อว่า “ISO” ซึ่งคนส่วนใหญ่เข้าใจว่าย่อมาจาก “International Standard Organization” แต่จริงๆ แล้วไม่ใช่ อย่างไรก็ตาม ISO เป็นองค์กรที่ออกแบบโพรโทคอล ISO (Open System

Interconnect) หรือโปรโตคอลการเชื่อมต่อเครือข่ายแบบเปิด จุดมุ่งหมายของการพัฒนามาตรฐานนี้ การจัดเรียงโปรโตคอลเป็นชั้นๆ หรือเลเยอร์นี้ก็เพื่อจำลองการไหลของข้อมูลจากเครื่องส่งถึงเครื่องรับ แต่ละชั้นจะส่งข้อมูลไปยังชั้นที่อยู่ติดกัน เช่น ถ้าเป็นการส่งข้อมูล ข้อมูลจะถูกส่งไปยังชั้นที่อยู่ต่ำกว่าถัดลงไป แต่ถ้าเป็นการรับข้อมูล ข้อมูลก็จะส่งจากข้างล่างขึ้นข้างบน แต่ละชั้นจะมีจุดเชื่อมต่อกับชั้นที่อยู่ใกล้เคียง เพื่อให้การติดต่อสื่อสารสำเร็จได้

การติดต่อสื่อสารของแต่ละชั้นจะเป็นแบบเพียร์ทูเพียร์ (Peer-to-Peer) หมายความว่าโปรโตคอลชั้นที่อยู่ฝั่งส่งจะติดต่อกับโปรโตคอลชั้นเดียวกันที่อยู่ฝั่งรับ ข้อมูลที่อยู่ชั้นนี้จะมี ความหมายเฉพาะกับโปรโตคอลที่อยู่ระดับเดียวกันของฝั่งตรงกันข้ามเท่านั้น ในปัจจุบันระบบเครือข่ายมีโปรโตคอลที่ใช้หลายประเภท ซึ่งพัฒนาโดยบางองค์กรหรือบางบริษัท โครงสร้างของโปรโตคอลเหล่านี้ก็แบ่งเป็นชั้นๆ คล้ายกับแบบอ้างอิง OSI แต่อาจจะไม่เหมือนกันทุกเลเยอร์ บางชุดโปรโตคอลอาจแบ่งชั้นตอนการรับส่งข้อมูลแค่ 4-5 ชั้นเท่านั้น แทนที่จะเป็น 7 ชั้นเหมือนแบบอ้างอิง OSI ซึ่งการทำงานแต่ละชั้นอาจไม่เหมือนของ OSI ทุกอย่าง อย่างไรก็ตามแบบอ้างอิง OSI ก็ถือได้ว่าเป็นชุดโปรโตคอลที่เป็นต้นแบบสำหรับการศึกษาโปรโตคอลชุดอื่นได้ดี

ชุดโปรโตคอล OSI ประกอบด้วยโปรโตคอลมาตรฐานหลายโปรโตคอล โปรโตคอลเหล่านี้เป็นส่วนหนึ่งของโครงการนานาชาติเพื่อพัฒนาโปรโตคอลและมาตรฐานอื่นๆ เพื่อเอื้ออำนวยให้อุปกรณ์เครือข่ายที่ผลิตจากบริษัทต่างๆ สามารถทำงานร่วมกันได้

2.3.3 OSI Protocols

OSI Reference Model	OSI Protocol
7 Application	CMIP, DS, FTAM, MHS, VTP
6 Presentation	Presentation Service/Presentation Protocol
5 Session	Session Service/ Session Protocol
4 Transport	TP0, TP1, TP2, TP3, TP4, TP5
3 Network	CONP/CMNS, CLNP/CLNS, IS-IS, ES-IS
2 Data link	IEEE 802.2, IEEE 802.3, IEEE 802.5, FDDI, X.25
1 Physical	

แสดงชุดโปรโตคอล OSI ที่ทำงานในแต่ละเลเยอร์ โดยในบทนี้จะกล่าวสรุปเกี่ยวกับโปรโตคอลเหล่านี้ ดังต่อไปนี้

1. ชั้นประยุกต์ (Application Layer)

โปรโตคอลชั้นที่อยู่ในด้านบนสุดของแบบอ้างอิง OSI ก็คือชั้นประยุกต์ (Application Layer) ถึงแม้ว่าจะเป็นแอปพลิเคชันเลเยอร์แต่ก็ไม่ได้รวมเอาแอปพลิเคชันของผู้ใช้ด้วย (User Application) แต่โปรโตคอลในชั้นนี้จะเป็นจุดเชื่อมต่อระหว่างแอปพลิเคชันของผู้ใช้กับกระบวนการสื่อสารผ่านเครือข่าย ชั้นนี้อาจถือได้ว่าเป็นชั้นที่เริ่มการติดต่อสื่อสาร—เช่น—เมื่อผู้ที่ต้องการส่งอีเมล โปรแกรมที่ผู้ใช้ใช้ส่งอีเมลจะทำการติดต่อกับโปรโตคอลในชั้นประยุกต์เพื่อเริ่มกระบวนการทั้งหมด ตัวอย่างของโปรโตคอลที่ทำงานในเลเยอร์นี้ เช่น

2. ชั้นนำเสนอ (Presentation Layer)

โปรโตคอลในชั้นนี้จะรับผิดชอบเรื่องเกี่ยวกับรูปแบบของข้อมูลที่รับส่งผ่านเครือข่าย เนื่องจากคอมพิวเตอร์ที่ต้องการแลกเปลี่ยนข้อมูลกันนั้นอาจมีวิธีการเข้ารหัส (Encoding) ที่ต่างกัน เช่น คอมพิวเตอร์บางเครื่องอาจใช้การเข้ารหัสแบบ ASCII (American Code for Information Interchange) หรือบางเครื่องอาจใช้การเข้ารหัสแบบ EBCDIC (Extended Binary Coded Decimal Interchange Code) ดังนั้นก่อนการส่งข้อมูลโปรโตคอลในเลเยอร์นี้ก็จะแปลงข้อมูลให้อยู่ในรูปแบบที่เป็นมาตรฐาน ส่วนทางด้านฝ่ายรับก็จะทำการแปลงเลขทศนิยมที่ต่างกันสามารถแลกเปลี่ยนข้อมูลกันได้

3. ชั้นเซสชัน (Session Layer)

ชั้นเซสชัน (Session Layer) ทำหน้าที่ควบคุมการสื่อสารผ่านเครือข่ายที่กำลังเกิดขึ้นทั้งสองฝั่ง การสื่อสารที่กำลังเป็นไปในช่วงขณะใดขณะหนึ่งจะเรียกว่า “เซสชัน (Session)” แอปพลิเคชันทั้งสองฝั่งสามารถแลกเปลี่ยนข้อมูลและรับส่งแฟ้มเกิดถึงกันและกันได้ในช่วงเวลาที่เซสชันยังอยู่ โดยเซสชันเลเยอร์จะรับผิดชอบเกี่ยวกับการสร้างเซสชัน ควบคุมการแลกเปลี่ยนข้อมูล และยกเลิกเซสชันเมื่อการสื่อสารสิ้นสุด

4. ชั้นเคลื่อนย้ายข้อมูล (Transport Layer)

ชั้นเคลื่อนย้ายข้อมูล หรือทรานสปอร์ตเลเยอร์ (Transport Layer) รับผิดชอบในการเคลื่อนย้ายข้อมูลระหว่างโปรเซสส์ของผู้รับและโปรเซสส์ของผู้ส่ง โดยโปรเซสส์ในที่นี้จะหมายถึงโปรแกรมที่กำลังรันบนเครื่องคอมพิวเตอร์ ในขณะที่ขณะใดขณะหนึ่งอาจจะมีหลายโปรเซสส์ที่กำลังรันอยู่—ดังนั้นชั้นนี้จะรับผิดชอบในการรับส่งข้อมูลให้ถึงโปรเซสส์ที่ต้องการ หน้าที่อีกอย่างของโปรโตคอลในชั้นนี้คือ การตรวจเช็คแพ็กเก็ตที่ละทิ้งโดยเราท์เตอร์ และทำการส่งข้อมูลใหม่อีกครั้ง โปรโตคอลในเลเยอร์นี้สามารถให้บริการได้หลายๆ แอปพลิเคชันในเวลาเดียวกัน เพื่อการกำหนดที่อยู่เพื่อใช้ติดต่อกับแอป

พลีเกชันที่อยู่อีกฝั่งหนึ่ง โดยที่อยู่ในที่นี้ส่วนใหญ่จะเรียกว่า “พอร์ต (Port)” แต่การเชื่อมต่อเข้ากับพอร์ตเรียกว่า “ซ็อกเก็ต (Socket)”

5. ชั้นเครือข่าย (Network Layer)

ชั้นเครือข่าย (Network Layer) จะรับผิดชอบในการจัดเส้นทางให้กับข้อมูลระหว่างสถานีส่งและสถานีรับ ถ้ามีเส้นทางเดียว เช่น ถ้ามีคอมพิวเตอร์แค่สองเครื่องเชื่อมต่อกันโดยตรง การจัดเส้นทางคงไม่ยากเพราะมีแค่เส้นทางเดียว แต่ถ้าเป็นเครือข่ายที่ซับซ้อนการจัดเส้นทางก็ไม่ใช่ว่าจะง่ายนัก ในชั้นนี้จะไม่มีการตรวจสอบข้อผิดพลาดของข้อมูล ดังนั้นฟังก์ชันนี้จึงเป็นหน้าที่ของชั้นเชื่อมโยงข้อมูล ชั้นเครือข่ายจะรับผิดชอบในการกำหนดเส้นทางข้อมูลระหว่างสถานีส่งและสถานีรับบนละ เครื่องข่าย การที่จะทำเช่นนี้ได้ต้องมีระบบการจัดการที่อยู่ (Addressing) ที่ไม่ขึ้นอยู่กับการที่อยู่ที่ใช้ในการเชื่อมโยงข้อมูล

6. ชั้นเชื่อมโยงข้อมูล (Data Link Layer)

เลขอร์ที่สองของแบบอ้างอิง OSI มีชื่อว่าชั้นเชื่อมโยงข้อมูล ชั้นนี้ก็ทำหน้าที่เหมือนกันชั้นอื่นๆ คือรับและส่งข้อมูล ซึ่งชั้นนี้จะรับผิดชอบในการรับส่งข้อมูลและมีการตรวจสอบความถูกต้องของข้อมูลด้วยทางด้านสถานีที่ส่งข้อมูลจะจัดข้อมูลให้เป็นเฟรม (Fram) ซึ่งในเฟรมจะมีข้อมูลที่ทำให้เฟรมสามารถส่งไปยังสถานีรับผ่านเครือข่ายท้องถิ่น (LAN) อย่างถูกต้องและสำเร็จ การส่งข้อมูลสำเร็จในที่นี้หมายถึงการที่เฟรมข้อมูลส่งถึงปลายทางที่สถานีส่งต้องการ โดยที่เฟรมข้อมูลไม่มีข้อผิดพลาด ดังนั้นในเฟรมต้องมีข้อมูลที่ใช้ในการตรวจสอบข้อผิดพลาดของเฟรมข้อมูลนั้นๆ ด้วย การส่งข้อมูลสำเร็จนั้นเหตุการณ์ต่อไปนี้จะเกิดขึ้น

- สถานีรับ เมื่อได้รับเฟรมแล้วต้องตรวจสอบข้อผิดพลาดของข้อมูลแล้วแจ้งให้สถานีส่งทราบ
- สถานีส่ง ต้องได้รับการตอบรับจากสถานีรับว่าได้รับเฟรมข้อมูลถูกต้องแล้ว

7. ชั้นกายภาพ (Physical Layer)

เลขอร์ที่อยู่ล่างสุดคือ ชั้นกายภาพ (Physical Layer) เลขอร์นี้จะรับผิดชอบเกี่ยวกับการรับส่งข้อมูลที่เป็นบิต หรือ 0 กับ 1 ในระบบเลขฐานสอง (Binary) ชั้นนี้จะรับข้อมูลจากเลขอร์ที่ 2 หรือชั้นเชื่อมโยงข้อมูล (Data Link Layer) ซึ่งข้อมูลชุดหนึ่งจะเรียกว่า “เฟรม (Fram)” และทำการส่งเฟรมของข้อมูลนี้ทีละบิตแบบเรียงตามลำดับ เหตุการณ์นี้จะเกิดขึ้นทางฝั่งสถานีที่ส่งข้อมูล ส่วนทางฝั่งสถานีรับข้อมูล ชั้นกายภาพก็จะทำการรับข้อมูลที่ส่งมาทีละบิตและจัดส่งผ่านข้อมูลเป็นบิตนี้ต่อไปยังชั้นเชื่อมโยงข้อมูลเพื่อทำการโปรเซสต่อไป

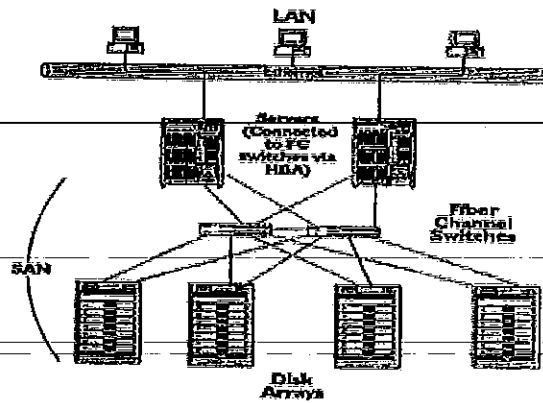
2.3.4 ชุดโปรโตคอล TCP/IP

ชุดโปรโตคอล TCP/IP (Transmission Control Protocol/Internet Protocol) ได้ถูกพัฒนามานานแล้วกว่า 20 ปีซึ่งเริ่มจากการวิจัยที่สนับสนุนโดยกระทรวงกลาโหมสหรัฐฯ จุดประสงค์ของการวิจัยนี้เพื่อเชื่อมต่อคอมพิวเตอร์ที่ต่างแพลตฟอร์มกันให้สามารถสื่อสารกันผ่านเครือข่ายได้ ซึ่งสามารถทำได้โดยการแบ่งโปรโตคอลเป็นชั้นและเป็นการแยกการทำงานของแอปพลิเคชันของผู้ใช้ ออกจากฮาร์ดแวร์ที่ใช้รับส่งข้อมูลผ่านเครือข่าย ชุดโปรโตคอลนี้จะมีการจัดรูปแบบที่แตกต่างจากแบบอ้างอิง OSI เล็กน้อย

การออกแบบชุดโปรโตคอล TCP/IP จะมุ่งเน้นไปที่การเชื่อมต่อระหว่างระบบที่ต่างกัน ในขณะที่แบบอ้างอิง OSI จะเน้นไปที่การแบ่งการทำงานของโปรโตคอลออกเป็นชั้นๆ การออกแบบ TCP/IP ยังคงเป็นแบบชั้นๆ เหมือนกัน แต่เมื่อถึงตอนทำจริงๆ ก็ให้ขึ้นอยู่กับความคิดเห็นของผู้ออกแบบ ซึ่งเป็นผลให้ชุดโปรโตคอล OSI เหมาะสำหรับการสื่อสารระหว่างคอมพิวเตอร์ในเครือข่ายได้ดีกว่า ในขณะที่ชุดโปรโตคอล TCP/IP เป็นที่นิยมมากกว่าการนำไปใช้จริง

2.4 เทคโนโลยีระบบเครือข่าย LAN

LAN (Local Area Network) คือเครือข่ายข้อมูลความเร็วสูงและทนทานต่อการเกิดข้อผิดพลาดระหว่างการรับส่งข้อมูล เครือข่าย LAN นั้นจะครอบคลุมพื้นที่ขนาดเล็ก โดยปกติจะเป็นการเชื่อมต่อคอมพิวเตอร์ เครื่องพิมพ์ และอุปกรณ์อื่นๆ ที่อยู่ไม่ห่างกันมากนัก เครือข่าย LAN ได้ให้ประโยชน์แก่ผู้ใช้หลายอย่าง เช่น การแชร์กันใช้อุปกรณ์และแอปพลิเคชัน การแลกเปลี่ยนไฟล์ระหว่างผู้ใช้ การใช้สื่อสารโดยใช้อีเมลและแอปพลิเคชันอื่นๆ LAN เป็นรากฐานของเครือข่ายคอมพิวเตอร์ทั่วไป กล่าวคือเกือบทุกๆ เครือข่ายต้องมี LAN เป็นส่วนประกอบ เครือข่ายแบบ LAN อาจเป็นได้ตั้งแต่เครือข่ายแบบง่ายๆ เช่น มีคอมพิวเตอร์สองเครื่องเชื่อมต่อกันด้วยสายสัญญาณ ไปจนถึงเครือข่ายที่สลับซับซ้อน เช่น มีคอมพิวเตอร์เป็นพันๆ เครื่องและมีอุปกรณ์เครือข่ายอื่นๆ อีกมาก แต่ลักษณะสำคัญของ LAN ก็คือ เครือข่ายประเภทนี้จะครอบคลุมพื้นที่จำกัด รูปที่ 2.26 แสดงเครือข่ายท้องถิ่นที่ประกอบด้วยเครื่องคอมพิวเตอร์ 4 เครื่อง และมีเครื่องพิมพ์ที่แชร์กันใช้ เครื่องเซิร์ฟเวอร์ที่ใช้ในการจัดการเครือข่าย ซึ่งเครือข่ายจะรวมกันอยู่ในห้องปฏิบัติการ



รูปที่ 2.2 เครื่องข่าย LAN

2.5 โทโปโลยี

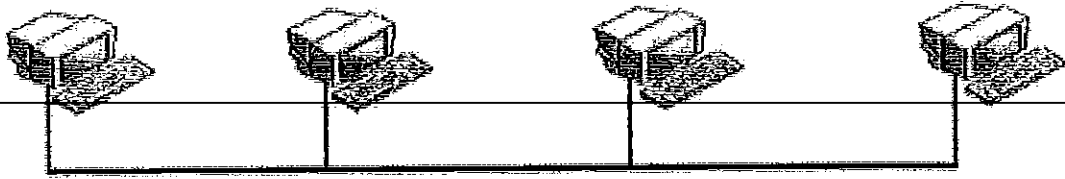
โทโปโลยีของเครือข่าย (Network Topology) จะอธิบายถึงแผนผังการเชื่อมต่อคอมพิวเตอร์ ตามลักษณะทางกายภาพ (Physical Topology) หรือทางตรรกะ (Logical Topology) ซึ่งจะแสดงถึง ตำแหน่งของคอมพิวเตอร์และอุปกรณ์เครือข่ายอื่นๆ และเส้นทางการเชื่อมต่อของอุปกรณ์เหล่านี้ โทโปโลยีของเครือข่ายอาจจะมีผลต่อสมรรถนะของเครือข่ายได้ การเลือกโทโปโลยีอาจมีผลต่อ

- ประเภทของอุปกรณ์ที่ใช้ในเครือข่าย
- สมรรถนะของอุปกรณ์เหล่านั้น
- ความสามารถในการขยายของเครือข่าย
- วิธีการดูแลและจัดการเครือข่าย

การเชื่อมต่อคอมพิวเตอร์เป็นเครือข่ายนั้น ไม่ใช่แค่การใช้สายสัญญาณเชื่อมต่อเข้ากับเน็ตเวิร์คการ์ดของแต่ละเครื่องเท่านั้น โทโปโลยีที่ใช้ต้องสัมพันธ์กับสายสัญญาณ เน็ตเวิร์คการ์ด ระบบปฏิบัติการเครือข่าย และอุปกรณ์เครือข่ายอื่นๆที่จะเชื่อมกันเป็นเครือข่าย ทุกเครือข่ายต้องประกอบด้วยโทโปโลยีใดโทโปโลยีหนึ่งต่อไปนี้

2.5.1 โทโปโลยีแบบบัส (Bus Topology) บางทีก็เรียกว่า "Linear bus" เพราะมีการเชื่อมต่อแบบเส้นตรงซึ่งเป็นลักษณะการเชื่อมต่อที่ง่ายที่สุด และเป็นโทโปโลยีที่นิยมกันมากที่สุด

แสดงการเชื่อมต่อแบบบัส ซึ่งการเชื่อมต่อแบบนี้จะใช้สายสัญญาณเพียงเส้นเดียวเชื่อมต่อคอมพิวเตอร์ทุกๆเครื่องเข้าด้วยกัน



รูปที่ 2.3 การเชื่อมต่อแบบบัส

คอมพิวเตอร์ที่เชื่อมต่อเข้ากับสายสัญญาณร่วมหรือบัส จะสื่อสารกันโดยใช้ที่อยู่ ซึ่งคอมพิวเตอร์แต่ละเครื่องจะมีที่อยู่ที่ไม่ซ้ำกัน ในการส่งสัญญาณในสายที่แชร์กันนี้จำเป็นต้องเข้าใจหลักการต่อไปนี้

ลักษณะการส่งข้อมูล การส่งข้อมูลบนเครือข่ายที่มีโทโปโลยีแบบบัสนั้นข้อมูลจะถูกส่งไปบนสายสัญญาณในรูปแบบของสัญญาณอิเล็กทรอนิกส์ ซึ่งสัญญาณนี้จะเดินทางไปถึงคอมพิวเตอร์ทุกเครื่องที่เชื่อมต่อเข้ากับสื่อกลางหรือบัส ดังนั้นเมื่อคอมพิวเตอร์เครื่องเดียวเท่านั้นที่ส่งข้อมูลได้ในเวลาใดเวลาหนึ่ง รูปที่ 2.4 แสดงการส่งข้อมูลจากเครื่อง D ไปยังเครื่อง B จะเห็นได้ว่าสัญญาณข้อมูลจะถูกส่งออกไปในรูปสัญญาณไฟฟ้าบนสายสัญญาณคอมพิวเตอร์ทุกเครื่องที่พ่วงต่อเข้ากับสายสัญญาณก็จะได้รับสัญญาณทุกเครื่อง แต่เฉพาะเครื่อง B เท่านั้นที่จะนำข้อมูลไปประมวลผลและใช้ต่อไป เนื่องจากเครื่อง B เท่านั้นที่มีที่อยู่ตรงกับที่อยู่ในข้อมูลที่ส่ง



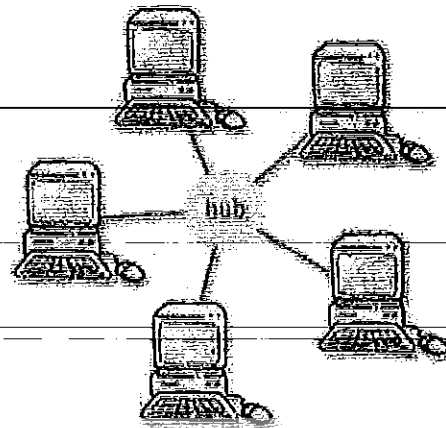
รูปที่ 2.4 แสดงการส่งข้อมูลจาก D ไป B บนบัส

เนื่องจากมีคอมพิวเตอร์เพียงเครื่องเดียวเท่านั้นที่จะสามารถส่งข้อมูลได้ในเวลาหนึ่ง ดังนั้นจำนวนคอมพิวเตอร์ที่พ่วงต่อเข้ากับสื่อกลางจะมีผลต่อประสิทธิภาพของเครือข่าย เพราะยิ่งจำนวนคอมพิวเตอร์มากเท่าไร ยิ่งทำให้คอมพิวเตอร์ต้องรอนานเพื่อที่จะส่งข้อมูล ซึ่งอาจมีผลทำให้เครือข่ายช้ามากขึ้น และยังไม่มียุติการที่เป็นมาตรฐานในการวัดว่าจำนวนคอมพิวเตอร์ที่เชื่อมต่อกันเข้ากับเครือข่ายนั้นมีผลกระทบต่อประสิทธิภาพของเครือข่ายอย่างไร ปัจจัยที่จะทำให้ประสิทธิภาพของเครือข่ายลดลงนั้นก็ไม่ใช่เฉพาะจำนวนคอมพิวเตอร์อย่างเดียว สิ่งต่อไปนี้เป็นปัจจัยอื่นๆ ที่อาจมีผลต่อประสิทธิภาพของเครือข่ายได้

- ประสิทธิภาพของฮาร์ดแวร์ของคอมพิวเตอร์ในเครือข่าย
- จำนวนของโปรแกรมที่กำลังรันบนเครื่องคอมพิวเตอร์
- ชนิดของแอปพลิเคชันที่ใช้เครือข่าย
- ประสิทธิภาพของสายสัญญาณที่ใช้
- ระยะห่างระหว่างคอมพิวเตอร์ในเครือข่าย

ในขณะที่ขณะหนึ่งคอมพิวเตอร์ที่เชื่อมต่อเข้ากับเครือข่ายก็จะถูกเช็คว่ามีข้อมูลส่งมาถึงตัวเองหรือไม่ หรือไม่ก็กำลังจะส่งข้อมูล เนื่องจากคอมพิวเตอร์แต่ละเครื่องไม่มีหน้าที่ในการส่งข้อมูลได้ ดังนั้นเมื่อคอมพิวเตอร์เครื่องใดเครื่องหนึ่งจะหยุดทำงานก็จะไม่ทำให้เครือข่ายล่มได้ ตัวเทอร์มินเตอร์ (Terminator) จะทำหน้าที่ดูดคลื่นสัญญาณเพื่อไม่ให้สะท้อนกลับ และจะถูกติดตั้งไว้ที่ปลายสายสัญญาณ การดูดคลื่นสัญญาณนี้จะทำให้สายสัญญาณว่าง และพร้อมสำหรับการส่งข้อมูลอีกที่ปลายทั้งสองข้างของสายสัญญาณ จะต้องเสียบเข้ากับสิ่งใดสิ่งหนึ่ง ตัวอย่าง เช่น เน็ตเวิร์คการ์ด หรือตัวเชื่อมต่อ ที่ใช้ในการเชื่อมต่อสายสัญญาณให้มีระยะยาวขึ้น ปลายที่ไม่ได้เสียบเข้ากับอุปกรณ์ใดๆ จะต้องติดตั้งตัวเทอร์มินเตอร์เพื่อป้องกันการสะท้อนกลับของสัญญาณ การรบกวนการสื่อสารของเครือข่าย เมื่อเกิดสายสัญญาณขาด ณ จุดใดจุดหนึ่ง หรือมีการถอดปลายสายออกจากเครื่องคอมพิวเตอร์ ซึ่งทำให้สายสัญญาณ ณ จุดนั้นไม่มีตัวเทอร์มินเตอร์ อันเป็นเหตุให้สัญญาณสะท้อนกลับ ซึ่งจะไปรบกวนสัญญาณเดิม และทำให้ข้อมูลนั้นเสียไป สัญญาณนี้ก็จะสะท้อนกลับ ไปกลับมาซึ่งทำให้ไม่สามารถส่งข้อมูลใหม่ได้ นี่เป็นประเภทหนึ่งที่ทำให้เครือข่ายนี้ล่ม ซึ่งมีผลให้เครือข่ายไม่สามารถทำงานได้

2.5.2 โทโปโลยีแบบดวงดาว (Star Topology) คอมพิวเตอร์แต่ละเครื่องจะเชื่อมต่อด้วยสายสัญญาณเข้ากับอุปกรณ์รวมศูนย์ที่เรียกว่า “ฮับ (Hub)” รูปที่ 2.5 แสดงการเชื่อมต่อเครือข่ายแบบโทโปโลยีแบบดวงดาว

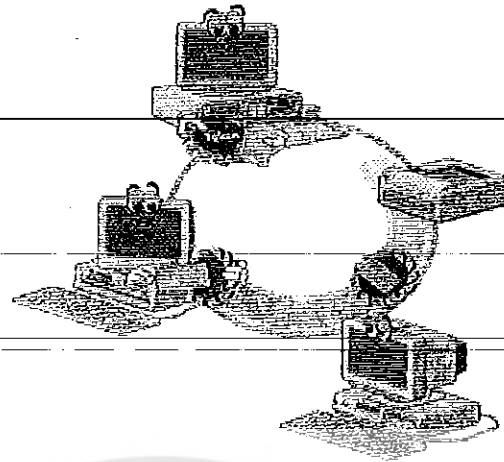


รูปที่ 2.5 โทโปโลยีแบบดวงดาว

การเชื่อมต่อในแบบนี้มีข้อดีคือ การรวมศูนย์เพื่อเป็นการบริหารทรัพยากร อย่างไรก็ตามการเชื่อมต่อแบบนี้จะสิ้นเปลืองสายสัญญาณมาก เนื่องจากเครื่องทุกเครื่องจะต้องใช้สายสัญญาณเชื่อมต่อเข้ากับฮับ และอีกอย่างหนึ่ง ถ้าหากอุปกรณ์ที่ทำหน้าที่เป็นศูนย์กลางรับส่งข้อมูลหยุดทำงาน ระบบเครือข่ายจะล่มทันที แต่อย่างน้อยก็รู้สาเหตุ ข้อดีอีกอย่างของโทโปโลยีแบบนี้คือ ถ้าสายสัญญาณขาดเฉพาะเครื่องที่ใช้สายสัญญาณนั้นเท่านั้นที่ไม่สามารถใช้เครือข่ายได้ ส่วนเครื่องอื่นๆยังใช้เครือข่ายได้เช่นเดิม เนื่องจากฮับจะทำหน้าที่เป็นตัวสิ้นสุดสัญญาณโดยอัตโนมัติเมื่อสายขาด การเชื่อมต่อแบบนี้จะเป็นที่นิยมมากในปัจจุบัน เนื่องมาจากอีเทอร์เน็ตซึ่งกลายมาเป็นมาตรฐานเครือข่ายแบบท้องถิ่น ซึ่งในปัจจุบันนั้น ก็จะใช้การเชื่อมต่อหรือโทโปโลยีแบบดวงดาว

2.5.3 โทโปโลยีแบบวงแหวน (Ring Topology)

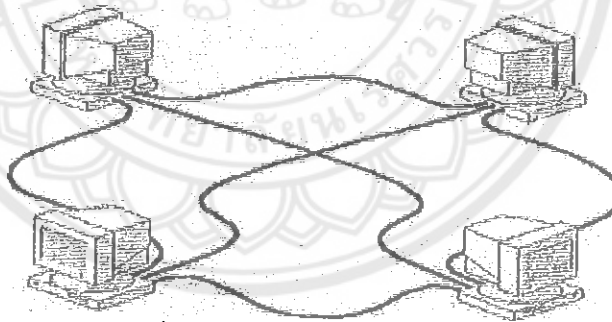
โทโปโลยีแบบวงแหวนนี้จะใช้สายสัญญาณเชื่อมต่อคอมพิวเตอร์เป็นห่วงหรือเป็นวงแหวน การเชื่อมต่อแบบนี้สัญญาณจะเดินทางเป็นวงกลมในทิศทางเดียว และจะวิ่งผ่านคอมพิวเตอร์แต่ละเครื่อง ซึ่งจะทำหน้าที่ทวนสัญญาณไปในตัวแล้วผ่านไปเครื่องถัดไป รูปที่ 2.30 เป็นการเชื่อมต่อแบบโทโปโลยีแบบวงแหวนของคอมพิวเตอร์ 4 เครื่อง ถ้าคอมพิวเตอร์เครื่องใดเครื่องหนึ่งหยุดทำงานก็จะทำให้ระบบเครือข่ายล่มเช่นกัน



รูปที่ 2.6 โทโปโลยีแบบวงแหวน

2.5.4 โทโปโลยีแบบเมช (Mesh Topology)

คือ การเชื่อมต่อคอมพิวเตอร์แบบสมบูรณ์ กล่าวคือ คอมพิวเตอร์ทุกเครื่องในเครือข่ายจะเชื่อมต่อถึงกันหมด โดยใช้สายสัญญาณทุกการเชื่อมต่อ วิธีการนี้จะเป็นการสำรองเส้นทางเดินของข้อมูลได้เป็นอย่างดี



รูปที่ 2.7 แสดงการเชื่อมต่อแบบเมช

2.6 อีเทอร์เน็ต

ปัจจุบันเป็นที่ยอมรับกันว่าอีเทอร์เน็ตนั้น เป็นเทคโนโลยีเครือข่ายที่เป็นมาตรฐานหลักของเทคโนโลยีสารสนเทศทั้งหมด เนื่องจากเป็นเทคโนโลยีเครือข่ายแบบท้องถิ่นที่นิยมมากที่สุด อีเทอร์เน็ตมีอายุกว่า 30 ปีแล้ว และได้มีการพัฒนาอย่างต่อเนื่อง ด้วยเหตุนี้จึงเป็นการยากที่จะพัฒนาเทคโนโลยีใหม่มาแทนที่ได้ เทคโนโลยีนี้จึงได้ถูกพัฒนาและปรับปรุงภายใต้การดูแลและรับผิดชอบ

ของสถาบัน IEEE (Institute of Electrical and Electronics Engineer) โดยสิ่งที่สำคัญอย่างหนึ่งในการเปลี่ยนแปลงและปรับปรุงคือ การเพิ่มความเร็วในการรับส่งข้อมูลหรือแบนด์วิธ

2.6.1 ประวัติอีเทอร์เน็ต

ในปี ค.ศ. 1973 บ็อบ เม็ทคาลเฟ (Bob Metcalfe) ได้คิดค้นระบบอีเทอร์เน็ตในการรับ - ส่งข้อมูลในระหว่างคอมพิวเตอร์และสามารถส่งข้อมูลไปยังเครื่องพิมพ์ได้ หลังจากนั้นอีเทอร์เน็ตถูกพัฒนาต่อที่ PARC (Palo Alto Research Center) ซึ่งเป็นศูนย์วิจัยของบริษัทซีร็อกซ์ (Xerox) คอมพิวเตอร์จะเชื่อมกันเป็นเครือข่ายโดยใช้ทรานสซีฟเวอร์ (Transceiver) และสายสัญญาณสำหรับการรับส่งข้อมูล จุดประสงค์ของการสร้างอีเทอร์เน็ต ในตอนแรกนั้นเพื่อให้ นักวิจัยสามารถแชร์ข้อมูลร่วมกันได้เท่านั้น ไม่ใช่เป็นการพัฒนาเทคโนโลยีใหม่ ในสมัยแรกจะใช้สายโคแอกซ์ซีลแบบหนาเป็นสายสัญญาณในการเชื่อมต่อคอมพิวเตอร์เหล่านั้น ในตอนนั้นอีเทอร์เน็ตก็จะถือเป็นเทคโนโลยีที่หน้าทึ่มากๆ ในการใช้คอมพิวเตอร์ในสมัยนั้น เพราะคอมพิวเตอร์ส่วนใหญ่ก็จะเป็นเครื่องเมนเฟรมที่มีราคาแพงมาก มีน้อยคนที่สามารถซื้อระบบเมนเฟรมมาใช้ และคนส่วนใหญ่จะไม่รู้จักรการใช้งานเฟรม แต่การพัฒนาอีเทอร์เน็ตทำให้การใช้คอมพิวเตอร์แพร่หลายมากขึ้น

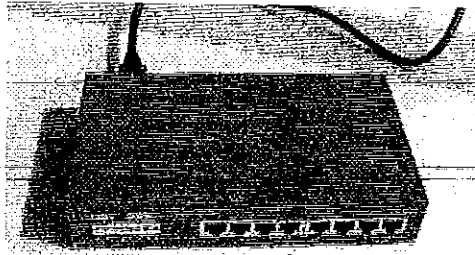
เมื่อในปี 1973 เม็ทคาลเฟได้เขียนอธิบายระบบเครือข่ายที่มีการพัฒนามาจากเครือข่ายอลฮา (Aloha) ซึ่งได้ถูกพัฒนาที่มหาวิทยาลัยฮาวาย ในทศวรรษ 1960 โดยนอร์แมน แอ็บรามสัน (Norman Abramson) และเพื่อนร่วมงาน โดยใช้พัฒนาระบบสื่อสารวิทยุสื่อสารระหว่างเกาะต่างๆ การพัฒนานี้เป็นการพัฒนาระบบเพื่อแชร์สื่อกลางการรับส่งข้อมูลซึ่งในที่นี้คือ อากาศที่เป็นสื่อนำคลื่นวิทยุนั่นเอง IEEE ได้ถูกตีพิมพ์มาตรฐานอีเทอร์เน็ตตั้งแต่ในปี 1985 แล้วก็ได้มีการพัฒนามาตรฐานมาเรื่อยๆ มาตรฐานแรกนั้นก็ใช้สายโคแอกซ์แบบหนา และต่อมาก็ได้เปลี่ยนมาใช้สายโคแอกซ์แบบบาง หลังจากนั้นได้พัฒนาสายสัญญาณอื่นๆ เช่น สายคู่เกลียวบิดและสายใยแก้ว เป็นต้น และได้มีการปรับปรุงความเร็วเป็น 100 Mbps และ 1,000 Mbps ปัจจุบันมาตรฐานล่าสุดของอีเทอร์เน็ตอยู่ที่ 10 Gbps

2.6.2 อุปกรณ์เครือข่ายอีเทอร์เน็ต

อุปกรณ์ที่ใช้ในระบบเครือข่ายนั้นทำหน้าที่จัดการเกี่ยวกับการรับ - ส่งข้อมูลในเครือข่าย หรือใช้สำหรับทวนสัญญาณ เพื่อให้การส่งข้อมูลได้ในระยะที่ไกลขึ้น หรือใช้สำหรับขยายให้มีขนาดใหญ่ขึ้น อุปกรณ์เครือข่ายที่พบเห็นโดยทั่วไป เช่น ฮับ สวิตช์ เราท์เตอร์ เป็นต้น

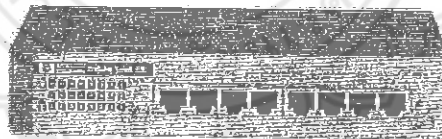
2.6.2.1 ฮับ (HUB) หรือบางที่เรียกว่า “รีพีทเตอร์ (Repeater)” คืออุปกรณ์ที่ใช้เชื่อมต่อกลุ่มของ

คอมพิวเตอร์ฮับมีหน้าที่รับส่งเฟรมข้อมูลทุกเฟรมที่ได้รับจากพอร์ตหนึ่งไปยังทุกๆ พอร์ตที่เหลือของคอมพิวเตอร์ที่เชื่อมต่อเข้ากับฮับ



รูปที่ 2.8 ฮีเทอร์เน็ตฮับ

2.6.2.2 สวิตช์ (Switch) คืออุปกรณ์เครือข่ายที่ทำหน้าที่ในเลเยอร์ที่ 2 สวิตช์บางทีก็เรียกว่า “สวิตช์ซึ่งฮับ (Switch Hub)” ซึ่งในช่วงแรกนั้นจะเรียกว่า “บริดจ์ (Bridge)” เหตุที่เรียกบริดจ์ในช่วงแรกนั้นเพราะส่วนใหญ่บริดจ์จะมีแค่สองพอร์ต และใช้สำหรับแยกคอลลิชัน โดเมน ปัจจุบันที่เรียกสวิตช์เพราะหมายถึงบริดจ์ที่มีมากกว่าสองพอร์ตนั่นเอง สวิตช์จะฉลาดกว่าฮับคือ สวิตช์สามารถส่งข้อมูลที่รับมาจากพอร์ตหนึ่งไปยังเฉพาะพอร์ตที่เป็นปลายทางเท่านั้น ทำให้คอมพิวเตอร์ที่เชื่อมต่อกับพอร์ตที่เหลือสามารถส่งข้อมูลถึงกันและกันได้ในเวลาเดียวกัน



รูปที่ 2.9 สวิตช์

2.6.2.3 เลเยอร์ 3 สวิตช์ (Layer 3 Switch) เมื่อพูดถึงสวิตช์นั้น หมายถึงอุปกรณ์เครือข่ายที่จะทำงานในเลเยอร์ที่ 2 แต่ในปัจจุบันเทคโนโลยีการผลิตอุปกรณ์เครือข่ายนั้นพัฒนาไปค่อนข้างมาก สวิตช์ที่มีในท้องตลาดปัจจุบันบางประเภทสามารถรับรองการทำงานที่เลเยอร์ที่ 3 ได้ด้วย ซึ่งอุปกรณ์เครือข่ายที่ทำหน้าที่ทั้งในเลเยอร์ที่ 2 และในเลเยอร์ที่ 3 สำหรับข้อแตกต่างระหว่างเลเยอร์ 3 สวิตช์และเราท์เตอร์อีกอย่างก็คือ สวิตช์นั้นจะผลิตโดยใช้เทคโนโลยีที่เรียกว่า “ASIC (Application Specific Circuit)” หรือเป็นวงจรรวมที่สร้างสำหรับสวิตช์โดยเฉพาะ ส่วนเราท์เตอร์นั้นโดยทั่วไปจะสร้างมา

จากโปรเซสเซอร์ทั่วไปและมีซอฟต์แวร์ที่ควบคุมการทำงานอีกครั้งหนึ่ง ดังนั้นการทำงานของสวิตซ์ก็จะเร็วกว่าเราเตอร์มาก

2.6.2.4 คอลลิชัน โดเมน (Collision Domain) หลักการที่สำคัญอย่างหนึ่งสำหรับการออกแบบเครือข่ายเน็ตเวิร์กก็คือ คอลลิชัน โดเมน ซึ่งหมายถึงส่วนของเครือข่ายที่แชร์ช่องสัญญาณ ในการรับ - ส่งข้อมูลเดียวกัน ส่วนของเครือข่ายนี้อาจประกอบด้วยคอมพิวเตอร์ สายสัญญาณ และ รีพีทเตอร์ เป็นต้น รีพีทเตอร์หรือฮับเป็นอุปกรณ์อีเธอร์เน็ตที่ทำงานในระดับฟิสิคอลลเยอร์ ซึ่งทำหน้าที่หลักก็คือทวนสัญญาณที่ได้รับจากพอร์ตหนึ่งไปยังพอร์ตที่เหนือ ส่วนสวิตซ์นั้นจะทำงานในเคาต์ดิงค์เลเยอร์ซึ่งจะตรวจสอบหมายเลข MAC หรืออยู่ที่เลเยอร์ที่ 2 ก่อนที่จะส่งต่อสัญญาณ ไปยังพอร์ตปลายทางเท่านั้น

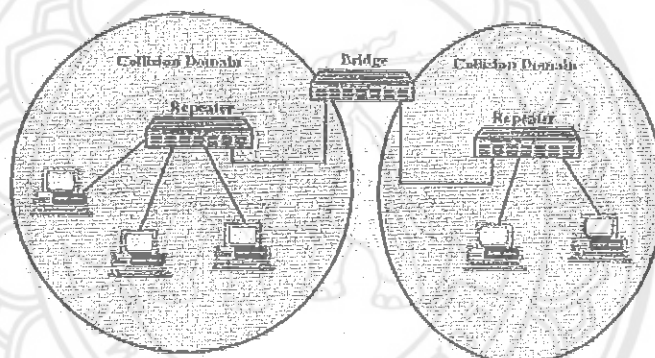
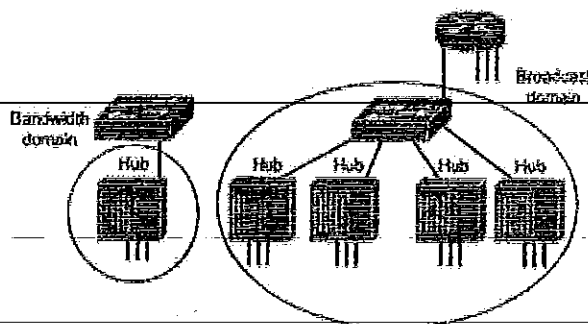


Figure 10.2 100BASE-T Collision Domains

รูปที่ 2.10 คอลลิชัน โดเมน

2.6.2.5 บรอดคาสต์โดเมน (Broadcast Domain) มัลติคาสต์โดเมน หมายถึง กลุ่มของหมายเลข MAC ซึ่งแต่ละโหนดสามารถโปรแกรมให้อยู่ในกลุ่มนี้ได้ ส่วนบรอดคาสต์โดเมนนั้นเป็นกรณีพิเศษของมัลติคาสต์โดเมน กล่าวคือ บรอดคาสต์โดเมน หมายถึง ทุกโหนดที่อยู่ในวง LAN เดียวกัน ดังนั้นเฟรมข้อมูลที่ส่งไปยังบรอดคาสต์โดเมนทุกๆ โหนดที่เชื่อมต่อเข้ากับเครือข่ายก็จะได้รับเฟรมนั้น สวิตซ์ถูกออกแบบมาสำหรับเชื่อมต่อหลายๆคอลลิชันโดเมนในวง LAN เดียวกัน ดังนั้นสวิตซ์จะทำการ ฟลัด (Flood) หรือส่งเฟรมข้อมูลแบบบรอดคาสต์ไปยังทุกๆพอร์ตของสวิตซ์ ยกเว้นพอร์ตที่รับเฟรมข้อมูลนั้นมา ด้วยวิธีเฟรมแบบบรอดคาสต์สามารถส่งไปยังทุกๆ โหนดในเครือข่าย ดังนั้นบางทีสวิตซ์ก็ทำหน้าที่เป็นรีพีทเตอร์เหมือนกัน



รูปที่ 2.11 บรอดคาสต์โดเมน

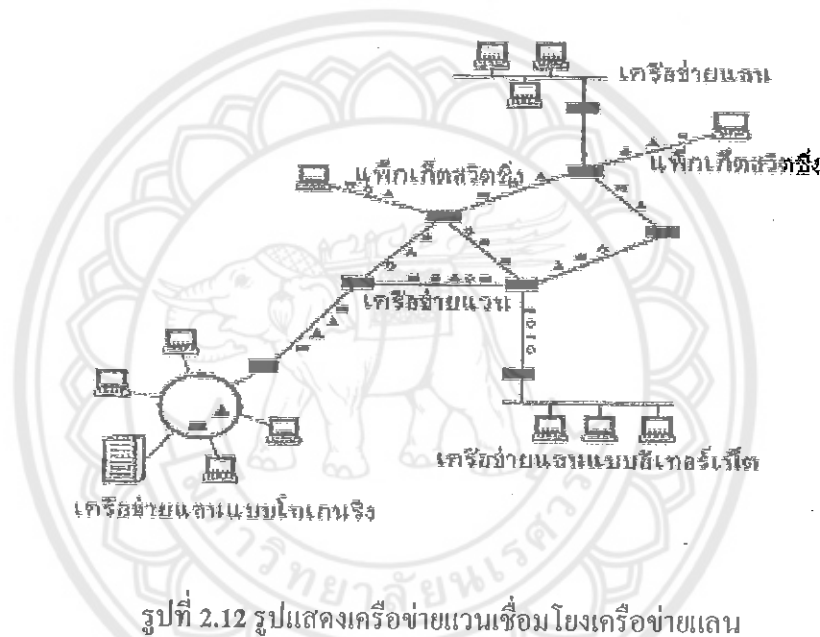
การส่งเฟรมข้อมูลแบบมัลติคาสต์หรือบรอดคาสต์นั้น มีข้อดีอยู่หลายประการ บางโปรโตคอลในแลเยอร์ที่เหนือกว่าก็จะใช้การส่งข้อมูลแบบบรอดคาสต์ เพื่อสำหรับการค้นหาที่อยู่ในแลเยอร์นั้น เช่น โปรโตคอล DHCP (Dynamic Host Configuration Protocol) จะใช้การส่งข้อมูลแบบบรอดคาสต์

2.7 เทคโนโลยี WAN

เทคโนโลยี LAN (Local Area Network) ใช้ในการสร้างเครือข่ายที่ครอบคลุมพื้นที่จำกัดเท่านั้น แต่ความต้องการเชื่อมต่อคอมพิวเตอร์เป็นเครือข่ายไม่ได้จำกัดเฉพาะในพื้นที่ใดพื้นที่หนึ่งเท่านั้น คอมพิวเตอร์ที่อยู่ห่างกันมากบางทีก็จำเป็นติดต่อสื่อสารเช่นกัน ถ้ามองอีกมุมหนึ่งคอมพิวเตอร์ก็จะมีลักษณะนิสัยคล้ายมนุษย์ก็คือ มนุษย์เป็นสัตว์สังคม ซึ่งต้องติดต่อสื่อสารซึ่งกันและกัน คอมพิวเตอร์ก็เช่นกันต้องติดต่อสื่อสารกับคอมพิวเตอร์เครื่องอื่นๆ ไม่ว่าจะอยู่ใกล้หรือไกล ถ้าคอมพิวเตอร์ที่อยู่ใกล้กันก็จะใช้เทคโนโลยี LAN ในการเชื่อมต่อกัน แต่ถ้าอยู่ห่างไกลออกไปก็จะใช้เทคโนโลยี WAN (Wide Area Network) ในการเชื่อมต่อ ในบทนี้จะขอแนะนำให้รู้จักกับเทคโนโลยี WAN ที่สำคัญและใช้ในการสร้างเครือข่ายในปัจจุบัน

WAN (Wide Area Network) เป็นเทคโนโลยีใช้สำหรับการเชื่อมต่อ LAN (Local Area Network) ที่อยู่ห่างไกลกัน และไม่สามารถเชื่อมต่อกันได้โดยเทคโนโลยี LAN ตัวอย่างเครือข่าย WAN ที่รู้จักกันดีและเป็นเครือข่ายที่ใหญ่ที่สุดในโลกก็คือ อินเทอร์เน็ต ซึ่งเป็นเครือข่ายที่ครอบคลุมทั่วโลก ข้อจำกัดในการออกแบบเครือข่าย WAN นั้นคือ ระยะทาง เพราะไม่ว่าจะเป็นสัญญาณประเภทใดก็ตามแล้วแต่เมื่อต้องส่งไประยะไกลๆกำลังของสัญญาณนั้นๆก็จะอ่อนลง ซึ่งมีผลกระทบต่อประสิทธิภาพในการรับส่งข้อมูล การออกแบบ WAN น้อยกว่าของ LAN มาก แต่รับส่งข้อมูลได้ระยะไกลกว่า

เทคโนโลยี WAN มีความสำคัญเพิ่มขึ้นเรื่อยๆ ในช่วงเริ่มแรกความต้องการในการเชื่อมต่อคอมพิวเตอร์ที่อยู่ห่างไกลกันมาก เป็นแค่การต่อระหว่างคอมพิวเตอร์ไม่กี่เครื่องเท่านั้น แต่ปัจจุบันจะเป็นการเชื่อมต่อ LAN หลายๆ ที่อยู่ห่างไกลกัน ซึ่งเอื้ออำนวยให้คอมพิวเตอร์ที่อยู่ต่าง LAN กันสามารถสื่อสารกันผ่านเครือข่าย WAN ได้ เช่น การเชื่อมต่อเครือข่ายของสำนักงานย่อยเข้ากับเครือข่ายของสำนักงานใหญ่ที่อยู่ในคนละเมืองได้ ในปัจจุบันหลายๆองค์กรมีเครือข่ายอินเทอร์เน็ต และมีความต้องการที่จะเชื่อมต่อเข้ากับอินเทอร์เน็ตด้วยเหตุผลทางธุรกิจ เช่น เป็นจุดติดต่อลูกค้า เป็นต้น การเชื่อมต่อเข้ากับเครือข่ายอินเทอร์เน็ตนั้นต้องใช้เทคโนโลยี WAN รูปที่ 2.45 แสดงการเชื่อมต่อเครือข่าย LAN



2.7.1 การเชื่อมต่อ WAN ด้วย LAN

เทคโนโลยี WAN นั้นจะแตกต่างกับเทคโนโลยี LAN มาก เทคโนโลยี LAN นั้นส่วนใหญ่จะมีมาตรฐานรองรับ แต่เทคโนโลยี WAN จะประกอบด้วยส่วนต่างๆ ที่สร้างจากหลายบริษัท บางส่วนก็มีมาตรฐาน บางส่วนก็เป็นเทคโนโลยีเฉพาะบริษัท ซึ่งจะแตกต่างกันทั้งทางด้านลักษณะ ประสิทธิภาพ และราคาที่ยากที่สุดในการสร้างเครือข่าย WAN คือการเลือกเทคโนโลยีที่สามารถทำงานร่วมกันได้ และสนองความต้องการของธุรกิจ ซึ่งจำเป็นอย่างยั้งที่คงเข้าใจทุกๆ ส่วนของเทคโนโลยี WAN เทคโนโลยี WAN มีองค์ประกอบที่สำคัญดังนี้

- ระบบส่งสัญญาณ (Transmission Facility)

- อุปกรณ์เครือข่าย เช่น เราท์เตอร์, สวิตช์, CSU/DSU (Channel Service Unit/Data Service Unit)

- ระบบจัดการที่อยู่ (Internet network Addressing)

- โพรโตคอลจัดเส้นทาง (Routing Protocol)

2.8 อินเทอร์เน็ตเลเยอร์ (Internet Layer)

โพรโตคอลที่สำคัญที่ทำงานในเลเยอร์อินเทอร์เน็ตคือ IP, ARP, ICMP และ IGMP ซึ่งความเข้าใจหลักการทำงานของโพรโตคอลนี้

2.8.1 Internet Protocol (IP)

โพรโตคอล IP จะทำหน้าที่เปรียบเสมือนกับที่ทำการไปรษณีย์ กล่าวคือ โพรโตคอล IP จะทำหน้าที่จัดการเกี่ยวกับการรับส่งแพ็กเก็ต หรือบางทีก็เรียกว่า “ดาต้าแกรม (Datagram)” คือหน่วยของข้อมูลที่รับมาจากโพรโตคอลที่อยู่เลเยอร์สูงกว่า เช่น TCP และ UDP ถ้าโฮสต์ปลายทางอยู่บนละเครือข่ายกับโฮสต์ที่ส่งข้อมูล IP จะรับผิดชอบในการจัดเส้นทาง (Routing) ให้แพ็กเก็ตส่งไปยังเครือข่ายที่โฮสต์นั้นอยู่ ซึ่งในการจัดส่งแพ็กเก็ตข้ามเครือข่ายนั้น IP จะใช้เราท์เตอร์ (Routing) ในการเชื่อมต่อเครือข่ายเหล่านั้นโดยทั่วไปแล้วอุปกรณ์ที่ทำหน้าที่รับส่งข้อมูลระหว่างเครือข่ายจะเรียกว่าเราท์เตอร์ แต่บางทีอุปกรณ์ตัวนี้ก็เรียกว่า “เกตเวย์ (Gateway)” ซึ่งจะทำหน้าที่เป็นเสมือนประตูไปยังเครือข่ายอื่นๆ อย่างไรก็ตามทั้งเราท์เตอร์ และเกตเวย์เป็นอุปกรณ์ที่ทำหน้าที่ในเลเยอร์ 3 เหมือนกัน โพรโตคอล IP นั้นจะเป็นโพรโตคอลที่ให้บริการแบบคอนเนกชันเลสส์ (Connectionless) ซึ่งทำให้มีความเชื่อถือได้น้อย เนื่องจากไม่มีโครงสร้างการเชื่อมต่อก่อนที่จะทำการรับ - ส่งข้อมูล กล่าวคือ ในการส่งข้อมูลในแต่ละครั้ง โฮสต์ส่งจะไม่ทำการติดต่อโฮสต์ปลายทางเพื่อตกลงเกี่ยวกับการรับส่งข้อมูลก่อน แต่โฮสต์ที่ต้องการส่งข้อมูลจะทำการส่งแพ็กเก็ตออกไปทันที โดยที่คาดหวังว่าโฮสต์ปลายทางจะได้รับแพ็กเก็ตนั้นในที่สุด ดังนั้นความเชื่อถือในการส่งข้อมูลจึงมีน้อยเพราะแพ็กเก็ตอาจสูญหายระหว่างทาง หรือถ้าข้อมูลประกอบด้วยหลายแพ็กเก็ต แต่แพ็กเก็ตอาจเดินทางมาถึงปลายทางไม่เป็นลำดับได้ หรือมีการส่งแพ็กเก็ตซ้ำกันหรือแพ็กเก็ตส่งถึงล่าช้า การแก้ปัญหานี้จะปล่อยให้ทำหน้าที่ของโพรโตคอลที่อยู่ในเลเยอร์ที่สูงกว่ารับผิดชอบ

2.8.2 Address Resolution Protocol (ARP)

การที่คอมพิวเตอร์ที่อยู่ในเครือข่ายเดียวกันต้องการที่จะสื่อสารกันจำเป็นต้องทราบหมายเลขเน็ตเวิร์กการ์ด หรือแม็กแอดเดรส (MAC Address) ของกันและกัน แพ็กเก็ตไอพีจะถูกห่อหุ้มด้วยเฟรมในระดับดาต้าลิงก์ ซึ่งแม็กแอดเดรสของเครื่องส่งและเครื่องรับจะต้องถูกใส่ไปด้วยปัญหาที่ก็คือเครื่องส่ง

อาจไม่ทราบหมายเลขแม็กแอดเดรสของเครื่องรับ โพรโทคอล ARP (Address Resolution Protocol) จะทำหน้าที่ค้นหาหมายเลขแม็กแอดเดรสของเครื่องที่มีหมายเลขไอพีที่ต้องการ หลักการทำงานของ ARP คือ โฮสต์ที่ต้องการทราบหมายเลขแม็กแอดเดรสของเครื่องที่มีหมายเลขไอพีนั้น จะทำการ broadcast ภาสตร์แพ็กเก็ตไปยังคอมพิวเตอร์ทุกเครื่องที่อยู่ในเครือข่ายเดียวกัน ถ้ามีเครื่องที่มีหมายเลขไอพีดังกล่าว-เครื่องนั้นก็จะตอบกลับพร้อมหมายเลขแม็กแอดเดรสของเครื่องนั้น-และหลังจากนั้น เครื่องส่งก็สามารถสื่อสารกับเครื่องปลายทางได้โดยตรง โดยใช้แม็กแอดเดรสที่ส่งมาพร้อมกับแพ็กเก็ตตอบกลับ ส่วนโพรโทคอลที่ทำหน้าที่ในทางตรงกันข้ามกับ โพรโทคอล ARP คือ โพรโทคอล RARP (Reverse Address Resolution Protocol) ซึ่งโพรโทคอลนี้จะช่วยให้โฮสต์ที่รู้หมายเลขแม็กแอดเดรสแต่ไม่รู้หมายเลขไอพี

2.8.3 Internet Control Messages Protocol (ICMP)

โพรโทคอล ICMP (Internet Control Messages Protocol) ทำหน้าที่รายงานข้อผิดพลาดต่าง ๆ ที่เกิดขึ้นในระหว่างที่มีการส่งแพ็กเก็ตในเครือข่าย ICMP ใช้ในการส่งแบบคอนเน็กชันเลสส์ (Connectionless) ซึ่งหมายถึงการรับส่งข้อมูลที่ฝ่ายรับและฝ่ายส่งไม่ได้ประสานกันก่อน กล่าวคือ ฝ่ายรับจะไม่ทราบว่าจะมีแพ็กเก็ตส่งมาหาตัวเอง ดังนั้นโอกาสที่แพ็กเก็ตจะส่งไม่ถึงปลายทางจึงเป็นไปได้สูง

2.8.4 Internet Group Management Protocol (IGMP)

โพรโทคอล IGMP (Internet Group Management Protocol) ทำหน้าที่แจ้งให้เราท์เตอร์เกี่ยวกับกลุ่มของเครื่องโฮสต์ที่เป็นมัลติคาสต์ (Multicast) ซึ่งข้อมูลนี้จะถูกส่งต่อ ๆ กันออกไปยังเราท์เตอร์ต่าง ๆ ที่อยู่ในเครือข่ายเพื่อให้เครือข่ายสามารถรองรับการรับส่งข้อมูลแบบมัลติคาสต์ได้ การส่งแพ็กเก็ตของ IGMP จะส่งเป็นไอพีดาต้าแกรมซึ่งเป็นการส่งแบบคอนเน็กชันเลสส์ โฮสต์ทูโฮสต์เลเยอร์ (Host to Host Layer)

อย่างที่กล่าวมาแล้วข้างต้น โพรโทคอลโฮสต์ทูโฮสต์เลเยอร์ (Host to Host Layer) นี้จะประกอบด้วย 2 โพรโทคอลคือ TCP (Transmission Control Protocol) UDP (User Datagram Protocol) ซึ่งเป็นโพรโทคอลแต่ละตัวจะบริการแตกต่างกัน และมีข้อดีข้อเสียต่างกัน โพรโทคอลทั้งสองตัวมีรายละเอียดดังนี้

2.8.4.1 Transmission Control Protocol (TCP)

โปรโตคอล TCP (Transmission Control Protocol) เป็นโปรโตคอลที่ให้บริการแบบคอนเน็กชันโอเรียนเต็ด (Connection-Oriented) ซึ่งเป็นการส่งข้อมูลที่เชื่อถือได้ TCP จะส่งข้อมูลทั้งหมดจนสำเร็จ ซึ่งถ้าข้อมูลมีขนาดใหญ่ก็จะถูกแบ่งย่อยเป็นหลายแพ็กเก็ต โปรโตคอล TCP จะทำหน้าที่ควบคุมการรับส่งแพ็กเก็ตข้อมูลย่อย ๆ เหล่านี้ สำหรับกลไกในการควบคุมการไหลของข้อมูลมีรายละเอียดดังนี้

2.8.4.2 การจัดการเกี่ยวกับเซสชัน

เนื่องจาก TCP เป็นโปรโตคอลที่ให้บริการแบบคอนเน็กชันโอเรียนเต็ด ดังนั้นก่อนการจะส่งข้อมูลจำเป็นต้องสร้างเซสชันเพื่อเชื่อมต่อกับโฮสต์ปลายทางก่อนเซสชันเป็นการสร้างการสนทนาอย่างเป็นทางการระหว่างทั้งสองโฮสต์เพื่อใช้สำหรับการกู้คืนข้อมูล เมื่อเกิดข้อผิดพลาดระหว่างการรับส่งข้อมูล ขั้นตอนในการสร้างเซสชันจะมีอยู่ 3 ขั้นตอนซึ่งบางทีก็เรียกว่า “ทรีเวย์แฮนด์เช็ก (Three-Way Handshake) โฮสต์ที่ต้องการส่งข้อมูลจะส่งแพ็กเก็ตไปยังโฮสต์ปลายทาง เพื่อแจ้งให้ทราบว่าต้องการส่งข้อมูล โฮสต์ปลายทางก็จะตอบตกลงมา พร้อมทั้งรหัสที่จะใช้ในการรับส่งข้อมูล โฮสต์ต้นทางก็จะส่งแพ็กเก็ตพร้อมรหัสที่ได้รับ เพื่อเป็นการยืนยันการเชื่อมต่อ หลังจากที่ได้มีการสร้างเซสชันสำเร็จแล้วถึงเริ่มขบวนการรับ-ส่งข้อมูลจริงๆ ซึ่งในการรับส่งข้อมูลในแต่ละครั้งก็จะมีการยืนยันการรับส่งข้อมูลจากโฮสต์ปลายทางทุกครั้ง เมื่อรับส่งข้อมูลเสร็จก็เป็นขั้นตอนการยกเลิกการเซสชัน ซึ่งจะคล้าย ๆ กับการสร้างเซสชัน

2.8.4.3 การควบคุมการไหลเวียนและกู้คืนข้อมูล

ในแต่ละเซสชัน โฮสต์ฝ่ายรับต้องตอบกลับทุก ๆ แพ็กเก็ตที่ได้รับภายในเวลาที่กำหนด เพื่อเป็นการยืนยันการรับข้อมูลทุก ๆ แพ็กเก็ตที่ส่ง ฝ่ายรับจะทำการเช็คความถูกต้องของแพ็กเก็ตข้อมูลทุกครั้ง และแจ้งให้ทราบถึงการตรวจตอบนั้น ถ้าฝ่ายส่งไม่ได้รับการตอบรับจากฝ่ายรับภายในเวลาที่กำหนด ฝ่ายรับก็จะคาดเดาว่าแพ็กเก็ตสูญหายระหว่างทาง ฝ่ายรับก็จะทำการส่งแพ็กเก็ตนั้นให้ใหม่อีกครั้ง เพื่อจะทำให้มั่นใจได้ว่าข้อมูลทุก ๆ แพ็กเก็ตส่งถึงปลายทางอย่างสมบูรณ์ ข้อมูลในส่วนหัวของโปรโตคอล TCP จะประกอบด้วยข้อมูลมากที่สุด 20 ไบต์ และประกอบด้วยส่วนต่าง ๆ ซึ่งมีฟิลด์มีความหมายดังนี้

TCP Source Port (16 บิต) : ส่วนนี้จะเป็นหมายเลขพอร์ตที่เป็นจุดเริ่มการสื่อสาร หมายเลขพอร์ตเมื่อรวมกับหมายเลข IP จะเป็นที่อยู่ของการส่งข้อมูลกลับ

TCP Destination Port (16 บิต) : เป็นหมายเลขพอร์ตเครื่องรับ ซึ่งพอร์ตนี้จะเป็นพอร์ตที่ใช้เชื่อมต่อกับแอปพลิเคชันที่จะนำข้อมูลที่ส่งไปให้ไปโปรเซสต่อไป

TCP Sequence Number (32 บิต) : เป็นหมายเลขที่บอกลำดับแพ็กเก็ตที่จะใช้ โดยฝั่งเครื่องรับในการเรียงข้อมูลให้อยู่ในรูปเดิมในการส่งข้อมูลผ่านเครือข่ายที่สลับซับซ้อนนั้นแพ็กเก็ตแต่ละชุดอาจจะถูกส่งไปบนเส้นทางที่ต่างกัน ดังนั้นจึงเป็นไปได้ที่แพ็กเก็ตจะเดินทางมาถึงปลายทางไม่เป็นไปตามลำดับที่ส่ง หมายเลขนี้จะใช้ในการจัดเรียงแพ็กเก็ตเหล่านี้ให้อยู่ในลำดับเดิม

TCP Acknowledgment Number (32 บิต) : เป็นหมายเลขลำดับแพ็กเก็ตถัดไปที่ทางฝั่งรับคาดหวัง ซึ่งเป็นการบอกเป็นนัยว่าแพ็กเก็ตที่มีหมายเลขลำดับก่อนหน้านี้นี้ได้รับหมดแล้วนั่นเอง

Data Offset (4 บิต) : เป็นตัวเลขที่บอกขนาดของข้อมูลส่วนหัว (TCP Header) ซึ่งมีหน่วยเป็น 32 บิต หรือ Word .

Reserved (6 บิต) : ส่วนนี้จะถูกกำหนดให้เป็นศูนย์ตลอด ซึ่งข้อมูลส่วนนี้ไม่มีความหมายอะไรแต่เป็นการสงวนไว้ใช้ในอนาคตเมื่อมีการปรับปรุงโปรโตคอล

Flags (6 บิต) : เป็นข้อมูลที่ใช้สำหรับควบคุมการรับส่งแพ็กเก็ต

Window Size (16) : เป็นตัวเลขที่เครื่องปลายทางบอกให้เครื่องต้นทางทราบขนาดวินโดว์ของเครื่องปลายทางสามารถรับรู้ได้

Checksum (16) : เป็นข้อมูลที่ใช้ในการตรวจสอบข้อผิดพลาดของข้อมูลในส่วนหัว โดยเครื่องส่งจะทำการคำนวณค่าเช็คซัม (Checksum) ของข้อมูลส่วนหัว เมื่อเครื่องปลายทางได้รับข้อมูลก็จะทำการคำนวณเช็คซัมด้วยวิธีเดียวกัน แล้วทำการเปรียบเทียบข้อมูลค่าที่คำนวณได้กับค่าที่อยู่ในฟิลด์นี้ ถ้าเหมือนกันแสดงว่าไม่มีข้อผิดพลาดในข้อมูลที่ได้รับ

Padding : เป็นข้อมูลที่เพิ่มเพื่อให้ข้อมูลส่วนหัวมีจำนวนบิตที่หารด้วย 32 ลงตัว

2.8.4.4 User Datagram Protocol (UDP)

โปรโตคอล UDP (User Datagram Protocol) จะให้บริการการส่งข้อมูลแบบคอนเน็กชันเลสส์ หรือบางทีก็เรียกว่า “ดาต้าแกรม (Datagram)” ซึ่งจะเป็นการให้บริการแบบตรงกันข้ามกับคอนเน็กชันโอเรียนเต็ลของโปรโตคอล TCP การส่งข้อมูลแบบนี้จะเป็นแบบที่เชื่อถือไม่ได้ โดยจะพยายามส่งข้อมูลให้ดีที่สุด ในการรับส่งข้อมูลในแต่ละครั้งนั้น จะไม่มีการสร้างเซกชันก่อน และไม่มีกลไกการตอบกลับแพ็กเก็ตเหมือนโปรโตคอล TCP เหตุที่ตัดกลไกนี้ออกเพื่อเพิ่มประสิทธิภาพในการส่งข้อมูลนั่นเอง แต่ข้อเสียก็คือ การรับส่งเชื่อถือไม่ได้ เพราะแพ็กเก็ตอาจสูญหายระหว่างทางซึ่งทางฝ่ายส่งจะไม่ทราบเลย ดังนั้น โปรโตคอลที่อยู่เหนือกว่าต้องรับผิดชอบเกี่ยวกับการตรวจสอบข้อผิดพลาดของการรับส่งข้อมูลเอง

ถึงแม้ว่าโปรโตคอล UDP จะมีความเชื่อถือได้น้อย แต่มันก็มีข้อดีหลายอย่าง เช่น ถ้าข้อมูลที่ต้องการส่งมีขนาดเล็กมากก็จะเป็นการเสียเวลา ถ้าต้องสร้างเซกชันการเชื่อมต่อระหว่าง 2 โฮสต์นั้นก่อนส่ง และอีกกรณีหนึ่งคือ การส่งข้อมูลแบบแพร่กระจาย หรือบรอดคาสต์ (Broadcast) และมัลติคาสต์ (Multicast) การสร้างเซกชันจะเป็นสิ่งที่เป็นไปไม่ได้ เนื่องจากเซกชันเป็นการเชื่อมต่อระหว่าง 2 โฮสต์เท่านั้น ดังนั้นการบรอดคาสต์และมัลติคาสต์จะใช้โปรโตคอล UDP เท่านั้น ข้อมูลในส่วนหัวของโปรโตคอล UDP ซึ่งในแต่ละฟิลด์มีความหมายดังนี้

UDP Source Port Number (16 บิต): เป็นหมายเลขพอร์ตของเครื่องส่ง เมื่อรวมหมายเลขพอร์ตนี้กับหมายเลข IP ก็จะเป็นที่อยู่สำหรับเครื่องรับในการตอบรับข้อความ

UDP Destination Port Number (16 บิต): เป็นหมายเลขพอร์ตของทางฝั่งเครื่องรับที่ใช้ในการส่งผ่านข้อมูลไปยังแอปพลิเคชันที่ต้องการติดต่อด้วย

UDP Checksum (16 บิต): เป็นข้อมูลที่ใช้ในการตรวจสอบข้อผิดพลาดของข้อมูล เครื่องทางฝั่งรับจะทำการคำนวณหมายเลขนี้ด้วยวิธีเดียวกัน แล้วเปรียบเทียบกับค่าที่ส่งมา ถ้าหมายเลขเท่ากัน แสดงว่าไม่มีข้อผิดพลาดในข้อมูลส่วนหัว

UDP Message Length (16 บิต): เป็นข้อมูลที่บอกความยาวของข้อมูลทั้งหมด ซึ่งจะเป็นข้อมูลที่ช่วยให้ทางฝ่ายรับทราบว่าข้อมูลควรมีขนาดเท่าใด

2.9 Routing Protocol

เราเตอร์จะใช้ข้อมูลที่อยู่ในตารางเราต์ติ้งเทเบิลสำหรับการส่งแพ็กเก็ตระหว่างเส้นทางที่จะถูกเลือกนั้นจะขึ้นอยู่กับอัลกอริทึม (Algorithm) ที่ใช้หรือที่เรียกว่า เราต์ติ้ง โปรโตคอล (“Routing Protocol”) ซึ่งจะแบ่งออกเป็น 2 ประเภท

2.9.1 Static IP Routing

สำหรับการจัดเส้นทางแบบนี้รายการในตารางเราต์ติ้งเทเบิลจะถูกป้อนโดยผู้ดูแลระบบ ซึ่งข้อมูลในรายการนี้จะไม่มีการเปลี่ยนแปลงหลังจากนั้น การคอนฟิกตารางนั้นจะค่อนข้างง่าย แต่ผู้ติดต่อบริการนั้นจะต้องป้อนข้อมูลทุก ๆ ฟิลด์ในตารางเองซึ่งจะเป็นเหมือนการบอกเราเตอร์ให้ทราบว่าเครือข่ายไหนที่สามารถติดต่อได้ ความถูกต้องของข้อมูลในตารางเราต์ติ้งเทเบิลจะขึ้นอยู่กับความรับผิดชอบของผู้ดูแลระบบนั้น ๆ โดยทั่วไปแล้ว สำหรับเครือข่ายเล็ก ๆ จะใช้เราต์ติ้งเทเบิลแบบสแตติกนี้ แต่เมื่อเครือข่ายขยายใหญ่ขึ้นก็จะใช้โปรโตคอลแบบไดนามิกซึ่งง่ายต่อการจัดการมากกว่า

2.9.2 Dynamic IP Routing

การจัดเส้นทางแบบไดนามิกนี้ก็จะใช้ชุด โพรโทคอลเพื่อใช้ในการสร้างตารางเร้าที่ดึงเทเบิลแทนการป้อนข้อมูลเองโดยคน ซึ่งวิธีการสร้างนั้นจะขึ้นอยู่กับโปรโตคอล เช่น โหลดของช่องสัญญาณแบนด์วิธของลิงค์ เป็นต้น ข้อได้เปรียบของการใช้โปรโตคอลแบบนี้คือ รายการในตารางจะถูกอัปเดตโดยอัตโนมัติทำให้ผู้ดูแลระบบไม่ต้องกังวลว่ารายการในตารางจะผิดพลาด ส่วนข้อเสียคือ ปริมาณการไหลเวียนของแพ็กเก็ตในเครือข่ายจะเพิ่มขึ้น โปรโตคอลแบบไดนามิกนี้ยังแบ่งย่อยออกเป็น 2 ประเภทคือ

2.9.2.1 Distance-Vector Routing Protocol

โปรโตคอลแบบคิสแทนส์เวกเตอร์จะเลือกเส้นทางที่ดีที่สุด โดยใช้เมตริก (Metric) เป็นเกณฑ์ โดยเมตริกนี้จะเป็นหน่วยที่วัดประสิทธิภาพของลิงค์ไปยังเครือข่ายนั้น และจะขึ้นอยู่กับโปรโตคอลที่ใช้ ซึ่งโดยส่วนใหญ่จะใช้จำนวนฮอป(Hop) เป็นหลัก เร้าเตอร์ที่ใช้โปรโตคอลนี้จะรักษาตารางเร้าที่ดึงเทเบิล โดยรายการตารางจะขึ้นอยู่กับสถานะของเครือข่ายนั้น ข้อเสียของโปรโตคอลนี้คือเร้าเตอร์จะต้องทำการแลกเปลี่ยนข้อมูลซึ่งกัน และกันเพื่ออัปเดตในรายการที่ต้องการในตารางหรือเพื่อให้ตารางเร้าที่ดึงเทเบิลของในแต่ละเร้าเตอร์ให้ทันสมัยอยู่ตลอดเวลา เร้าเตอร์แต่ละตัวต้องบรอดคาสต์ในช่วงเวลาที่กำหนดตลอดเวลา ดังนั้นจึงทำให้จำนวนแพ็กเก็ตที่ไหลเวียนในเครือข่าย

2.9.2.2 Link-State Routing Protocol

โปรโตคอลแบบลิงค์สเตท (Link State Routing Protocol) จะสร้างเส้นทางข้อมูลเหมือนกับต้นไม้ (Tree) โดยรากของต้นไม้ก็คือ เร้าเตอร์ตัวมันเอง โดยเร้าเตอร์ในแต่ละตัวจะทำการบรอดคาสต์ข้อมูลเกี่ยวกับเครือข่ายที่เชื่อมต่อตรงกับเร้าเตอร์เท่านั้น และเมตริกเร้าเตอร์จะทำการบรอดคาสต์เฉพาะตอนที่มีการเปลี่ยนแปลงเท่านั้น จึงทำให้ลดจำนวนแพ็กเก็ตในเครือข่ายลงได้

2.10 Domain Name System (DNS)

ในช่วงแรก ๆ ของการใช้เครือข่ายที่ใช้โปรโตคอล TCP/IP นั้น แต่ละโฮสต์จะใช้หมายเลขไอพีเป็นสิ่งที่บ่งชี้ตัวการที่ผู้ใช้อีกเครื่องหนึ่งจะติดต่อกับผู้ใช้อีกเครื่องหนึ่ง ผู้ใช้คนนั้นต้องรู้จักหมายเลขไอพีของอีกเครื่องหนึ่งเครือข่ายนี้ใช้งานได้ถ้าเครือข่ายไม่ใหญ่มากนัก เนื่องจากการเติบโตของอินเทอร์เน็ตเป็นไปอย่างรวดเร็ว ทำให้การจำหมายเลขไอพีของแต่ละโฮสต์เป็นไปได้ยาก ดังนั้นจึงได้มีการพัฒนาระบบเพื่อตั้งชื่อให้แต่ละโฮสต์แทนการใช้หมายเลขไอพี ซึ่งระบบนี้เรียกว่า “DNS (Domain Name System)” ระบบ DNS จะทำหน้าที่แปลงชื่อโฮสต์ให้เป็นหมายเลขไอพีหรือในทางตรงข้าม DNS ทำหน้าที่คล้ายกับสมุดโทรศัพท์ กล่าวคือเมื่อมีคนต้องการจะโทรศัพท์หาใคร คน

นั่นจะเปิดสมุดโทรศัพท์เพื่อดูเพื่อค้นหาหมายเลขโทรศัพท์ ของคนที่ต้องการติดต่อ คอมพิวเตอร์ก็เช่นกัน เมื่อต้องการติดต่อสื่อสารกับคอมพิวเตอร์เครื่องอื่น เครื่องนั้นก็จะทำการสอบถามหมายเลขไอพีของ เครื่องที่ต้องการสื่อสารด้วยกับ DNS แบ่งออกเป็น 3 ส่วนคือ

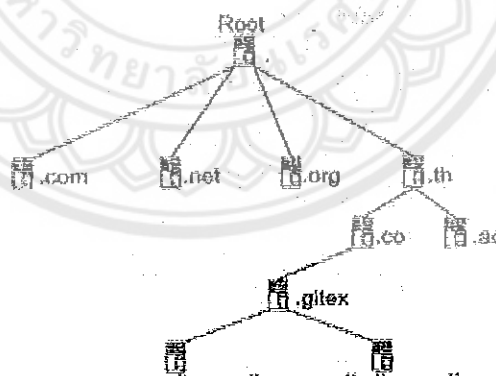
Name Resolvers : จุดประสงค์หลักของ DNS คือการแปลงชื่อคอมพิวเตอร์ให้เป็นหมายเลขไอพี ในเทอมของ DNS แล้วเครื่องไคลเอนท์ที่ต้องการสอบถามหมายเลขไอพีจะเรียกว่า “รีโซลฟเวอร์ (Resolvers)” ซอฟต์แวร์ที่ทำหน้าที่เป็นรีโซลฟเวอร์นั้นจะถูกสร้างขึ้นมากับแอปพลิเคชันหรืออาจจะ เป็นไลบรารีที่มีอยู่ในเครื่องไคลเอนท์

Domain Name Space : ฐานข้อมูลของระบบ DNS มีโครงสร้างเป็นต้นไม้ ซึ่งจะเรียกว่า “โดเมนเนมสเปซ (Domain Name Space)” แต่ละโดเมนจะมีชื่อและสามารถมีโดเมนย่อย หรือซับโดเมน (Subdomain) การเรียกชื่อจะใช้จุด (.) เป็นตัวแบ่งแยกระหว่าง โดเมนหลักและ โดเมนย่อย

Name Servers : เนมเซิร์ฟเวอร์ (Name Servers) คือเครื่องคอมพิวเตอร์ที่รันโปรแกรมที่จัดการ ฐานข้อมูลบางส่วนของระบบ DNS เนมเซิร์ฟเวอร์จะตอบกลับการร้องขอทันทีโดยการค้นหาข้อมูลใน ฐานข้อมูลตัวเอง หรือจะส่งต่อการร้องขอไปยังเนมเซิร์ฟเวอร์อื่น ถ้าเนมเซิร์ฟเวอร์มีเรCORDของส่วน ของโดเมน แสดงว่าเนมเซิร์ฟเวอร์นั้นเป็นเจ้าของโดเมนนั้น (Authoritative) ถ้าไม่มีก็จะเรียกว่า Non-Authoritative

2.10.1 Domain

โครงสร้างของระบบ DNS นั้นจะเป็นแบบมีลำดับชั้น (Hierarchy) ดังรูป



รูปที่ 2.13 โครงสร้างของระบบ DNS

Root Domain : ลำดับสูงสุดของระบบโดเมนคือ รุกโดเมน (Root Domain) ทุก ๆ โดเมนจะรู อยู่ภายใต้รุกโดเมนหมด ดังนั้นรุกโดเมนจึงเป็นส่วนที่สำคัญมากของระบบ DNS ในระบบ อินเทอร์เน็ตนั้นรุกโดเมนประกอบด้วยเซิร์ฟเวอร์ 7 เครื่อง

Top-Level Domain : ระดับโดเมนที่รองลงมาจากชื่อโดเมนจะเรียกว่า โดเมนระดับหนึ่ง (Top-Level Domain) โดเมนในระดับนี้จะถูกกำหนดให้โดยประเภทขององค์กรและประเทศ โดเมนในระดับนี้จะมีคนแบ่งออกเป็น 3 ส่วนคือ โดเมนขององค์กร โดเมนของประเทศ และโดเมนการแปลงกลับ

Second-Level Domain : สำหรับโดเมนระดับรองรองลงมาจากชื่อโดเมนนี้เป็นโดเมนที่แจกจ่ายให้กับองค์กรหรือบุคคลที่ต้องการชื่อโดเมน

2.10.2 โดเมนแบ่งตามหน้าที่ขององค์กร

โดเมนในระดับหนึ่งนี้จะอยู่ถัดจากชื่อโดเมน แต่ละโดเมนจะใช้โค้ดที่เป็นตัวอักษร 2-4 ตัวเพื่อบ่งบอกจุดประสงค์หรือหน้าที่หลักขององค์กรนั้น ๆ ตัวอย่างเช่น .COM เป็นโดเมนในระดับนี้

2.10.2.1 โดเมนของประเทศ

นอกจากในการตั้งชื่อโดเมนให้เหมาะกับประเภท หรือหน้าที่ขององค์กรแล้ว การตั้งชื่อโดเมนยังใช้ประเทศในการแบ่ง ซึ่งจะใช้ตัวอักษร 2 ตัวเป็นการบอกชื่อ โดเมนประเทศ

โดเมน	ประเทศ
.th	ไทย
.uk	อังกฤษ
.au	ออสเตรเลีย
.jp	ญี่ปุ่น
.kr	เกาหลี

ชื่อโดเมนยังสามารถใช้แบบผสมระหว่างทั้งสองประเภท ที่กล่าวข้างต้น โดยโดเมนที่บ่งบอกประเทศจะอยู่ขวาสุด และถัดมาจะเป็นตัวอักษร 2-3 ตัวของโดเมนที่บอกประเภทขององค์กร เช่น .co + .th จะได้โดเมนเป็น .co.th หมายความว่า เป็นโดเมนของบริษัทหนึ่งที่อยู่ในประเทศไทย

2.10.3 ประเภทของ DNS เซิร์ฟเวอร์

ข้อมูลที่เก็บไว้ใน DNS เซิร์ฟเวอร์แต่ละเครื่องจะแตกต่างกัน ขึ้นอยู่กับหน้าที่ หรือประเภทของ DNS เซิร์ฟเวอร์นั้น หน้าที่ของเซิร์ฟเวอร์นั้นจะเป็นสิ่งที่กำหนดว่าข้อมูลจะถูกเก็บไว้ในเซิร์ฟเวอร์อย่างไร โดย DNS เซิร์ฟเวอร์แบ่งเป็นประเภทต่าง ๆ ดังนี้

2.10.3.1 Primary Name Server

เนมเซิร์ฟเวอร์หลัก (Primary Name Server) คือ เซิร์ฟเวอร์ที่อ่านข้อมูลเกี่ยวกับคอนฟิกรูชันจากไฟล์ที่เก็บอยู่ในเครื่องนั้น การเปลี่ยนแปลงข้อมูลของโซน เช่น การเพิ่มเรCORDต่าง ๆ จะต้องทำที่เนมเซิร์ฟเวอร์หลักเท่านั้น

2.10.3.2 Secondary Name Server

เนมเซิร์ฟเวอร์รอง (Secondary Name Server) จะถ่ายโอนข้อมูลของโซนจากเนมเซิร์ฟเวอร์เครื่องอื่นซึ่งอาจเป็นเนมเซิร์ฟเวอร์หลัก หรือเนมเซิร์ฟเวอร์รองก็ได้

กระบวนการถ่ายโอนข้อมูลเกี่ยวกับโซนนี้จะเรียกว่า “โซนทรานสเฟอร์ (Zone Transfer)” การมีเครื่องเซิร์ฟเวอร์รองนั้นมีประโยชน์ดังนี้

Redundancy : แต่ละโซนจะต้องมีเนมเซิร์ฟเวอร์หลักหนึ่งเครื่อง และเซิร์ฟเวอร์รองหนึ่งเครื่อง เซิร์ฟเวอร์รองจะทำหน้าที่แทนเซิร์ฟเวอร์หลักเมื่อเซิร์ฟเวอร์หลัก

Distribution : เซิร์ฟเวอร์รองควรตั้งอยู่คนละที่กับเนมเซิร์ฟเวอร์หลัก หรือที่มีไคลเอนท์มากพอสมควร เพื่อเป็นการช่วยลดปริมาณแพ็กเก็ตที่ต้องวิ่งผ่านระบบ WAN เนื่องจากเนมเซิร์ฟเวอร์รองก็ทำหน้าที่เหมือนกับเนมเซิร์ฟเวอร์หลัก

Load Balancing : การใช้เนมเซิร์ฟเวอร์รองนั้นจะช่วยแบ่งเบาโหลดของเนมเซิร์ฟเวอร์หลักได้ ซึ่งจะช่วยให้เวลาในการ โพรเซสและตอบกลับ (Response Time) เร็วขึ้น

2.10.3.3 Master Name Server

มาสเตอร์เนมเซิร์ฟเวอร์ (Master Name Server) เป็นแหล่งข้อมูลโซนของเซิร์ฟเวอร์รอง ดังนั้นเมื่อมาสเตอร์เนมเซิร์ฟเวอร์อาจจะเป็นเนมเซิร์ฟเวอร์หลักก็ได้ หรือเนมเซิร์ฟเวอร์รองก็ได้ เมื่อเปิดเนมเซิร์ฟเวอร์รองครั้งแรกทำการติดต่อกับมาสเตอร์เนมเซิร์ฟเวอร์เพื่อทำโซนทรานสเฟอร์ สำหรับในแต่ละโซนที่เซิร์ฟเวอร์นี้จะทำหน้าที่เป็นเนมเซิร์ฟเวอร์รอง โซนทรานสเฟอร์จะเกิดขึ้นเป็นช่วงๆ หรือเมื่อใดก็ตามที่ข้อมูลเปลี่ยนแปลงบนมาสเตอร์เนมเซิร์ฟเวอร์

2.10.3.4 Forwarders and Slaves

เมื่อเนมเซิร์ฟเวอร์ได้รับการตอบถาม (Query) เข้ามา เครื่องนั้นก็จะทำการตรวจสอบข้อมูลเกี่ยวกับโซนนั้นในเซิร์ฟเวอร์นั้นก่อน แต่ถ้าเซิร์ฟเวอร์นั้นไม่มีข้อมูลอยู่ หรือไม่มีข้อมูลที่เป็นต้นฉบับ (Non-Authoritative) ของโซนนั้น มันก็จะทำการติดต่อกับเนมเซิร์ฟเวอร์เครื่องอื่น โดยส่วนใหญ่แล้วกรณีนี้จะเกิดขึ้นเมื่อโซนที่สอบถามมานั้น ไม่อยู่ในระบบเครือข่ายเดียวกัน DNS จะกำหนดให้เนมเซิร์ฟเวอร์เครื่องหนึ่งทำหน้าที่เป็นเครื่องส่งต่อ (Forwarder) เพื่อทำหน้าที่ร้องขอข้อมูลไปยังเนม

เซิร์ฟเวอร์อื่นที่อยู่บนอินเทอร์เน็ต และส่งผลที่ได้กลับไปยังเนมเซิร์ฟเวอร์ที่ทำการร้องขอมา ถ้าเครื่องฟอร์เวิร์ดเคอร์ไม่สามารถกลับไปร้องขอได้ เนมเซิร์ฟเวอร์ที่ร้องขอมาจะต้องตัดสินใจว่าจะตอบการร้องขออย่างไร การที่เนมเซิร์ฟเวอร์จะตอบการร้องขอเอง ในกรณีที่ฟอร์เวิร์ดเคอร์ไม่ทำงาน จะเรียกว่าเป็น“นอนเอ็กซ์คลูซีฟ โหมด(Nonexclusive Mode)”

2.10.3.5 Caching-only Name Server

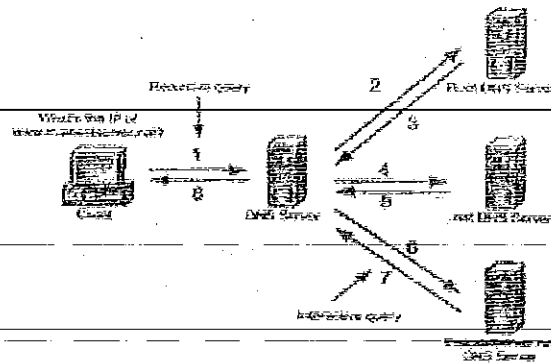
ดีเอ็นเอสเซิร์ฟเวอร์ จะทำการเก็บเรCORDของโชนที่ได้รับการตอบกลับแล้วไว้ในแคชเป็นเวลาช่วงหนึ่ง ซึ่งถ้าเซิร์ฟเวอร์นี้ไม่มีข้อมูลเกี่ยวกับโชนที่เก็บไว้ในรูปไฟล์ จะเรียกว่า “แคชซิงโอนลิ เนมเซิร์ฟเวอร์ (Caching-only Name Server)” ดังนั้นมันจึงไม่มีการทำโชนทรานสเฟอร์ ในตอนที่แคชซิงเซิร์ฟเวอร์ทำงาน ในเซิร์ฟเวอร์จะไม่มีข้อมูลใด ๆ เลยดังนั้นมันจึงทำการส่งต่อการร้องขอทั้งหมดไปยังเนมเซิร์ฟเวอร์เครื่องอื่น ในขณะที่เดียวกันก็ทำการเก็บข้อมูลการร้องขอต่าง ๆ ที่ได้รับการตอบกลับมาไว้ในแคช ครั้งต่อไปที่มีการร้องขอที่เหมือนกันมันก็สามารถตอบกลับได้ทันที ในตอนแรกนั้นแคชเซิร์ฟเวอร์จะรับส่งข้อมูลในปริมาณที่มาก เนื่องจากตอนแรกยังไม่มีข้อมูลอยู่ในแคชเลย แต่เนื่องจากแคชซิงเซิร์ฟเวอร์ไม่ต้องทำโชนทรานสเฟอร์ ดังนั้นปริมาณแพ็กเก็ตก็จะน้อยลงเมื่อถึงช่วงเวลาหนึ่ง

2.10.4 ชั้นตอนการทำงานของ DNS

กระบวนการในการร้องขอ (Query) ของระบบ DNS จะมีด้วยกัน 3 วิธีคือ รีเคอร์ซีฟ (Recursive) , อินเตอร์แอ็คทีฟ (Interactive) และอินเวอร์ส (Inverse)

2.10.4.1 การร้องขอแบบรีเคอร์ซีฟและอินเตอร์แอ็คทีฟ

โดยปกติการร้องขอแบบรีเคอร์ซีฟ (Recursive Query) จะเกิดขึ้นระหว่างไคลเอนท์และเนมเซิร์ฟเวอร์ การที่เนมเซิร์ฟเวอร์ได้รับการร้องขอแบบนี้จะตอบกลับด้วยข้อมูลที่เกี่ยวกับ โดเมนนั้น หรืออาจตอบกลับเป็นข้อความที่บอกการผิดพลาดถ้าข้อมูลของโดเมน หรือโฮสต์นั้นไม่มีระบบ เนมเซิร์ฟเวอร์ที่ได้รับการร้องขอแบบรีเคอร์ซีฟนี้ จะรับผิดชอบเกี่ยวกับการค้นหาข้อมูลของโดเมนหรือโฮสต์ โดยจะไม่สามารถส่งต่อเพื่อการร้องขอให้เนมเซิร์ฟเวอร์อื่นได้ อย่างไรก็ตามเนมเซิร์ฟเวอร์สามารถร้องขอแบบอินเตอร์แอ็คทีฟ (Interactive) กับเนมเซิร์ฟเวอร์อื่นได้ เพื่อค้นหาเนมเซิร์ฟเวอร์ที่มีสิทธิ์ในการจัดการ โดเมนดังกล่าว



รูปที่ 2.14 การร้องขอแบบไร้เคอร์ซีฟและอินเทอร์เน็ตพีซี

2.11 การออกแบบระบบเครือข่าย

ไม่ว่าจะเป็นเครือข่ายสำหรับองค์กรขนาดใดก็ตาม สิ่งที่จะต้องมามีคือ บุคลากรที่จะดูแลและจัดการระบบเครือข่าย สำหรับองค์กรขนาดเล็กอาจจะมีแค่หนึ่งหรือสองคนที่คอยดูแลและจัดการเรื่องต่างๆ ที่เกี่ยวกับคอมพิวเตอร์และระบบเครือข่าย ซึ่งหลายองค์กรมักจะจ้างเจ้าหน้าที่แบบชั่วคราวหรือทำงานแบบไม่เต็มเวลามาดูแลระบบ สำหรับการออกแบบและการติดตั้งระบบเครือข่ายในครั้งแรกนั้นหลายองค์กรอาจใช้วิธีการจ้างบริษัทที่ปรึกษาทางด้านเครือข่ายในครั้งแรกนั้นหลายองค์กรอาจใช้วิธีการจ้างบริษัทที่ปรึกษาทางด้านเครือข่ายทำการออกแบบและติดตั้งระบบให้ใช้งานได้ก่อน เมื่อติดตั้งเสร็จแล้วทางองค์กรที่รับผิดชอบทางด้านนี้เฉพาะ เพื่อทำหน้าที่เป็นผู้ประสานงานกับบริษัทที่จะมาออกแบบและติดตั้งระบบเครือข่ายให้กับองค์กรก่อน ในบทนี้จะเป็นการให้คำแนะนำสิ่งต่างๆ ที่ควรพิจารณาให้กับผู้ที่ได้รับมอบหมายจากองค์กรให้ทำหน้าที่นี้

2.11.1 การประเมินความต้องการ

ก่อนที่จะเริ่มลงมือออกแบบเครือข่ายผู้ออกแบบระบบต้องมีจุดประสงค์ที่ชัดเจนก่อน สิ่งหนึ่งที่สำคัญที่ต้องทำก่อนที่จะออกแบบระบบเครือข่าย การวิเคราะห์ระบบ ซึ่งจะรวมถึงการศึกษาระบบการทำงานขององค์กรแล้ววิเคราะห์ว่าสามารถใช้ระบบสารสนเทศเพื่อเพิ่ม ประสิทธิภาพในการทำงานขององค์กรได้อย่างไรขั้นต้นก็อาจลองพิจารณาว่ามีระบบงาน

อะไรบ้างถ้าใช้ เครือข่ายแล้วจะช่วยทำให้การปฏิบัติงานนั้นมีประสิทธิภาพมากขึ้น จะต้องใช้ซอฟต์แวร์ทางด้านธุรกิจอะไรบ้าง ผู้ใช้ต้องการที่จะแชร์ไฟล์ชนิดต่าง ๆ เท่านั้น หรือองค์กรต้องการที่จะมีระบบฐานข้อมูลที่มีผู้ใช้หลายคนหรือไม่ องค์กรต้องการที่จะใช้อีเมลหรือไม่ แล้วเว็บเซิร์ฟเวอร์จำเป็นไหม องค์กรต้องการนำธุรกิจเข้าสู่ระบบอีคอมเมิร์ซหรือไม่ ซึ่งถ้าใช่ นั่นก็หมายความว่าเครือข่ายขององค์กรต้องการที่จะเชื่อมต่อเข้ากับอินเทอร์เน็ต ที่กล่าวมาเป็นส่วนหนึ่งของหลาย ๆ สิ่ง ที่ ผู้ออกแบบระบบต้องศึกษาและค้นคว้าก่อนที่จะเริ่มลงมือออกแบบระบบ เมื่อได้ศึกษาและวิเคราะห์งานด้านธุรกิจที่จำเป็นต้องใช้เครือข่ายแล้วต่อไปให้ลำดับความสำคัญของแต่ละงาน เพื่อจะได้ วางแผนได้ว่าสิ่งไหนที่ต้องทำก่อนหรือสิ่งใดที่สามารถรอก่อนได้ ซึ่งผู้ออกแบบต้องให้ความสำคัญกับงานที่มีผลกระทบกับธุรกิจขององค์กรมากที่สุดเป็นอันดับแรก

ลักษณะการใช้งานคอมพิวเตอร์ในสำนักงานขนาดเล็กจะแตกต่างจากลักษณะการใช้งานคอมพิวเตอร์ในบริษัทขนาดใหญ่ หรือเอ็นเตอร์ไพรซ์ (Enterprise) ระบบเครือข่ายสำหรับองค์กรขนาดใหญ่ มักจะมีผู้ใช้หลายพันคน มีเครื่องเซิร์ฟเวอร์หลายสิบเครื่อง ซึ่งอาจจะมีเครื่องเมนเฟรมเชื่อมต่อเข้ากับระบบ และเครือข่ายอาจจะครอบคลุมบริเวณกว้างซึ่งอาจจะมีหลายอาคาร และแต่ละอาคารอาจมีหลายชั้น เครือข่ายประเภทนี้อาจจะต้องเชื่อมต่อกับเครือข่ายของสาขาย่อยหลายเครือข่ายของสาขาย่อยหลายเครือข่าย และในแต่ละสาขาย่อยอาจจะมีหลายอาคารซึ่งไม่เป็นการผิดปกติเลยที่เครือข่ายแบบเอ็นเตอร์ไพรซ์นี้จะประกอบด้วยอุปกรณ์เครือข่ายหลายพันเครื่องและต้องอาศัยเครือข่ายหลักหรือแบ็คโบน (Backbone) ในการเชื่อมต่อกันระหว่างเครือข่ายย่อย ๆ เข้าด้วยกัน เครือข่ายประเภทนี้มักจะมีอุปกรณ์ที่ซับซ้อนและราคาแพง ดังนั้นจึงจำเป็นต้องใช้บุคลากรที่มีความชำนาญสูงเฉพาะด้านนี้เฉพาะเพื่อดูแลและจัดการระบบ สำหรับองค์กรขนาดเล็กจะมีความต้องการทางด้านเครือข่ายที่กระทัดรัดกว่า เครือข่ายอาจจะประกอบด้วยเครื่องลูกข่ายไม่ถึงร้อยเครื่อง เครื่องพิมพ์เลเซอร์ไม่ถึงสิบเครื่อง โดยเครือข่ายนี้สามารถอำนวยความสะดวกแก่พนักงานขององค์กรในการแชร์ข้อมูลได้ด้วย รวมทั้งเครื่องพิมพ์และทรัพยากรเครือข่ายอื่น ๆ ร่วมกัน ความต้องการทางด้านเครือข่ายขององค์กรขนาดเล็กนี้สามารถที่จะตอบสนองด้วยระบบเครือข่ายเล็ก ๆ ซึ่งเครื่องเซิร์ฟเวอร์อาจหาซื้อได้ตามท้องตลาดทั่วไป เครือข่ายสำหรับองค์กรขนาดเล็กนี้ไม่จำเป็นต้องมีผู้ดูแลระบบที่มีความชำนาญและประสบการณ์มากนักไม่เหมือนเครือข่ายแบบเอ็นเตอร์ไพรซ์ ถึงแม้ว่าเครือข่ายขององค์กรขนาดเล็กจะมีความซับซ้อนไม่มากนัก อย่างไรก็ตามยังมีบางประเด็นที่เหมือนกัน ตัวอย่างเช่น เครือข่ายทั้งสองประเภทต้องถูกออกแบบให้สามารถตอบสนองความต้องการของผู้ใช้ได้อย่างมีประสิทธิภาพ มีการควบคุมความปลอดภัยของข้อมูลสามารถที่จะขยายได้ง่ายในอนาคตหรือเมื่อธุรกิจขยายตัวใหญ่ขึ้นก็สามารถที่จะขยายเครือข่ายได้ง่าย และ ไม่ให้การวางแผนเครือข่ายในตอนแรกมีผลบังคับต่อการเลือกใช้เทคโนโลยีใหม่ ๆ ที่เกิดขึ้นหลังจากการติดตั้งแล้วซึ่งอาจจะมีผลกระทบต่อธุรกิจขององค์กรได้ สำหรับองค์กรขนาดใหญ่อาจมีสำนักงานย่อยที่ตั้งอยู่ในที่ห่างไกลจากสำนักงานใหญ่ สำนักงานสาขาย่อยเหล่านี้มักจะ

ได้รับการสนับสนุนทางด้านเทคโนโลยีสารสนเทศจากสำนักงานใหญ่ได้ไม่เต็มที่ ดังนั้นระบบเครือข่ายสำหรับสำนักงานขนาดเล็กนี้ควรจะมีการออกแบบให้อยู่ในลักษณะที่ไม่ซับซ้อนจนเกินไป และง่ายต่อการจัดการและดูแล เนื่องจากแต่ละองค์กรมีความต้องการที่จะใช้เครือข่ายในรูปแบบที่ต่างกัน ดังนั้นจึงเป็นการยากที่จะออกแบบระบบเครือข่ายหนึ่งแล้วใช้งานได้อย่างเหมาะสมกับทุก ๆ องค์กรได้ เพราะเหตุนี้ผู้เขียนจะแนะนำหลักการทั่ว ๆ ไปที่ใช้ในการออกแบบระบบเครือข่าย ผู้เขียนจะยึดหลักการง่าย ๆ เพื่อให้บรรลุวัตถุประสงค์ขององค์กร การใช้งานเครือข่ายนั้นส่วนใหญ่จะไม่จำกัดเพียงเฉพาะในองค์กรเท่านั้น ตอนนี้เราควรจะต้องทราบแล้วว่า องค์กรต้องการที่จะเชื่อมต่อกับภายนอกอย่างไรบ้าง สิ่งหนึ่งที่จะปฏิเสธไม่ได้ในยุคปัจจุบันนี้ก็คือ การเชื่อมต่อเข้ากับอินเทอร์เน็ต แต่สิ่งต่อไปที่ต้องพิจารณาก็คือ การเชื่อมต่อด้วยโมเด็มจะเพียงพอกับความต้องการหรือไม่ หรือว่าองค์กรต้องการการเชื่อมต่อแบบตลอดเวลา แล้วต้องการแบนด์วิธเท่าไร องค์กรต้องการที่จะเชื่อมต่อเข้ากับเครือข่ายส่วนบุคคลอื่น เช่น เครือข่ายที่ใช้ที่บ้านหรือไม่ สิ่งที่ยากอย่างหนึ่งสำหรับการออกแบบเครือข่ายคือ การเชื่อมต่อเข้ากับเครือข่ายอื่นไม่เฉพาะความยุ่งยากทางด้านเทคนิคแต่เกี่ยวข้องกับราคาที่แพงด้วย หลังจากได้ศึกษาและวิเคราะห์ระบบงานขององค์กรแล้ว ผู้ออกแบบระบบก็สามารถสรุปได้ว่าระบบเครือข่ายควรจะมีขนาดเท่าไร ซึ่งประมาณได้โดยการพิจารณาจากจำนวนพนักงานที่ต้องใช้คอมพิวเตอร์ในการทำงาน และต้องการใช้มากน้อยเท่าใด จากนั้นค่อยออกแบบเครือข่ายให้สามารถรองรับการใช้งานของผู้ใช้ทุกคนและเผื่อไว้สำหรับการขยายตัวของเครือข่ายในอนาคต ผู้ออกแบบระบบอาจคำนวณความต้องการที่ต้องใช้เครือข่ายขององค์กรในช่วง 2-3 ปีข้างหน้าโดยการพิจารณาจำนวนผู้ใช้ใหม่ที่将会เพิ่มขึ้น และอาจคำนึงถึงความต้องการเพิ่มทางด้านพื้นที่เก็บข้อมูลหรือขนาดของฮาร์ดดิสก์ต่อผู้ใช้ ซึ่งอาจจะเพิ่มขึ้นอย่างคาดไม่ถึงก็ได้ เครือข่ายที่ออกแบบจะต้องสามารถขยายได้ง่ายโดยการเพิ่มเครื่องเซิร์ฟเวอร์หรือเครื่องลูกข่าย หรือการใช้เทคโนโลยี ใหม่ ๆ กับระบบได้โดยง่าย

2.11.2 การเลือกประเภทของเครือข่าย

ในเครือข่ายใดเครือข่ายหนึ่งคอมพิวเตอร์มักจะทำหน้าที่ไม่เป็นเซิร์ฟเวอร์ก็เป็นไคลเอนท์ เครื่องไคลเอนท์คือ เครื่องที่ใช้บริการต่าง ๆ ที่เครื่องเซิร์ฟเวอร์มีให้ ซึ่งบทบาทนี้จะไม่ตามตัวเสมอไปนั่นคือ บางเครื่องอาจจะเป็นเครื่องไคลเอนท์ในการทำงานลักษณะหนึ่ง หรืออาจมีบทบาทเป็นเครื่องเซิร์ฟเวอร์เมื่อทำงานอีกอย่างหนึ่ง เครือข่ายแบบไคลเอนท์เซิร์ฟเวอร์นี้ต้องมีเครื่องคอมพิวเตอร์เฉพาะที่ทำหน้าที่ให้บริการอย่างเดียว การทำงานแบบนี้จะแยกเครื่องเซิร์ฟเวอร์ออกจากเครื่องไคลเอนท์อย่างเด็ดขาด สำหรับเครื่องเซิร์ฟเวอร์นั้นต้องเลือกใช้คอมพิวเตอร์ที่มีประสิทธิภาพสูง และใช้เกี่ยวกับการให้บริการด้านนั้น ๆ โดยเฉพาะ ส่วนไคลเอนท์อาจเป็นคอมพิวเตอร์ที่มีประสิทธิภาพธรรมดาหรือไม่สูงมากนัก เนื่องจากไคลเอนท์จะอาศัยเซิร์ฟเวอร์คือความสะดวกในการบริหารและจัดการเครือข่ายในเรื่องต่าง ๆ เช่น ทรัพยากรเครือข่าย การรักษาความปลอดภัยและยังทำให้การวางแผนให้ระบบมีประสิทธิภาพเพียงพอสำหรับการใช้งานของ

องค์กรได้ เครื่องข่ายแบบเพียร์ทูเพียร์ (Peer-to-Peer) นั้น คอมพิวเตอร์แต่ละเครื่องอาจทำหน้าที่เป็นทั้งไคลเอนท์และเซิร์ฟเวอร์ ซึ่งเครือข่ายประเภทนี้ไม่จำเป็นต้องมีเครื่องเซิร์ฟเวอร์ที่ทำงานเฉพาะในการให้บริการในเครือข่ายระบบปฏิบัติการที่ใช้ในปัจจุบัน เช่น วินโดวส์ Me/XP, MacOS และ Linux ส่วนใหญ่จะมีพีซีที่สามารถให้บริการในรูปแบบต่าง ๆ พร้อมทั้งเชื่อถืออำนาจให้ทำงานทั่ว ๆ ไปได้เช่นกัน เมื่อใช้ระบบปฏิบัติการเหล่านี้แล้วเราสามารถที่จะสร้างเครือข่ายแบบเพียร์ทูเพียร์ได้เลย ซึ่งคอมพิวเตอร์เครื่องหนึ่งสามารถทำงานได้ทั้งแบบที่เป็นเครื่องใช้งานโดยทั่วไปและยังสามารถให้บริการในเครือข่ายในรูปแบบต่าง ๆ ได้ เช่น การแชร์ฮาร์ดดิสก์ เครื่องพิมพ์ หรือทรัพยากรอื่น ๆ ตัวอย่างเช่น ถ้าหากว่าคอมพิวเตอร์เครื่องหนึ่งมีเครื่องพิมพ์เลเซอร์เราสามารถที่จะแชร์เครื่องพิมพ์นี้ให้กับผู้ใช้คนอื่น ๆ ผ่านเครือข่ายได้ หรือเราต้องการแชร์ไฟล์เตอร์โคโพลเดอร์หนึ่งหรือฮาร์ดดิสก์ก็สามารถทำได้เช่นกัน และระบบปฏิบัติการบางระบบสามารถที่จะใช้เป็นเว็บเซิร์ฟเวอร์ได้เช่นกัน เครือข่ายแบบเพียร์ทูเพียร์จะมีราคาถูกกว่าเพราะไม่จำเป็นต้องมีเครื่องเซิร์ฟเวอร์ที่มีประสิทธิภาพสูงและราคาแพง แต่เมื่อเครือข่ายต้องขยายใหญ่ขึ้นจนถึงจุด ๆ หนึ่ง ก็จะทำให้การบริหารและการจัดการระบบเครือข่ายยากขึ้น ซึ่งบางทีอาจทำให้ระบบเครือข่ายมีประสิทธิภาพต่ำลงได้ และอาจทำให้เครือข่ายไม่อาจจะรองรับการทำงานตามที่องค์กรต้องการก็ได้ ถ้าหากว่าทรัพยากรเครือข่ายที่แชร์กันใช้อยู่ในเครื่องศูนย์กลางแล้วก็จะทำให้ผู้ใช้ค้นหาทรัพยากรเหล่านี้ได้ง่าย และ ผู้จัดการระบบก็สามารถที่จะกำหนดสิทธิ์ต่างๆ ของผู้ใช้ที่จะเข้ามาใช้ทรัพยากรเหล่านี้ได้ง่าย เช่นกัน ในทางตรงกันข้ามถ้าหากว่าทรัพยากรเหล่านี้กระจายอยู่ตามเครื่องผู้ใช้ทั่วไปใน เครือข่ายก็จะทำให้การค้นหาข้อมูลเหล่านี้เป็นไปด้วยความยากลำบากและซับซ้อนมาก ประสิทธิภาพก็อาจจะเป็นปัญหาอย่างหนึ่งของเครือข่ายแบบเพียร์ทูเพียร์ กล่าวคือเราต้องมั่นใจว่าเครื่องคอมพิวเตอร์ในเครือข่ายประเภทนี้มีประสิทธิภาพพอที่จะทำงานได้ ทั้งงานของผู้ใช้เครื่องและงานที่ต้องให้บริการทางเครือข่ายด้วย เช่น เราสามารถที่จะใช้คอมพิวเตอร์เครื่องหนึ่งเป็นทั้งเว็บเซิร์ฟเวอร์และในขณะเดียวกันก็ทำงานบนแอปพลิเคชันเกี่ยวกับระบบบัญชีขององค์กร เว็บเซิร์ฟเวอร์อาจจะทำงานช้าในขณะที่ผู้ใช้ทำงานเกี่ยวกับการคำนวณภาษีรายได้ขององค์กรก็ได้ ซึ่งงานทั้งสองประเภทนี้จะทำในคอมพิวเตอร์เครื่องเดียวกันก็ต่อเมื่อโหลดของทั้งสองงานไม่มากนัก ในโลกของความเป็นจริงแล้วเครือข่ายขององค์กรโดยทั่วไปก็จะเป็นแบบผสมระหว่างเครือข่ายสองประเภทนี้ ตัวอย่างเช่น เครือข่ายขององค์กรใหญ่ ๆ อาจจะมีคอมพิวเตอร์ที่ทำหน้าที่เป็นเฉพาะคาค้าเบสเซิร์ฟเวอร์ขององค์กร ในขณะที่เดียวกันอาจมีคอมพิวเตอร์ของผู้ใช้เครื่องหนึ่งแชร์ไฟล์เตอร์ให้ผู้ใช้คนอื่นเข้ามาใช้ก็ได้ แต่การแชร์ดังกล่าวนี้ อาจจะไม่เป็นการแชร์แบบทางการ ซึ่งคณะผู้ทำงานทางด้านไอทีขององค์กรอาจจะไม่รับรองข้อมูลดังกล่าวก็ได้ ในการพิจารณาเบื้องต้นของการสร้างเครือข่ายสำหรับธุรกิจขนาดเล็กก็ขอให้พิจารณาความเป็นไปได้ และความเหมาะสมของเครือข่ายทั้งสองประเภทนี้ เครือข่ายแบบเพียร์ทูเพียร์จะเหมาะสมสำหรับเครือข่ายขนาดเล็ก ๆ เท่านั้น ถ้าหากการใช้เครือข่ายมีความสำคัญอย่างมากต่อ

ทำธุรกิจขององค์กรแล้วก็ควรที่จะมีเครื่องเซิร์ฟเวอร์ที่ทำงานทางด้านนั้น โดยเฉพาะ เพื่อประกันประสิทธิภาพ และความเชื่อถือได้ของการใช้งานระบบ แต่ถ้าเครือข่ายที่จะสร้างเริ่มต้น โดยมีเครื่องประมาณ 10-20 เครื่อง เครือข่ายแบบเพียร์ทูเพียร์ก็น่าจะเพียงพอ แต่ก็ให้วางแผนล่วงหน้าเพื่อที่อาจจะต้องเพิ่มเครื่องเซิร์ฟเวอร์เมื่อต้องการที่จะขยายเครือข่าย

2.12 การเขียนโปรแกรมเน็ตเวิร์กโดยใช้ วินซ็อก (Winsock)

การเขียนโปรแกรมผ่านเน็ตเวิร์กกระหว่างฝั่งไคลเอนต์กับฝั่งเซิร์ฟเวอร์ด้วยการใช้โปรโตคอล ทีซีพี/ไอพี ใน Windows Socket API หรือ Winsocks;

2.12.1 ความหมายของซ็อกเก็ต (socket)

ซ็อกเก็ตถูกนิยามหรือกำหนดไว้ว่าเป็นคู่ของการสื่อสารหรือคู่ของ โปรเซส(หรือ เซรด) โดยที่การสื่อสารบนเน็ตเวิร์กใช้คู่ของซ็อกเก็ตสำหรับแต่ละ โปรเซสสำหรับซ็อกเก็ตประกอบไปด้วย ไอพีแอดเดรสกับหมายเลข Port (Port Number)

โดยทั่วไป ซ็อกเก็ตใช้สถาปัตยกรรมไคลเอนต์เซิร์ฟเวอร์ เซิร์ฟเวอร์จะรอการเข้ามาตามการร้องขอของไคลเอนต์โดยการฟังที่พอร์ตเฉพาะเมื่อการร้องขอได้รับเซิร์ฟเวอร์ก็จะยอมรับการเชื่อมต่อจากซ็อกเก็ตไคลเอนต์เพื่อให้สมบูรณ์ในการเชื่อมต่อเซิร์ฟเวอร์ที่สร้างการบริการเฉพาะ เช่น เทลเน็ต, เอฟทีพี, เมล์ และ เอชทีทีพี จะฟังที่พอร์ตมีชื่อ เช่น เซิร์ฟเวอร์เทลเน็ตจะฟังที่พอร์ต 23, เซิร์ฟเวอร์เอฟทีพีจะฟังที่พอร์ต 21 หรือ เซิร์ฟเวอร์เอชทีทีพีจะฟังที่พอร์ต 80 เป็นต้น หมายเลขพอร์ตทั้งหมดที่ต่ำกว่า 1024 จะถูกพิจารณาว่าเป็นพอร์ตมาตรฐานที่เราสามารถใช้พอร์ตเหล่านี้เพื่อการสร้างบริการตามมาตรฐานได้

2.12.2 ตัวอย่างการสื่อสารด้วยซ็อกเก็ต

เมื่อเซรดฝั่งไคลเอนต์เริ่มต้นการขอร้องสำหรับการเชื่อมต่อจะถูกกำหนดพอร์ตโดยโฮสต์คอมพิวเตอร์(Host computer) พอร์ตนี้เป็นหมายเลขใดๆ ก็ได้ที่มากกว่า 1024 ตัวอย่าง เช่น ถ้าไคลเอนต์บนโฮสต์ A มีไอพีแอดเดรส 2 ต้องการที่จะสร้างการเชื่อมต่อกับเซิร์ฟเวอร์ เอชทีทีพี (ซึ่งฟังที่พอร์ต 80)ที่มีไอพีแอดเดรส 1 โฮสต์ A จะถูกกำหนดพอร์ต 2544 และที่ฝั่งเซิร์ฟเวอร์จะเป็นพอร์ต 80-สถานการณ์นั้นสามารถแสดงได้ดังรูป-เพื่อก็คิดไปมาระหว่างโฮสต์ทั้งสองจะถูกส่งไปยังเซรดที่เหมาะสม ซึ่งขึ้นอยู่กับหมายเลขพอร์ตปลายทางด้วย

การเชื่อมต่อทั้งหมดเป็นคุณสมบัติเฉพาะดังนั้นถ้าโปรเซสอื่นๆบนโฮสต์ A ต้องการสร้างการเชื่อมต่ออื่นๆกับเซิร์ฟเวอร์เอชทีทีพีเดียวกัน เซิร์ฟเวอร์จะกำหนดหมายเลขพอร์ตที่มากกว่า 1024 และต้องไม่เท่ากับพอร์ต 2544 (เนื่องจากถูกใช้ไปแล้ว) การทำอย่างนี้เพื่อให้แน่ใจว่าการเชื่อมต่อทั้งหมดประกอบด้วยคู่ที่เป็นยูนิค (unique) ของซ็อกเก็ตหรือเป็นสิ่งที่ไม่ซ้ำกับการเชื่อมต่ออื่นๆของซ็อกเก็ต

โดยปกติแล้วเซิร์ฟเวอร์จะมีหลาย ๆ การร้องขอที่พร้อมกันจะต้องใช้ระยะเวลาหนึ่งที่ไกลเกินที่ต้องรอคอยเพื่อที่จะถูกบริการโดยเซิร์ฟเวอร์เซตเดียว ซึ่งจะไม่สามารถรับได้เพื่อแก้ไขสถานการณ์นี้ เซิร์ฟเวอร์ต้องจัดการการขอร้องที่พร้อม ๆ กัน โดยการกำหนดแรมคแยกออกมาเพื่อบริการแต่ละการร้องขอที่เข้ามา ตัวอย่างเช่น เซิร์ฟเวอร์ เอชทีทีพี ที่ไม่ว่าจะกำหนดแรมคแยกออกมาเพื่อบริการแต่ละการร้องขอสำหรับเว็บเพจ

2.12.3 ชนิดของซ็อกเก็ต

ชนิดของซ็อกเก็ตมีอยู่สามชนิด คือ

- Connection-Oriented Socket
- Connectionless Socket
- Raw Socket

2.12.3.1 Connection-Oriented Socket

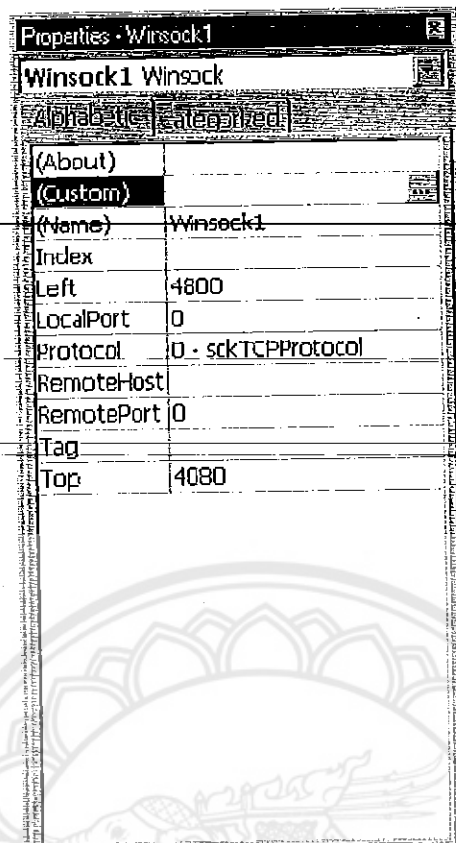
เป็นซ็อกเก็ตสำหรับการเชื่อมต่อแบบต่อเนื่องที่จะยอมอนุญาตให้โพรเซสนั้นๆสามารถเชื่อมต่อกับโพรเซสจากระยะไกล (Remote) ซึ่งใช้โพรโตคอล ทีซีพี (Transmission Control Protocol) ดังนั้นด้วยวิธีการนี้ทำให้ข้อมูลมีความเชื่อถือได้เมื่อการเชื่อมต่อได้เกิดขึ้น โพรเซสก็จะมี การส่งข้อมูลกลับไปจนกระทั่งฝั่งใดฝั่งหนึ่งหรือฝั่งอื่นๆปิดการเชื่อมต่อชนิดของซ็อกเก็ตแบบนี้ บางครั้งเรียกว่า สตรีมซ็อกเก็ต (Stream socket) ทั้งเอฟทีพีและ เอชทีทีพีใช้ซ็อกเก็ตแบบนี้ในการเชื่อมต่อของการสื่อสาร

2.12.3.2 Connectionless Socket

ซ็อกเก็ตแบบนี้บางที่จะเรียกว่า เดทาแกรมซ็อกเก็ต ซึ่งซ็อกเก็ตแบบนี้เป็นการเชื่อมต่อแบบไม่ต่อเนื่อง และนำมาใช้ประโยชน์ในการส่งเมสเซจสั้นๆซึ่งไม่สนับสนุนส่วนหัว ดังนั้นจึงพิจารณา การเชื่อมต่อประเภทนี้เป็นแบบเชื่อถือไม่ได้ ซึ่งก็คือ การไม่รับประกันข้อมูลที่ส่งออกไป ไม่เหมือนกับซ็อกเก็ตการเชื่อมต่อแบบต่อเนื่องที่ซ็อกเก็ตปลายทางถูกตรวจสอบเมื่อแพ็กเก็ตถูกส่งออกไป ซ็อกเก็ตแบบไม่ต่อเนื่อง เปรียบเสมือนกับการบริการของไปรษณีย์ที่ผู้ส่งจดหมายไปตามที่อยู่แล้วใส่ในกล่องรับจดหมาย ผู้ส่งจะไม่ทราบว่าผู้รับได้รับจดหมายหรือไม่ ซ็อกเก็ตแบบนี้ นิยมใช้กันในเซิร์ฟเวอร์ DNS (Domain Name System) ที่ใช้ซ็อกเก็ตเดทาแกรมในการตอบสนองต่อการร้องขอที่เข้ามาหลายๆ นอกจากนั้นจะใช้เดทาแกรมซ็อกเก็ตในการกระจาย (Broadcast) เมสเซจ หรือ Multicast เพื่อไปยังปลายทางหลายๆแห่งพร้อมกัน ซึ่งเหมือนกับการกระจายเสียงวิทยุ หรือ โทรทัศน์

2.12.3.3 Raw Socket

เป็นซ็อกเก็ตที่อนุญาตให้การเข้าถึง โพรโตคอล Transport Raw Socket ยังสามารถนำมาใช้เพื่อจัดการข้อมูลของ "ไอพีเฮคเตอร์" นอกจากนี้แล้วการใช้ซ็อกเก็ตชนิดนี้ยังต้องการความรู้อย่างมากเกี่ยวกับโครงสร้างโพรโตคอลพื้นฐาน



2.13 วินซ็อก (Winsock)

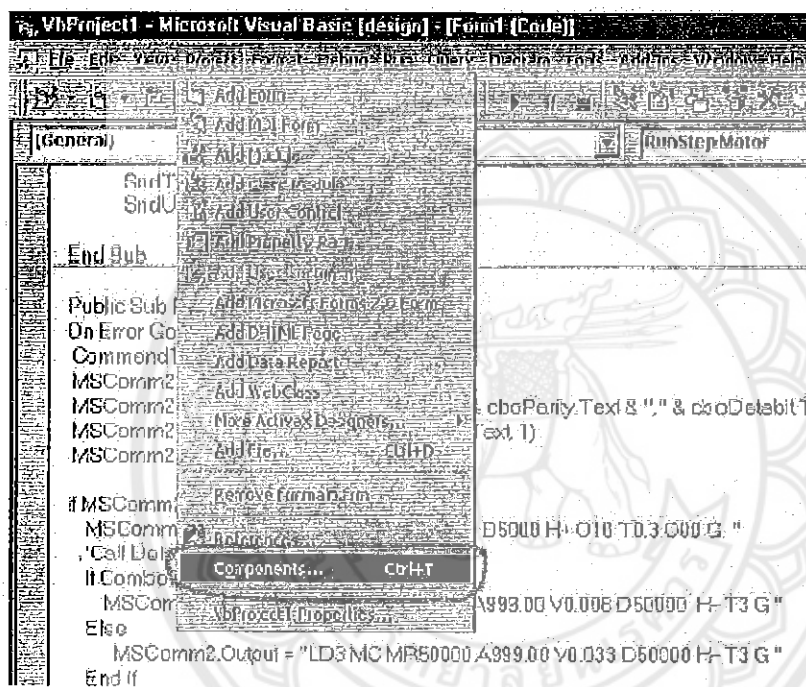
วินซ็อก เป็นมาตรฐานเปิดเน็ตเวิร์ก เอพีไอ โดยที่วินซ็อกถูกออกแบบมาครั้งแรกเพื่อการสร้างโปรแกรมอินเทอร์เน็ตเฟสที่เป็นมาตรฐานสำหรับที่ซีพี/ไอพี ในทุกเวอร์ชันของของระบบปฏิบัติการวินโดวส์รวมทั้ง วินโดวส์ 2000, วินโดวส์ เอ็นที, วินโดวส์ 98 ซึ่งจะเป็นวินซ็อกเวอร์ชัน 2.2 แต่ถ้าเป็นระบบปฏิบัติการวินโดวส์รุ่นดั้งเดิม เช่น วินโดวส์ 95, วินโดวส์ ซีอี นั้นจะใช้วินซ็อกเวอร์ชัน 1.1 มีสองเหตุผลหลักในการใช้วินซ็อก คือ การคอนโทรลและความมีประสิทธิภาพ

วินซ็อก เป็นเน็ตเวิร์กแอปพลิเคชันโปรแกรมมิ่งอินเทอร์เน็ตเฟสไม่ใช่โพรโตคอล ซึ่งวินซ็อกนั้นมีรากฐานเดียวกับซ็อกเก็ตตระกูล บีเอสดี (Berkeley Software Distribution) เวอร์ชัน 4.3 จากมหาวิทยาลัยแคลิฟอร์เนีย วิทยาเขต เบร์keley รายละเอียดได้รวมเอาทั้งรูทีนซ็อกเก็ตสไคล์ บีเอสซี (BSC) และการขยายสเปกมาใช้กับวินโดวส์การใช้วินซ็อกอนุญาตให้แอปพลิเคชันสามารถทำการติดต่อสื่อสารข้ามเน็ตเวิร์กใดๆก็ได้ที่กระทำกับ วินซ็อก เอพีไอ แพลตฟอร์ม วิน 32 ซึ่งวินซ็อกได้ให้เซรคที่ปลอดภัย (thread safety)

2.13.1 MS Winsock Control 6

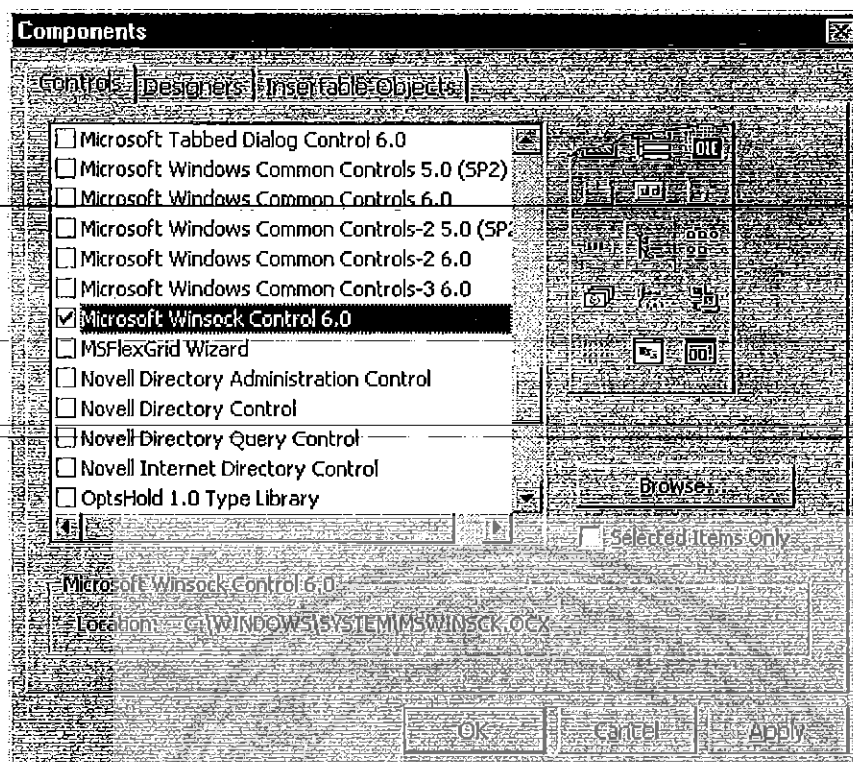
เขียนโปรแกรมบนระบบ Windows ให้สามารถติดต่อสื่อสารบนระบบเครือข่ายได้ ถ้าเป็นเมื่อก่อนบนระบบปฏิบัติการ DOS นั้นคงจะยากทีเดียว แต่เดี๋ยวนี้อุปกรณ์คอมพิวเตอร์อะไรก็

Windows กันแล้วเนื่องจากมีเครื่องมือ (Tool) ที่ช่วยให้เราสามารถเขียนโปรแกรมติดต่อผ่านระบบเครือข่าย ที่จะกล่าวถึงก็คือ MS Winsock Control 6 เป็นเครื่องมือที่อำนวยความสะดวกสำหรับการเขียนโปรแกรมติดต่อผ่านระบบเครือข่าย โดยโปรโตคอล TCP/IP ซึ่งเราจะทดลองเขียนด้วย Visual Basic หรือถ้าถนัด C/C++ ก็ใช้ Visual C++ ถนัด ปาสคาล ก็ใช้ Delphi นอกเหนือจาก Winsock Control แล้วยังมี API ที่มีนักพัฒนาเขียนขึ้นมาหลายตัวแต่ Winsock จะใช้งานง่ายกว่า ตัวอย่างการเขียนโปรแกรมด้วย Visual Basic ขึ้นแรกเปิดโปรแกรม VB ขึ้นมาแล้วสร้าง Project Standard EXE ขึ้นใหม่ จากนั้นเพิ่ม Winsock Control ก่อน โดยคลิกที่เมนู AddWinsockControl



แสดงการเพิ่ม components

ให้เลือก Control ชื่อ Microsoft Winsock Control 6.0 (สำหรับ VB6)



แสดง Properties Winsock Dialog

Properties	Detail
Name	ชื่อของคอนโทรลที่เราสามารถตั้งได้
LocalPort	Ports ที่เราจะกำหนดในการติดต่อ
Protocal	เลือกชนิดของโพรโตคอล ปกติเราใช้ 0-TCPProtocol
RemoteHost	IP Address ของเครื่องที่ใช้ ติดต่อ
RemotePort	Port Address ของเครื่องที่ใช้ ติดต่อ

Winsock Procedure ในส่วนของ Windows Code Object --> Winsock มี Procedure สำหรับ
กระทำติดต่อสื่อสารทั้งทางฝ่าย Server และ Client

-Close คือ เหตุการณ์เมื่อมีหยุดหรือยกเลิกการติดต่อสื่อสารของฝ่าย Server หรือ Client โดย Function Winsock.Close ซึ่งเราจะสามารถใช้ตรวจสอบฝ่ายตรงข้ามว่ามีการติดต่ออยู่หรือไม่โดยอาจจะใส่ Message เตือนเป็นต้น

-Connect เป็นเหตุการณ์ที่ฝ่าย Client มีการส่งสัญญาณติดต่อกับมายัง Server ส่งผลให้ Procedure นี้ของฝ่าย Server ก็เลยทำงานขึ้นมา เหมือนเดิมครับเราสามารถนำ Code Message ไปใส่เพื่อตรวจสอบได้เช่นกัน

-Connection Request เป็นเหตุการณ์เมื่อฝ่าย Client ส่งสัญญาณติดต่อกับมายัง Server Procedure ส่วนนี้ก็จะทำงานพร้อมกับค่า request ID As Long ซึ่งเป็นหมายที่ Gen ขึ้นมาในระบบ ค่านั้นจะไม่เหมือนเดิม โดยจะให้ฝ่าย Server รับรู้ว่าใช้ ID จากคอนโทรลตัวใดเพื่อจะได้สื่อสารถึง

-Data Arrival เหตุการณ์นี้เกิดขึ้นเมื่อมีการส่งข้อมูลระหว่าง Server และ Client Procedure นี้ก็จะทำงานขึ้นมา พร้อมกับค่าจำนวน bytesTotal As Long ที่รับเข้ามา

-Error เหตุการณ์ที่เกิดความผิดพลาดระหว่างการติดต่อสื่อสารระหว่าง Server และ Client โดยจะส่งค่า Number As Integer มาให้ว่าเป็นหมายเลขใดพร้อมทั้งรายละเอียดของการผิดพลาดในเหตุการณ์นั้นๆ คือ Description As String

-Send Progress จะเกิดขึ้นในขณะที่มีการส่งข้อมูลอยู่เหตุการณ์นี้ก็จะทำงานเมื่อส่งข้อมูลหมดแล้วก็จะส่งผลทำให้เกิด Event Send Complete

-Send Complete เหตุการณ์เมื่อมีการส่งข้อมูลออกไปยังฝ่ายตรงข้ามเสร็จเรียบร้อยแล้ว

2.13.2. Winsock Properties & Events

Accept (request ID) คือการตกลงกันระหว่าง Server และ Client ในการเลือกหมายเลข ID Control ให้ตรงกันเพื่อสามารถสื่อสารได้ถูกต้อง

Close เป็นการส่งสัญญาณยกเลิกการติดต่อกัน จะเป็นฝ่าย Server หรือ Client ก็ได้ที่จะใช้ Function นี้ จากนั้นจะทำให้ Procedure close ในฝ่ายตรงข้ามทำงาน

Connect เป็นการส่งสัญญาณว่าตอนนี้ทำการติดต่อเรียบร้อยแล้ว ซึ่งจะส่งผลให้ Procedure ฝ่ายตรงข้ามทำงาน

Get data เป็นการรับข้อมูลเมื่อฝ่ายตรงข้ามส่งมาโดยประโยคคำสั่งนี้จะอยู่ในส่วนของ Procedure

Data Arrival เนื่องจากเป็นเหตุการณ์ที่การกระทำขณะเมื่อฝ่ายตรงข้ามส่ง ข้อมูลเข้ามา

Listen การกระทำที่จะคอยตรวจสอบสัญญาณที่ส่งไปว่าฝ่ายตรงข้ามตอบรับการร้องขอ การติดต่อ

Local Host Name คำสั่งนี้จะส่งชื่อของ Computer name ของเครื่องนั้นๆ

Debug. Print Winsock1. Local Host Name

Local IP คำสั่งนี้จะทำการส่งหมายเลข IP Address

Debug. Print Winsock1.Local IP

Local Port คำสั่งที่จะส่งค่าของหมายเลขในการติดต่อ TCP/IP ของเครื่องนั้นๆ

Debug. Print Winsock1.LocalPort

RemoteHost กำหนดหรือคืนค่าชื่อ Computer nameของเครื่องที่จะทำการติดต่อ
Winsock1.RemoteHost = MyServer

RemoteHostIP กำหนดหมายเลข IP Address ของเครื่องที่จะทำการติดต่อ Winsock1.

RemoteHostIP=10.10.0.0

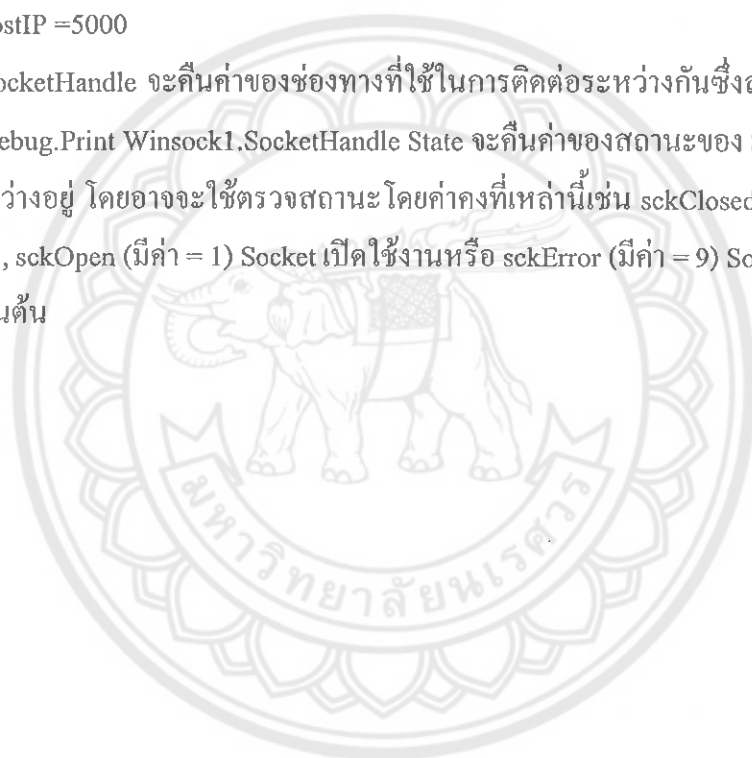
RemoteHostPort กำหนดหมายเลข Port ที่จะใช้ในการติดต่อระหว่างกัน Winsock1.

RemoteHostIP =5000

SocketHandle จะคืนค่าของช่องทางที่ใช้ในการติดต่อระหว่างกันซึ่งสามารถเรียกดูได้ดังนี้

Debug.Print Winsock1.SocketHandle State จะคืนค่าของสถานะของ Socket ขณะที่ใช้

ติดต่อระหว่างอยู่ โดยอาจจะใช้ตรวจสอบสถานะโดยค่าคงที่เหล่านี้เช่น sckClosed (มีค่า=0) Socket ปิด
การใช้งาน, sckOpen (มีค่า = 1) Socket เปิดใช้งานหรือ sckError (มีค่า = 9) Socket มีความผิดพลาด
เกิดขึ้น เป็นต้น

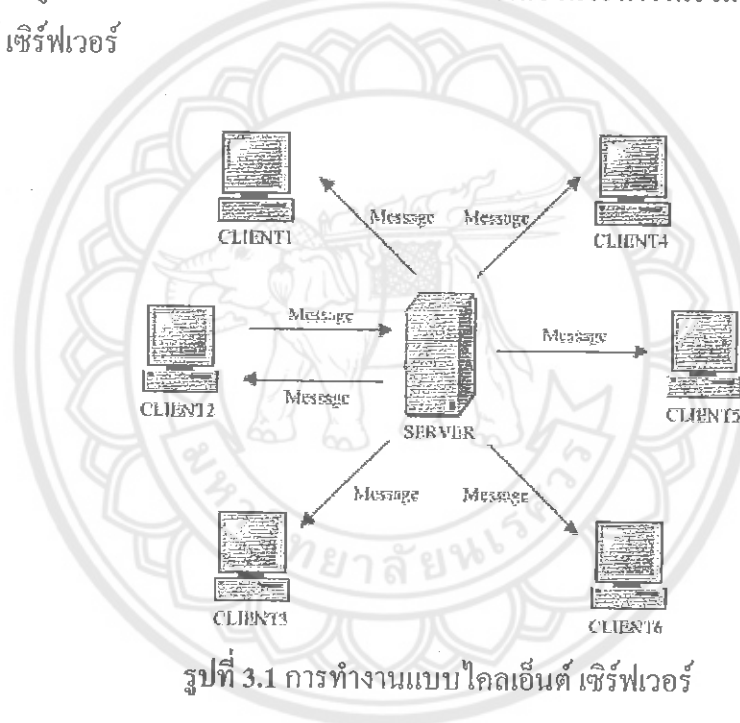


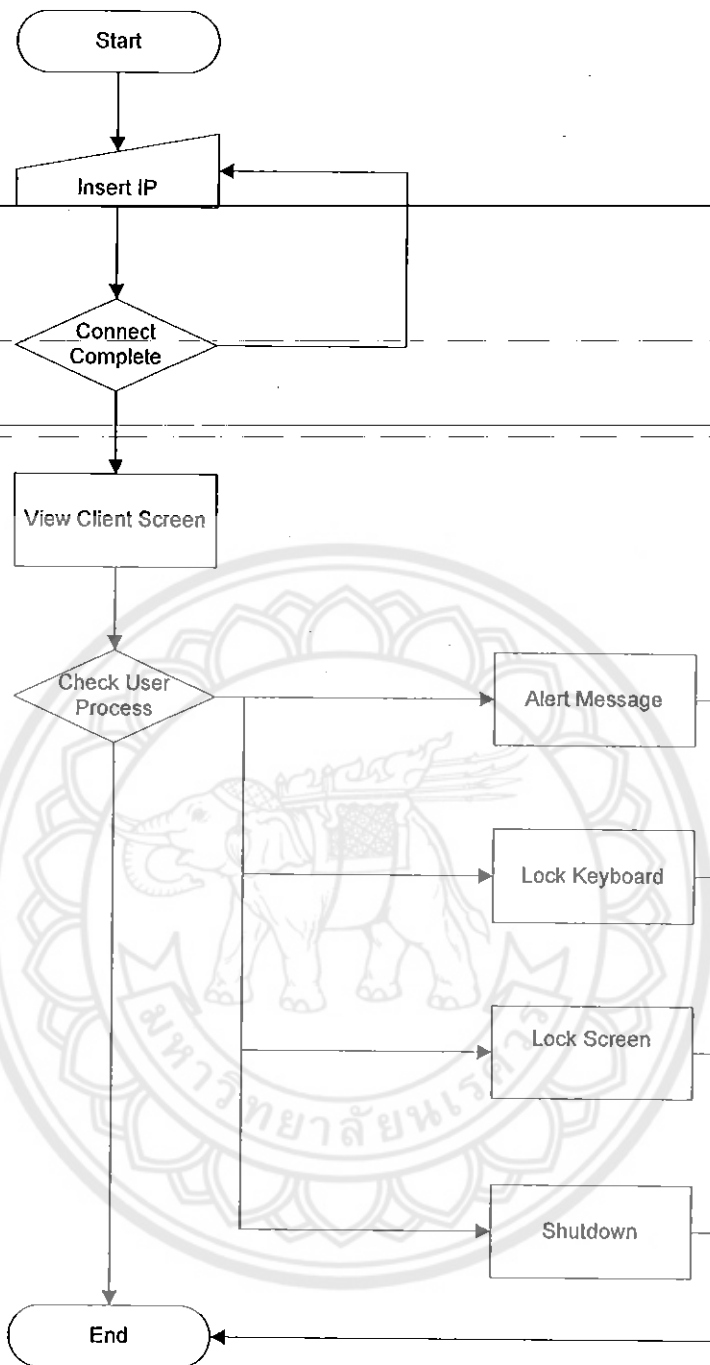
บทที่ 3

วิธีการดำเนินงานโครงการวิศวกรรม

3.1 การออกแบบซอฟต์แวร์

ซอฟต์แวร์ตัวนี้ประกอบด้วย 2 ส่วนสำหรับใช้ติดตั้ง คือ ส่วนที่เป็นตัวขอบริการในที่นี้จะเรียกว่า เครื่องมาสเตอร์ (Master) หรือเครื่องแม่ข่ายเป็นเครื่องที่ทำหน้าที่ควบคุมหรือใช้เฝ้าดูการทำงานของเครื่องเป้าหมายซึ่งในเครื่องเป้าหมายจะต้องมีโปรแกรมที่ทำหน้าที่ส่งข้อมูลที่มีการเปลี่ยนแปลง หรือส่งคำสั่งมาให้กับตัวมาสเตอร์นี้ และฝั่งให้บริการจะเรียกว่าเครื่องลูกข่าย (Client) ที่จะทำการส่งข้อมูลที่มาสเตอร์ต้องการ ตัวโปรแกรมทั้งสองนี้จะทำงานร่วมกัน โดยทำงานเป็นแบบไคลเอ็นต์ เซิร์ฟเวอร์





รูปที่ 3.2 โฟลชาร์ต

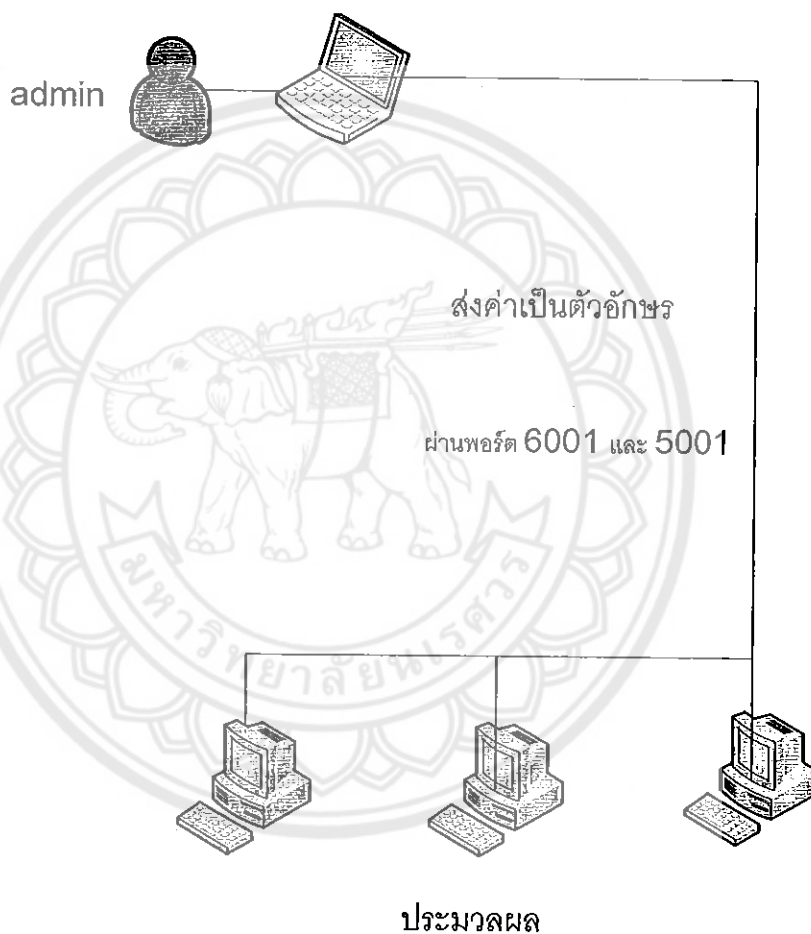
3.2 สถานการณ์ทำงานของโปรแกรม

ชุดโปรแกรมนี้จะประกอบด้วยส่วนประกอบย่อยที่ประกอบเราด้วยกันเพื่อให้ชุดโปรแกรมนี้มีความสามารถที่จะตอบสนองความจำเป็นพื้นฐานในการจัดการกับเครื่องปลายทางที่ต้องการ แต่ส่วนหลักของชุดโปรแกรมนี้จะอยู่ที่ส่วนของตัวที่ทำการเข้าควบคุมการทำงานของเครื่องลูกข่าย โดยจะมีลักษณะ โดยรวมดังนี้

- Shutdown, restart, stand by

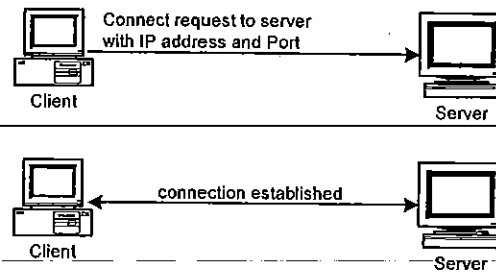
- ส่งข้อความ และ ตัวอักษร ไปให้เครื่องลูกข่ายได้
- สามารถเปิดปิดเสียง ล็อก ซีดีรอม และเข้าโปรแกรมต่างๆได้
- สามารถ ล็อก หน้าจอได้
- จะไม่เห็นใน แอป สตาร์ท เมนู ด้านล่าง
- เมื่อฝังตัว จะไม่เจอใน task manager
- ในขณะที่โปรแกรมทำงาน จะไม่รบกวนและทำให้ผู้ใช้เครื่องคอมพิวเตอร์ทราบ

ส่งคำสั่งที่ต้องการจะทำ



รูปที่ 3.3 ลักษณะการทำงานของโปรแกรม

โดยโปรแกรมนี้จะใช้การทำงานโดยการใช้ วินซอร์ค ก็สามารถส่งข้อมูลในเครือข่าย TCP/IP ได้โดย ใช้หมายเลขพอร์ต (port) ที่ทำการเชื่อมต่อ โดยในการติดต่อจะอยู่ในรูปแบบของ ไคลเอนต์/เซิร์ฟเวอร์ ซึ่งเครื่องเซิร์ฟเวอร์จะเป็นผู้รับการติดต่อ และเครื่องไคลเอนต์จะเป็นผู้ที่ทำการติดต่อไปยังเซิร์ฟเวอร์ ด้วยหมายเลข IP และพอร์ตของเซิร์ฟเวอร์หลังจากเกิดการเชื่อมต่อขึ้นแล้ว เครื่องไคลเอนต์และเซิร์ฟเวอร์จะสามารถติดต่อสื่อสารข้อมูลกันได้



รูปที่ 3.4 การคอนเนกต์ ของ IP

3.2.1 รายละเอียดต่างๆเกี่ยวกับการใช้ วินซอร์ค คอนโทรลเขียนโปรแกรม

ในการใช้วินซอร์ค ติดต่อกับโปรโตคอล TCP/IP เพื่อให้เกิดการติดต่อกันระหว่าง ไคลเอ็นต์ เซิร์ฟเวอร์นั้นจะต้องมีทั้งฝั่ง เซิร์ฟเวอร์ และ ไคลเอ็นต์ ซึ่งคล้ายว่าจะมี 2 โปรแกรม แต่ความจริงแล้วก็คือโปรแกรมเดียวกันแต่เปิด 2 หน้าต่างโดยกำหนดว่าฝั่งไหนเป็น เซิร์ฟเวอร์ ฝั่งไหนเป็น ไคลเอ็นต์ แล้วทำรูปร่างให้สามารถใช้งานได้ เหมือนกับความต้องการที่จะใช้เหตุการณ์แรกที่ต้องทำในฐานะที่เป็น ฝ่าย เซิร์ฟเวอร์ คือการตรวจสอบสัญญาณจากฝั่ง ไคลเอ็นต์ ในที่นี้ใช้ อีเวนต์

Click ของปุ่มชื่อ cmdListen

Private Sub cmdListen_Click()

Winsock1.LocalPort = txtPortSvr.Text

Winsock1.Listen

End Sub

เหตุการณ์ที่สองในฐานะไคลเอ็นต์ ก็จะตอบรับ โดยส่งสัญญาณไปให้กับ เซิร์ฟเวอร์

Private Sub cmdConnect_Click()

Winsock1.RemoteHost = "xx.xx.xx.xx" (ถ้าติดต่อเครื่องตัวเองใช้ 127.0.0.1)

Winsock1.RemotePort = "5001" ' กำหนดหมายเลขอื่นก็ได้ แต่ต้องให้ตรงกันทั้ง 2 ฝ่าย

Winsock1.Connect

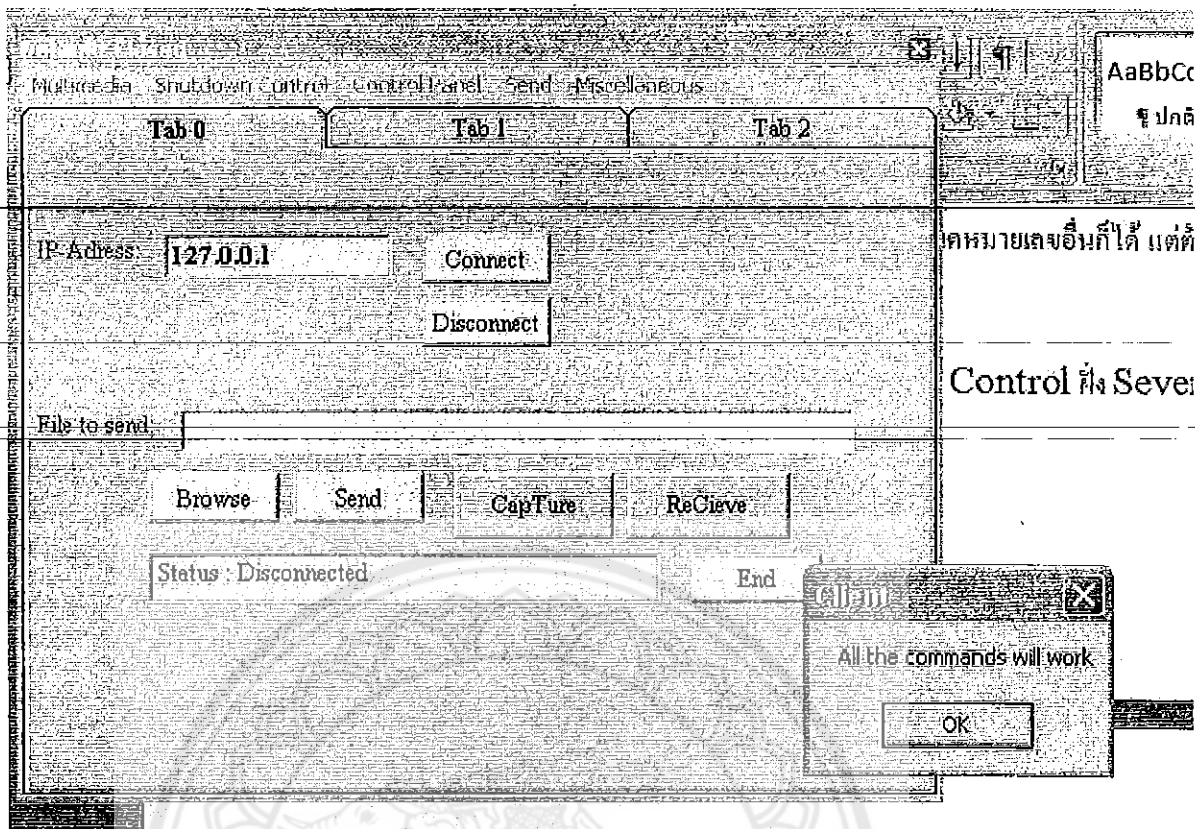
End Sub

เมื่อคลิกปุ่ม cmdConnect แล้วทาง Winsock Control ฝั่ง เซิร์ฟเวอร์ จะเกิด Event Connect ซึ่งใน Event นี้ก็ให้ใส่ Message เตือนขึ้น เช่น

Private Sub Winsock1_Connect()

MsgBox "Connected", vbExclamation, "All command Work"

End Sub



รูปที่ 3.5 สถานะเมื่อคอนเนกต์แล้ว

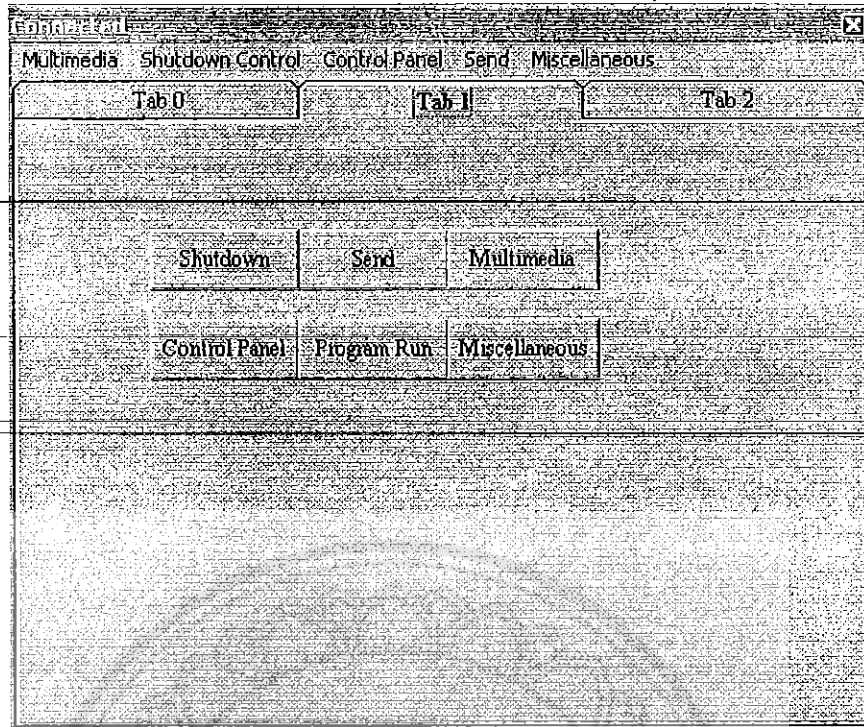
3.3 การควบคุมเครื่องลูกข่ายผ่านกราฟิกโหมด

3.3.1 หลักการ

ตัวโปรแกรมส่วนนี้ถือได้ว่ามีความสำคัญมาก เนื่องจากว่าส่วนนี้จะอนุญาตให้สามารถทำการควบคุมเครื่องลูกข่ายได้อย่างง่ายดายคล้ายเสมือนทำงานอยู่หน้าจอของเครื่องของเราเอง ทำให้เราสามารถทำการติดตามพฤติกรรมการใช้งานของผู้ใช้งานได้เพื่อจะหามาตรการที่จะควบคุมการใช้งานของผู้ใช้งานในขั้นต่อไป หรือผู้ดูแลอาจเข้าทำการควบคุมเครื่องลูกข่ายได้ในกรณีที่มีการใช้งานเครื่องผิดวัตถุประสงค์

3.3.2 การทำงาน

การทำงานของโปรแกรมตัวนี้จะใช้หลักการในการส่งข้อมูลเป็น text ไปแล้วให้ประมวลผลในปลายทาง กล่าวคือ จะให้เครื่องทาง เซิร์ฟเวอร์ ส่งข้อมูลที่เป็นตัวอักษรที่กำหนดไว้แล้วไปให้ทางฝั่ง ไคลเอ็นต์ ดำเนินการประมวลผลเพื่อกระตามคำสั่งที่ทาง เซิร์ฟเวอร์ ต้องการได้ ยกตัวอย่างเช่น กำหนดให้เครื่อง เซิร์ฟเวอร์ ส่งคำสั่งไปว่า ชัคดาวน์ ก็ให้เครื่องทางฝั่ง ไคลเอ็นต์ทำการ ชัคดาวน์ ตัวเองเป็นต้น



รูปที่ 3.6 แถบคำสั่ง



รูปที่ 3.7 การแสดงผลหน้าจอของเครื่องลูก

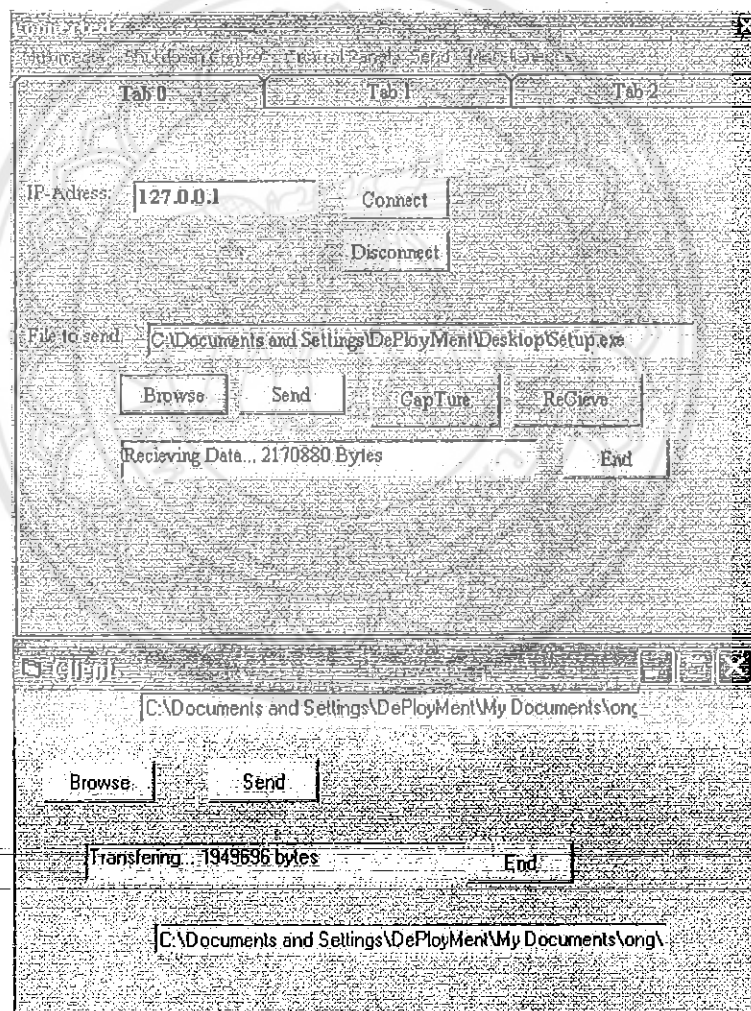
3.4 โปรแกรมโอนย้ายข้อมูล

3.4.1 หลักการ

โปรแกรมส่วนนี้จะใช้ในการโอนย้ายข้อมูลระหว่างเครื่องภายใต้การดูแลของผู้ใช้งาน เครื่องแม่ข่ายเพื่อความสะดวกของผู้ใช้งานในการย้ายข้อมูลไปยังเครื่องต่างๆ เพื่อความรวดเร็วและสะดวกในการดูแลระบบทำให้การจัดการระบบง่ายยิ่งขึ้น

3.4.2 การทำงาน

การทำงานของส่วนนี้ก็คล้ายๆกับการทำงานของโปรแกรมสนทนาข้างต้นคือ จะทำการรอคอยการเชื่อมต่อเข้ามาอยู่แล้วเมื่อมีการยอมรับการเชื่อมต่อเรียบร้อยแล้วก็จะสามารถรับส่งไฟล์ได้ เช่นเดียวกับการส่งข้อความหรือข้อมูลประเภทต่างๆ



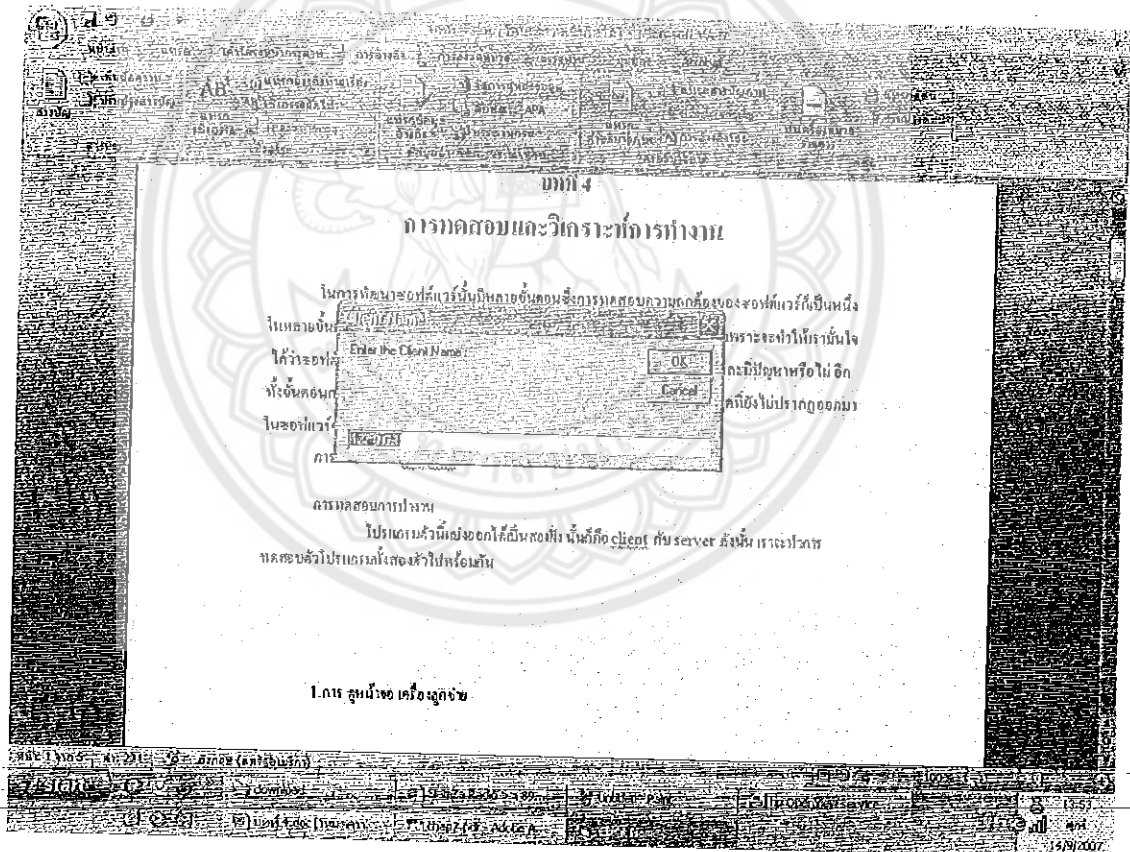
รูปที่ 3.8 การเลือกส่งไฟล์ข้อมูล

บทที่ 4

การทดสอบและวิเคราะห์การทำงาน

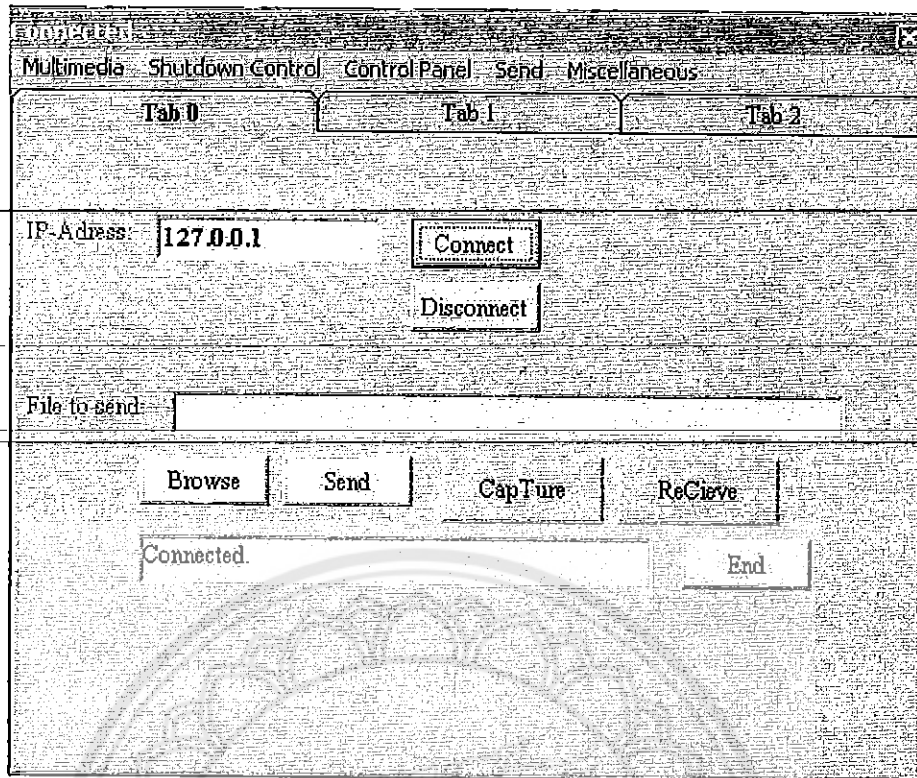
ในการพัฒนาซอฟต์แวร์นั้นมีหลายขั้นตอนซึ่งการทดสอบความถูกต้องของซอฟต์แวร์ก็เป็นหนึ่งในหลายขั้นตอนของการพัฒนาซอฟต์แวร์ ซึ่งนับได้ว่าเป็นมีความสำคัญมากอย่างหนึ่งเพราะจะทำให้เรามั่นใจได้ว่าซอฟต์แวร์ที่เราทำการพัฒนาขึ้นมาสามารถนำไปใช้งานได้ถูกต้องและมีปัญหาหรือไม่ อีกทั้งขั้นตอนการทดสอบความถูกต้องของซอฟต์แวร์นี้จะทำให้เราสามารถค้นพบข้อผิดพลาดที่ยังไม่ปรากฏออกมาในซอฟต์แวร์ของเรา ซึ่งทำให้เราแก้ไขได้ก่อนที่จะมีการนำไปใช้งานได้จริง

การทดสอบการคอนเนกต์ (connect) ของเครื่อง client-server



รูปที่ 4.1 การระบุ ip เครื่องลูกข่าย

ขั้นแรกให้ใส่ ip ของเครื่องลูกข่ายในช่อง



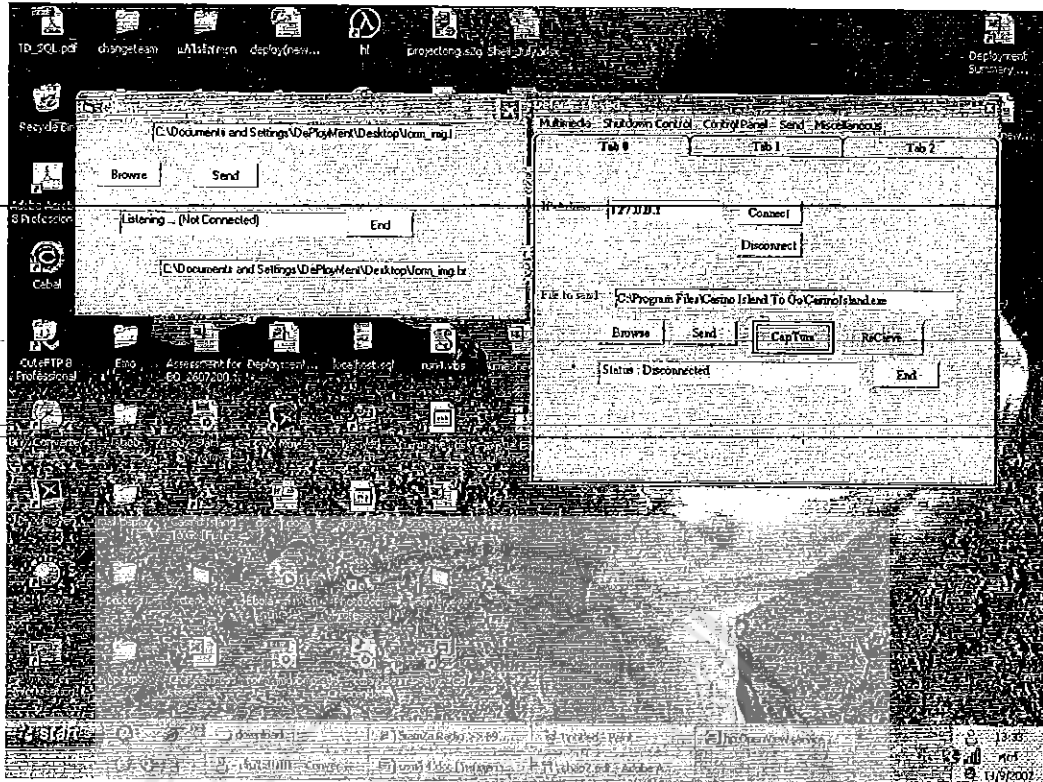
รูปที่ 4.2 หน้าจอแสดงสถานะว่า connect

เมื่อทำการ connect แล้ว จะได้น้ำจอดังรูป จะมี status ด้านล่างว่า “connect”

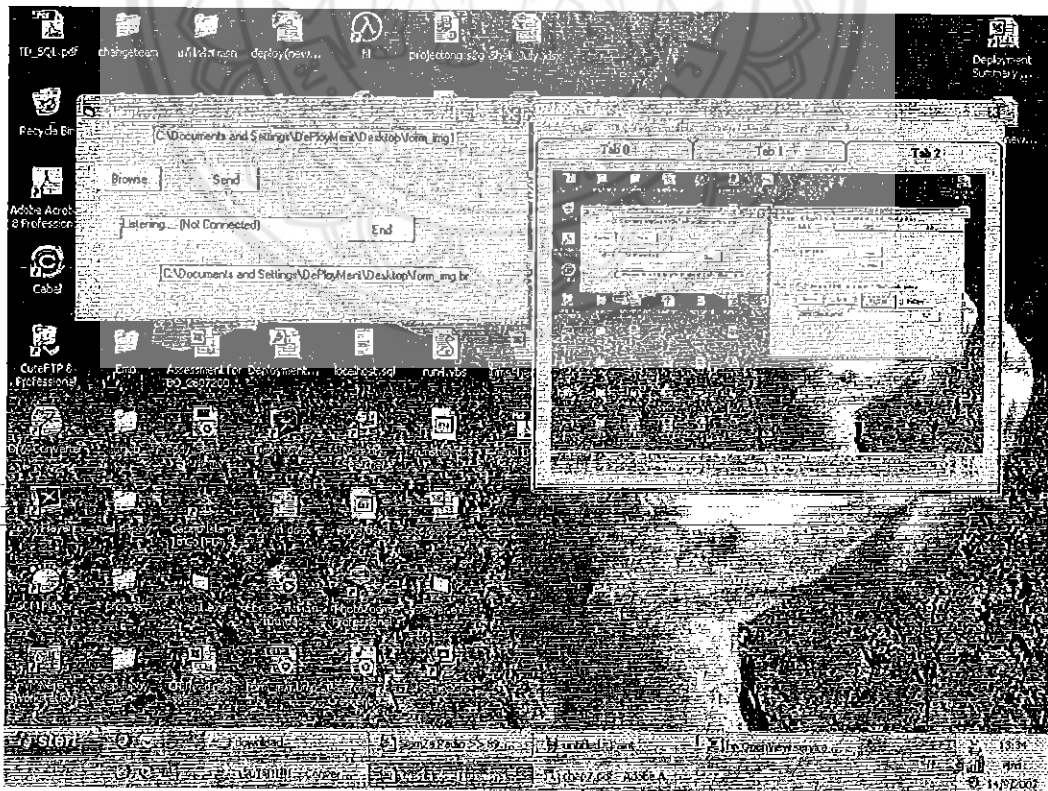
4.1 การทดสอบการทำงาน

โปรแกรมตัวนี้แบ่งออกได้เป็นสองฝั่ง นั่นก็คือ client กับ server ดังนั้น เราจะทำการทดสอบตัว โปรแกรมทั้งสองตัวไปพร้อมกัน

4.1.1 การดูหน้าจอเครื่องลูกข่าย คือ การที่เราสามารถมองเห็นหน้าจอของเครื่องลูกข่ายได้ว่ากำลังทำอะไรอยู่ เพื่อที่จะได้พิจารณาต่อไปว่า ใช้คอมพิวเตอร์ไปในทางที่ถูกต้องหรือไม่



รูปที่ 4.3 ก่อนเช็คหน้าจอ

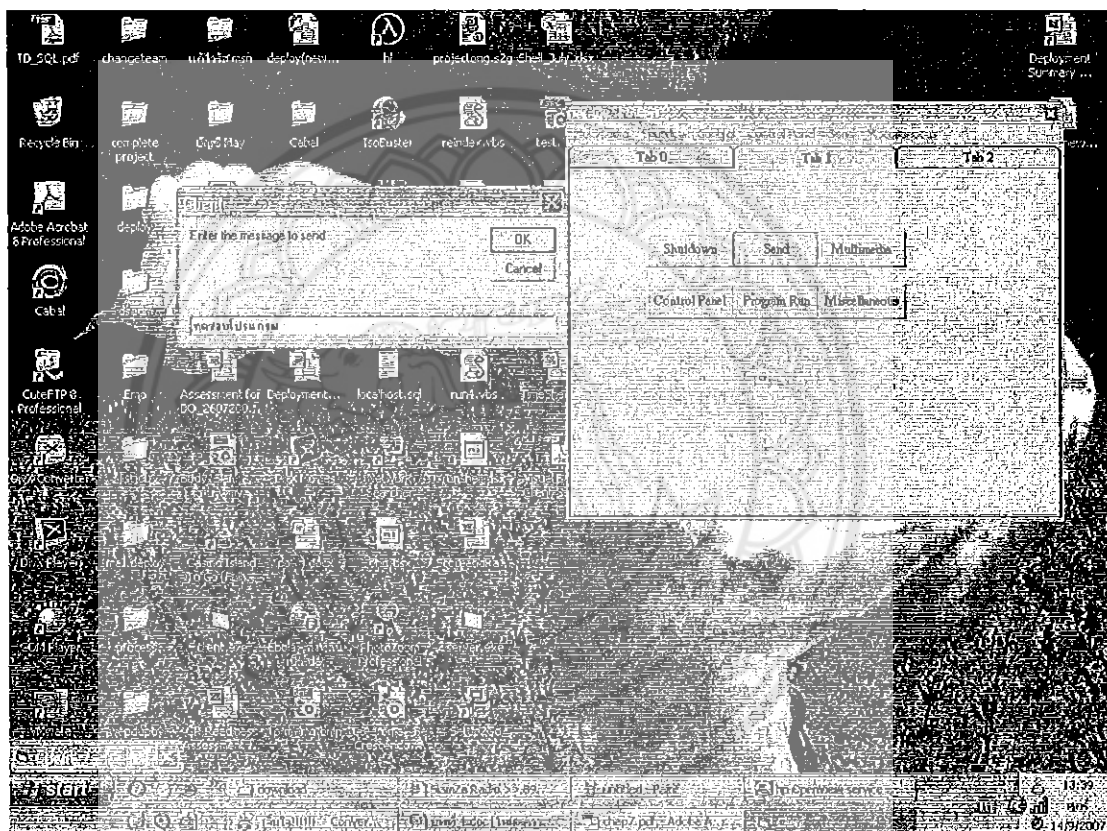


รูปที่ 4.4 หลังเช็คหน้าจอ

4.1.2 ปัญหาที่พบ คือ มีการส่งไฟล์หน้าจอ ที่ซ้ำมาก เพราะว่าไฟล์รูปภาพหน้าจอจะมีขนาดใหญ่เป็นเหตุผลมาจากไฟล์ที่เกิดขึ้นจะเป็นไฟล์.bmpซึ่งมีขนาดใหญ่ และ ในโปรแกรม vb เองก็สามารถรองรับได้เพียง นามสกุลเดียว

4.1.3 การสื่อสารไปยังเครื่อง client

เราสามารถสื่อสารกับเครื่องลูกข่ายได้ทั้งสองทางคือ แบบ chat และ แบบส่งคำพูดไป ซึ่งในที่นี้เราจะใช้การสื่อสารแบบส่งคำพูดไป หรือ ก็คือการส่งคำพูดเดือนไปยังเครื่องลูกข่าย ซึ่งเป็นการบังคับให้ต้องรับฟังเพราะว่าไม่ต้องการคำพูดตอบโต้กลับมา



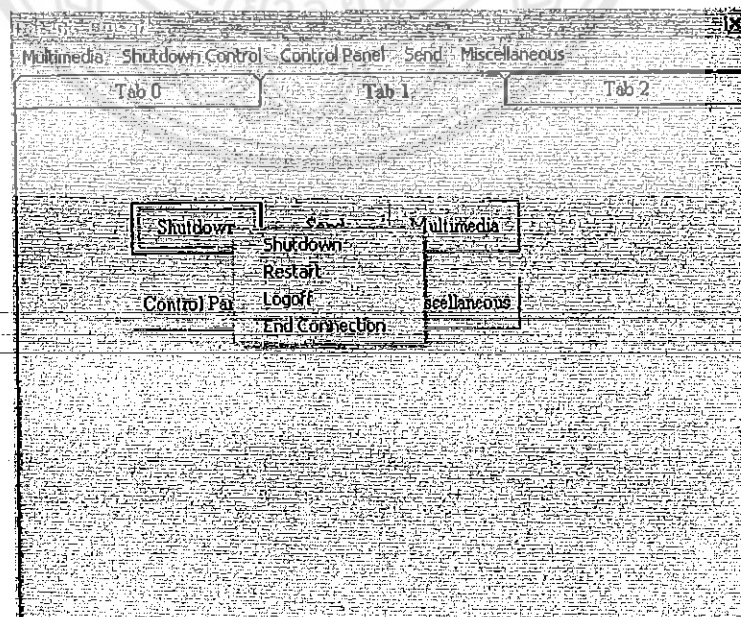
รูปที่ 4.5 หน้าจอเครื่อง server



รูปที่ 4.6 ผลที่ได้รับ

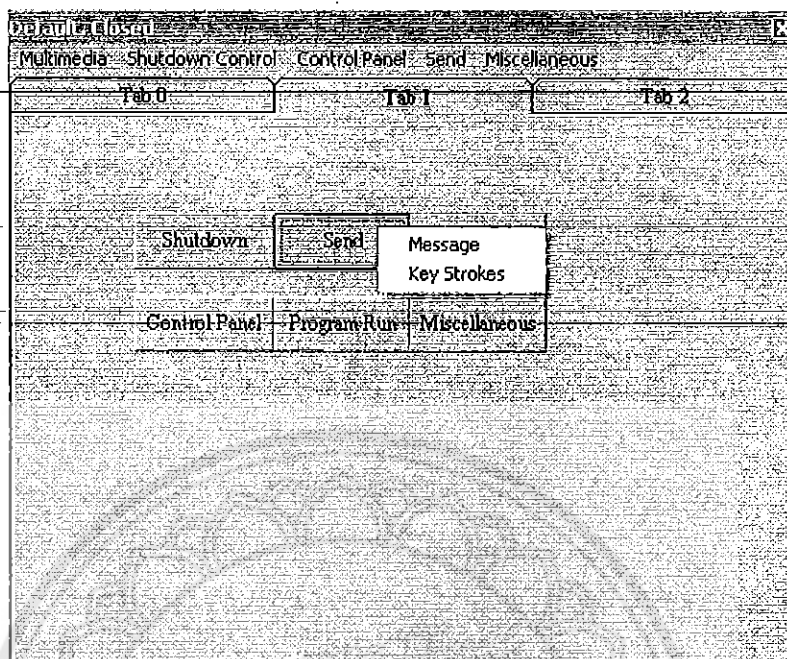
4.1.4 การควบคุมเครื่องลูกข่าย

4.1.4.1 เป็น command ในการสั่งปิดเครื่อง



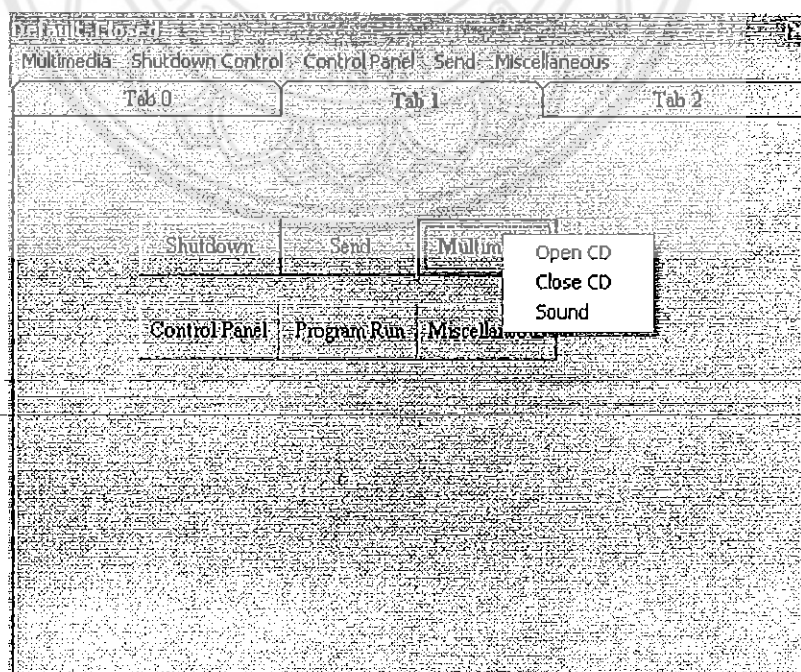
รูปที่ 4.7 การปิดเครื่อง

4.1.4.2 เป็น command ในการส่งคำพูดไปยังเครื่องลูกข่าย



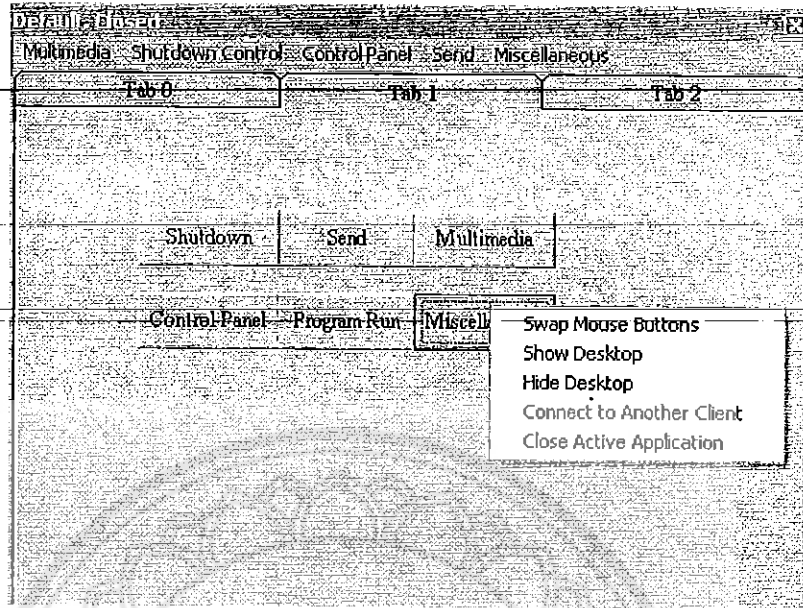
รูปที่ 4.8 การส่งคำพูดไปยังเครื่องลูกข่าย

4.1.4.3 เป็น command ในการเปิดปิด เครื่องเล่น CD และ sound



รูปที่ 4.9 การเปิดปิด เครื่องเล่น CD และ sound

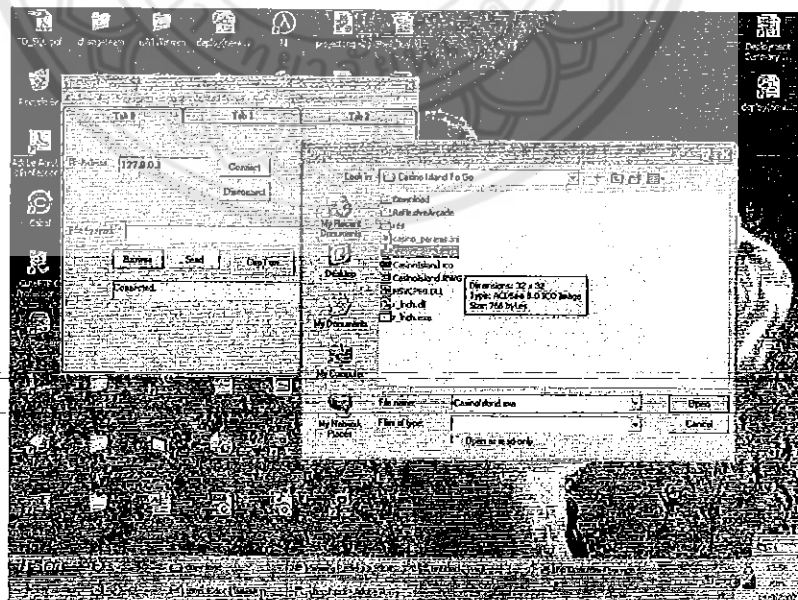
4.1.4.4 เป็น command ในการควบคุม mouse และ หน้าจอ



รูปที่ 4.10 การควบคุม mouse และ หน้าจอ

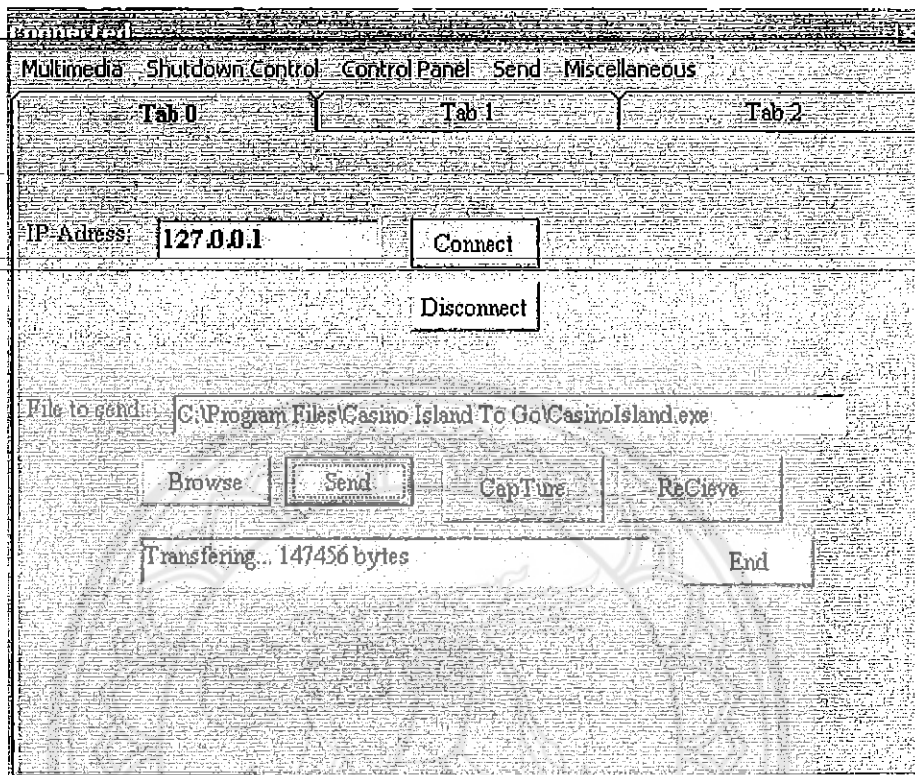
4.1.4.5 การส่งไฟล์ไปให้เครื่องลูกข่าย

4.1.4.5.1 ให้เลือกไฟล์ที่จะส่งไป

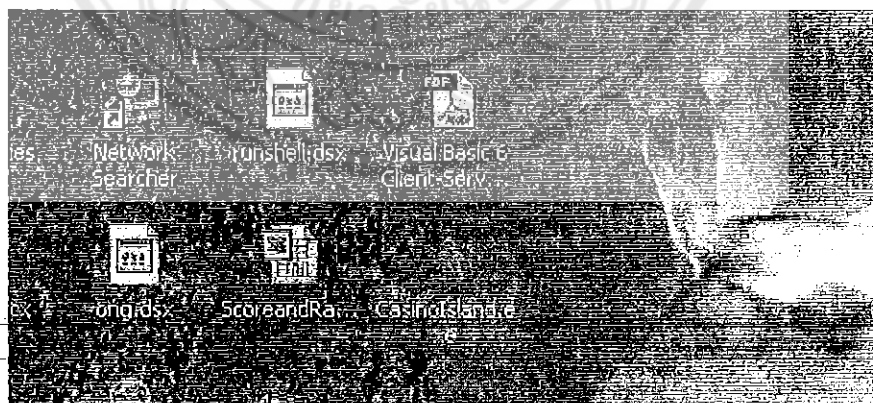


รูปที่ 4.11 แสดงการเลือกไฟล์ที่จะส่ง

4.1.4.5.2 เมื่อเลือกไฟล์เสร็จแล้วให้กด send จะมีค่า status ขึ้นมาว่าส่งไฟล์ไปแล้วเป็นจำนวนเท่าไร ให้รอจนstatus เป็น complete



รูปที่ 4.12 Status ขณะส่ง



รูปที่ 4.13 เมื่อส่งเสร็จแล้วก็มีไฟล์ยังจุดหมาย

ปัญหาที่เกิดขึ้น คือ มีการ transfer ข้อมูลที่ช้า จึงไม่สมควรที่จะส่งไฟล์ใหญ่ๆ เพราะอาจจะทำให้มีการerror เกิดขึ้นได้ จึงแนะนำว่าไม่ควรส่งไฟล์เกินครึ่งละ 10 Mb

บทที่ 5

บทวิจารณ์และสรุป

5.1 วิจารณ์โครงการ

โครงการนี้ในตอนแรกตั้งใจทำให้มีการ refresh ภาพ เป็นแบบ real-time แต่พบว่าต้องการความเร็วในการส่งข้อมูลระหว่างเครื่องและต้องการความเร็วในการรีเฟรชข้อมูลของการปรับปรุงภาพหน้าจอของฝั่งไคลเอนต์สูง หากความเร็วในการส่งข้อมูลไม่เร็วพอจะทำให้ภาพที่เกิดขึ้นเกิดอาการกระตุกได้และ vb เองก็ มีการกระพริบอยู่มากในการ refresh ความเร็วสูง จึงทำให้ต้องมาทำเป็น การส่งรูปหน้าจอมาแบบภาพ ต่อ ภาพแทนแต่ว่าเครื่องมือชิ้นนี้เป็นเครื่องมือที่ใช้ในระบบปิดคือออกแบบให้ผู้ดูแลระบบนำมาใช้กับกลุ่มคอมพิวเตอร์ที่เป็นเครื่องสาธารณะอยู่แล้ว เช่น เครื่องคอมพิวเตอร์ในห้องปฏิบัติการ หรือห้องคอมพิวเตอร์ที่เปิดให้พนักงานเข้ามาหาข้อมูล ซึ่งมีการควบคุมการใช้งานเครื่องคอมพิวเตอร์ที่คืออยู่แล้ว จึงไม่เกิดปัญหาตามมา

5.2 ปัญหาทั่วไปที่เป็นอุปสรรคต่อการพัฒนาโปรแกรม

5.2.1 ในการหาข้อมูลในแต่ละเรื่องที่จะเอามาใช้นั้นบางเรื่องต้องเสียเวลาในการค้นหาข้อมูลเป็นเวลานานทำให้เสียเวลาเป็นจำนวนมาก

5.2.2 ปัญหาจากความสามารถของตัวภาษาที่นำมาใช้ จึงทำให้ต้องค้นหาเทคนิคต่างๆมาแก้ปัญหาเหล่านั้น

5.2.3 เครื่องมือชุดนี้ทำงานอยู่บนแพลตฟอร์มวินโดวส์ไม่สามารถทำงานบนแพลตฟอร์มอื่นๆที่ไม่ใช่ระบบปฏิบัติการของไมโครซอฟท์ ในระบบคอมพิวเตอร์ที่มีเครื่องคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการอื่นนั้นไม่สามารถเข้ามาดูแลผ่านโปรแกรมชุดนี้ได้

5.3 แนวทางการแก้ไข และการพัฒนาต่อ

จากปัญหาต่างๆที่ผู้จัดทำโครงการได้พบมานั้นเมื่อนำมาวิเคราะห์เพื่อหาทางแก้ไขและพัฒนาต่อไปนั้น ทางผู้จัดทำได้สรุปออกมาได้ดังนี้คือ

5.3.1 ควรจะเปลี่ยนภาษาที่ใช้ให้กลายเป็น ภาษา c เพราะว่าจะสามารถเข้าถึง io ของเครื่องได้มากขึ้น

5.3.2 การส่งไฟล์หรือข้อมูลระหว่างเครื่องแม่ข่ายและลูกข่าย ต้องมีการ เข้ารหัสไฟล์ไว้เพื่อความปลอดภัย และความรวดเร็ว

5.3.3 ควรจะทำให้การสอคนเครื่องถูกข่ายสามารถทำได้ในลักษณะของ real-time เพราะว่า ถ้ามีการเข้ารหัสไฟล์และเปลี่ยนภาษาที่ใช้งานแล้วจะสามารถมีความสามารถที่จะtransfer data เร็วขึ้น ทำให้สามารถสอคนเครื่องถูกข่ายในลักษณะ real – time ได้

5.3.4 ควรจะทำให้มีการ log in ได้ที่หลายๆเครื่อง เพราะจะทำให้ admin มีความสะดวกมากกว่าที่จะเข้าถึงการใช้งานของเครื่องถูกข่ายทีละเครื่อง

5.3.5 ควรจะมีการพัฒนาส่วนของการส่งไฟล์และการส่งคำสั่งให้แยกออกจากกันในส่วน ของโปรแกรมด้าน client เพราะว่า ในตัวโปรแกรมนี้ ทางผู้จัดทำได้ประสบปัญหาในการที่จะซ่อน โปรแกรมไม่ให้ user พบ ดังนั้น เมื่อทาง server ต้องการจะส่งไฟล์ให้ทาง user ต้องมีการตอบรับ การส่ง จากทาง user ด้วย ทำให้ทาง user สามารถรู้ตัวได้ว่า มีโปรแกรมที่กำลัง run เป็น background process อยู่ด้วย

5.4 สรุปผลโครงการ

โครงการชิ้นนี้ประกอบด้วยฟังก์ชันการทำงานที่หลากหลายทำให้เสียเวลาในการค้นหา ข้อมูลและทำความเข้าใจเป็นเวลานาน และผลงานที่ออกมาก็สามารถทำงานได้ในระดับนี้ และมี ประสิทธิภาพที่จะนำไปใช้งานจริงได้ จากโครงการชิ้นนี้ทำให้ผู้จัดทำได้ความรู้ในการ เขียน โปรแกรมผ่านเน็ตเวิร์กมากยิ่งขึ้น และยังทำให้ผู้พัฒนานำความรู้ที่ได้ไปพัฒนาเพิ่มทั้งในด้าน vb, network หรือโปรแกรมอื่นๆได้อีกมากมาย

เอกสารอ้างอิง

<http://www.101-idea.com/vb/indexo.html>

<http://www.vbthailand.com/>

<http://thaidev.com/>

http://www.webboard.mahamodo.com/webboard_others/dev_vb/web_main.asp





ความหมายและความเป็นมาของ Visual Basic

ภาษา Visual Basic เป็นภาษาคอมพิวเตอร์ (Programming Language) ที่พัฒนาโดยบริษัท ไมโครซอฟท์ซึ่งเป็นบริษัทที่สร้างระบบปฏิบัติการ Windows ที่ใช้กันอย่างแพร่หลายในปัจจุบัน โดยตัวภาษามีรากฐานมาจากภาษาเบสิก (Basic) ซึ่งย่อมาจาก Beginner's All Purpose Symbolic (ชุดคำสั่งหรือ ภาษาคอมพิวเตอร์ สำหรับผู้เริ่มต้น) ภาษาเบสิกมีจุดเด่นคือ ผู้ที่ไม่มีพื้นฐานเกี่ยวกับการเขียนโปรแกรมเลย ก็สามารถเรียนรู้และนำไปใช้งานได้โดยง่ายและรวดเร็ว เมื่อเทียบกับภาษาอื่นๆ เช่น ภาษาซี (C) ภาษาปาสคาล (Pascal) หรือแอสเซมบลี (Assembler)

ไมโครซอฟท์ได้พัฒนาโปรแกรมภาษา Basic นับตั้งแต่ภาษา MBasic (Microsoft Basic), BasicA (Basic Advanced) และ Quick Basic ซึ่งได้คิดค้นมาพร้อมกับระบบปฏิบัติการ MS Dos ในที่สุดใช้ชื่อว่า QBasic แต่ละเวอร์ชันที่ออกมามีการเพิ่มคำสั่งต่างๆ เข้าไปโดยตลอด ซึ่งโปรแกรมเหล่านี้ล้วนแต่ทำงานใน Text Mode คือเป็นตัวอักษรล้วนๆ จนกระทั่งมีระบบปฏิบัติการ Windows ทางไมโครซอฟท์ก็ได้ปรับปรุงภาษา Basic ออกมาใหม่เพื่อสนับสนุนการทำงานบน Windows ทำให้ Visual Basic ถือกำเนิดตั้งแต่นั้น

Visual Basic 1.0 เป็นเวอร์ชันแรกออกมาเมื่อปี 1991 โดยในช่วงแรกนั้นยังไม่มี ความแตกต่างจาก QBasic มากนัก แต่จะเน้นเครื่องมือที่ใช้ในการเขียนโปรแกรมบนวินโดวส์ ซึ่งได้รับความนิยมเป็นอย่างดี ไมโครซอฟท์จึงได้พัฒนา Visual Basic ให้ดีขึ้นเรื่อยๆ จนในปัจจุบัน เวอร์ชันล่าสุดคือ Visual Basic 6.0 ออกมาในช่วงปี 1998 ได้เพิ่มความสามารถ ในการเขียนโปรแกรมติดต่อกับเครือข่ายอินเทอร์เน็ต ด้านการเชื่อมต่อกับระบบฐานข้อมูล รวมทั้งเครื่องมือและการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ให้สมบูรณ์และสะดวกยิ่งขึ้น

ข้อดีของการเขียนโปรแกรมด้วย Visual Basic

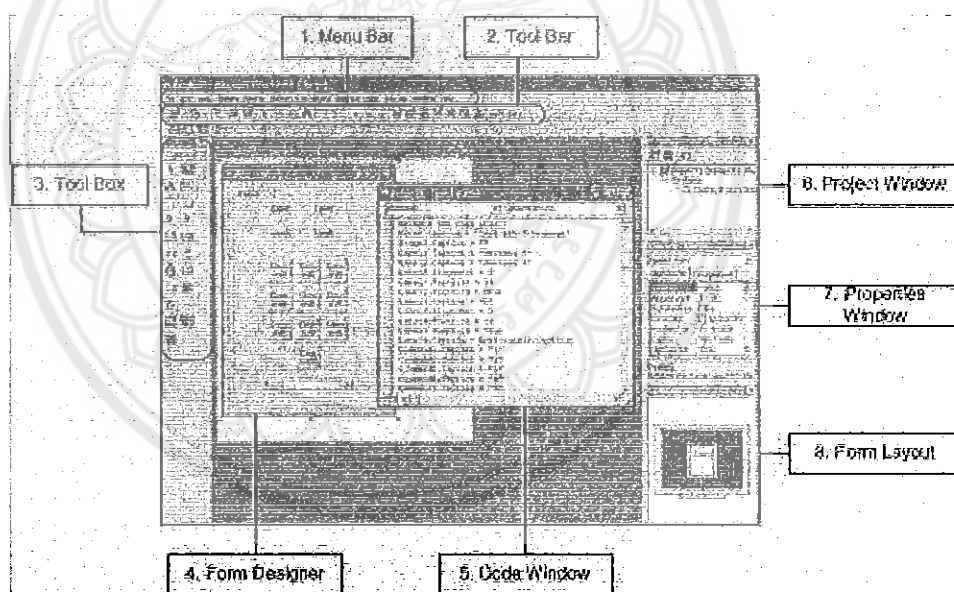
ข้อดีของการเขียนโปรแกรมด้วย Visual Basic คือง่ายต่อการเรียนรู้ และเหมาะสำหรับผู้เริ่มต้น ทั้งในเรื่องของไวยากรณ์ของภาษา และเครื่องมือในการใช้งาน ภาษา Basic เป็นภาษาที่มีคนเรียนรู้ และมีการใช้งานมากที่สุดในประวัติศาสตร์ของคอมพิวเตอร์ ภาษา Visual Basic มีการพัฒนาอย่างต่อเนื่องทั้งในด้านการปรับปรุงประสิทธิภาพของตัวภาษาและความเร็วในการประมวลผล และในด้านความสามารถใหม่ๆ เช่น ความสามารถในการติดต่อกับระบบฐานข้อมูล การเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต เป็นต้น

ลักษณะการเขียนโปรแกรมของ Visual Basic

การเขียนโปรแกรมของ Visual Basic จะอยู่ในลักษณะของ Event-Driven คือ เป็นการเขียนโปรแกรมที่ตอบสนองต่อการควบคุมเหตุการณ์ต่างๆ มากมาย ที่เกิดจากการกระทำของผู้ใช้ เป็นการเขียนโปรแกรมที่ผูกไว้กับเหตุการณ์ ที่สามารถเกิดขึ้นกับออบเจ็กต์ หรือส่วนประกอบต่างๆ ที่อยู่บนหน้าจอ

ส่วนประกอบต่างๆ ที่ใช้ในการพัฒนาโปรแกรมของ Visual Basic 6 (IDE)

คำว่า IDE หรือ Integrated Development Environment คือ สภาพแวดล้อมการทำงานในการพัฒนาโปรแกรมโดยใช้ Visual Basic หรืออุปกรณ์เครื่องมือต่างๆ ที่เตรียมมาช่วยในการพัฒนาโปรแกรมด้วย Visual Basic เมื่อเปิดโปรแกรม Visual Basic ขึ้นมาครั้งแรก จะปรากฏหน้าจอของ IDE ซึ่งมีส่วนประกอบดังนี้

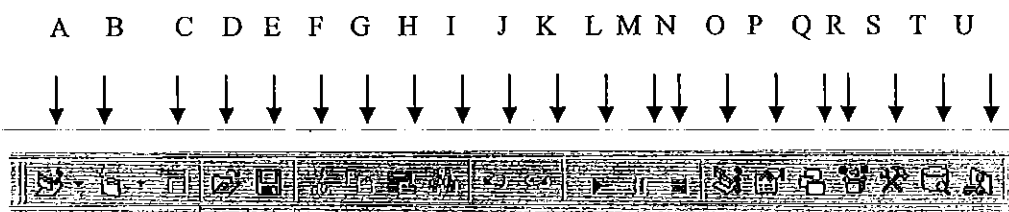


1. ส่วน IDE ของ VB จะประกอบด้วยส่วนสำคัญดังนี้

1.1 Menu : เป็นส่วนที่ใช้ในการสั่งงานทั้งหมดของ VB



1.2 Toolbar : เป็นส่วนที่ใช้ในการใช้งาน VB เช่นเดียวกัน แต่จะเลือกเฉพาะบางคำสั่งที่ใช้ งานบ่อยเท่านั้น เพื่อให้สามารถเรียกใช้คำสั่งได้สะดวกรวดเร็วยิ่งขึ้น ส่วนนี้เป็นส่วนที่สามารถ เปลี่ยนแปลงได้ตามการใช้งาน



- A. ใช้สำหรับเปิดโปรเจกต์ใหม่ขึ้นมา
- B. เพิ่มฟอร์ม โมดูล หรือออปเจกต์ ประเภทต่างๆ เขาไปใช้ในโปรเจกต์ หรือโปรแกรมที่เรากำลัง พัฒนาอยู่
- C. เปิด Menu Editor ซึ่งเป็นเครื่องมือที่ใช้ในการสร้างเมนูของโปรแกรม
- D. เปิดไฟล์โปรเจกต์
- E. บันทึกโปรเจกต์ Save
- F. ตัด(cut)
- G. ก๊อปปี้ (copy)
- H. วาง (paste)
- I. ค้นหา (Find)
- J. ยกเลิกการกระทำ (undo)
- K. เรียกคืนกับสิ่งที่ undo ไป (redo)
- L. สั่งให้โปรแกรมทำงาน (Run)
- M. ให้โปรแกรมหยุดทำงานชั่วคราว (Pause)
- N. ให้โปรแกรมหยุดทำงาน
- O. เปิดวินโดว์ Properties
- P. เปิดวินโดว์ From layout

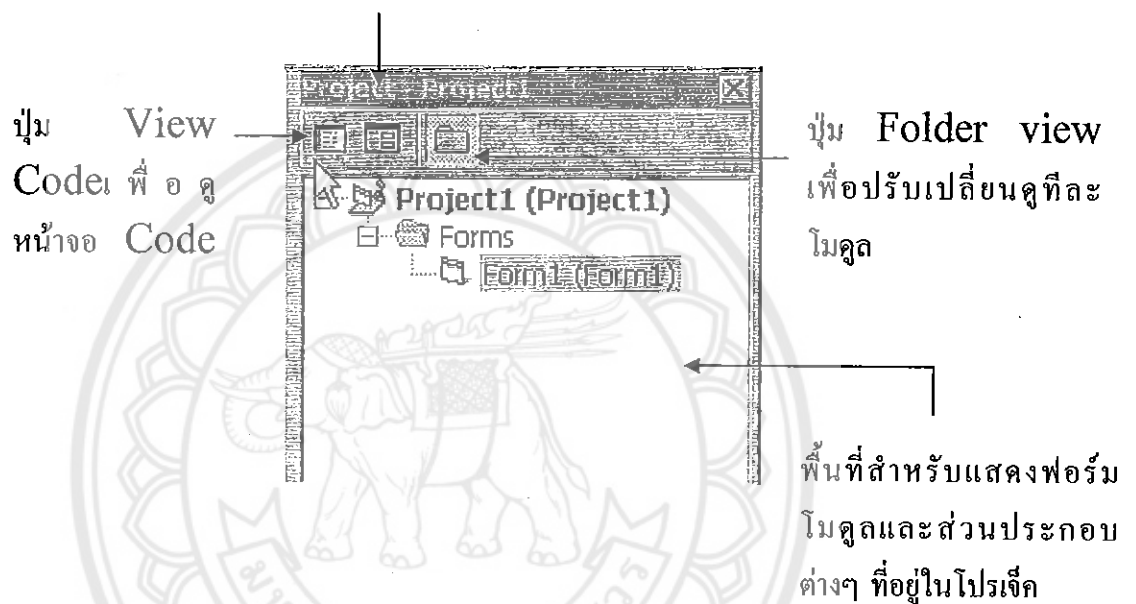
Q. Object Browser ช่วยในการค้นหารายละเอียดของออปเจ็กต์ต่างๆ

S. Toolbox เป็นแหล่งรวบรวมออปเจ็กต์ต่างๆ

T. เปิดวินโดว์ data view เพื่อดูการติดต่อกับฐานข้อมูลต่างๆ

1.3 Project Explorer

ปุ่ม Object View เพื่อกลับไป

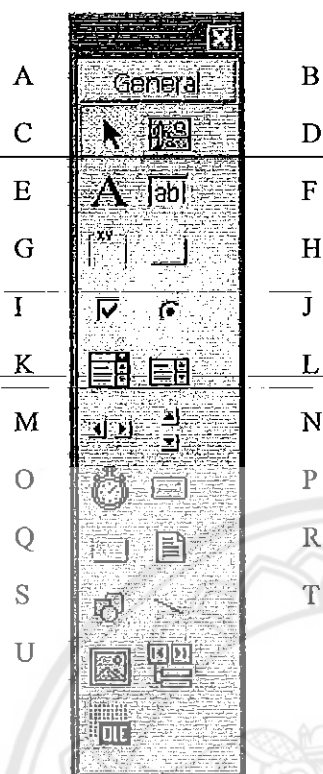


- เป็นส่วนที่ใช้ในการจัดการกับ Application ที่กำลังพัฒนาอยู่ ซึ่งจะเรียกว่า Project โดย Project จะประกอบด้วยส่วนของ Form ที่เป็นส่วนติดต่อกับผู้ใช้ (User Interface) และส่วนของ Program ที่ใช้ในการควบคุมการทำงานของ Form และการทำงานอื่น ๆ

- ในการใช้งาน IDE สามารถทำได้กับ Project มากกว่า 1 Project พร้อม ๆ กันได้

1.4 ToolBox

เป็นส่วนที่แสดงองค์ประกอบต่าง ๆ ที่สามารถใช้ในการสร้าง Application ได้ ซึ่งใน VB จะเรียกอองค์ประกอบเหล่านี้ว่า control โดยcontrolเหล่านี้จะแบ่งออกเป็น 2 กลุ่มคือ control ที่แสดงผล (Window Control) และ control ที่ไม่แสดงผล (Windowless Control) โดย control ที่แสดงผลจะใช้ในการสร้างส่วนติดต่อกับผู้ใช้ เช่น TextBox ใช้ในการป้อนข้อมูล, PictureBox ใช้ในการแสดงรูป เป็นต้น ส่วน control ที่ไม่แสดงผล จะใช้ในการทำงานอื่น เช่น Winsock จะใช้ในการติดต่อกับ Network หรือ Timer จะใช้ในการนับเวลา เป็นต้น



ToolBox สามารถเพิ่มcontrolอื่น ๆ ได้ โดยใช้คำสั่งเมนู Project → Components... ซึ่งจะ
ทำให้ Application มีความสามารถมากขึ้นตามความต้องการในการใช้งาน

A. Pointer ใช้ในการจัดขนาด เคลื่อนย้าย และวางตำแหน่งของออปเจ็คต่างๆ

B. Picture ใช้ควบคุมและแสดงภาพต่างๆ ลงบนฟอร์ม

C. Label ใช้แสดงข้อความต่างๆ ลงบนฟอร์ม

D. Textbox ใช้สำหรับรับข้อความที่ผู้ใช้ป้อนเข้ามา

E. Frame ใช้สำหรับจัดกลุ่มและรวมออปเจ็คต่างๆ เข้าด้วยกัน

F. Command Button เป็นปุ่มคำสั่งเพื่อใช้ในการสั่งงาน

G. Checkbox เป็นปุ่มที่ใช้เลือกว่าต้องการหรือไม่

H. Option Button ใช้เป็นตัวเลือกว่าใดค่าหนึ่งจากหลายๆค่า

I. Combo Box ผู้ใช้สามารถเลือกตัวเลือกได้จากการกดปุ่ม Drop Down เพื่อแสดงทางเลือกต่างๆ มีความสามารถเหมือนกับ List Box และ Text Box ผสมกัน

J. List Box ใช้แสดงตัวเลือกต่างๆ ในลักษณะของบรรทัดรายการ โดยสามารถเลือกรายการใดรายการหนึ่งหรือว่าหลายรายการก็ได้

K. Horizontal Scroll Bar เป็นแถบเลื่อนทางแนวนอน

L. Vertical Scroll Bar เป็นแถบเลื่อนทางแนวตั้ง

M. Time ใช้ในการควบคุมเวลา ในโปรแกรมที่มีเวลามาเกี่ยวข้อง

N. Drive List Box ใช้ในการติดต่อเพิ่มข้อมูลในแบบของ List Box

O. Directory List Box ใช้ในการติดต่อเพิ่มข้อมูลในแบบของ Directory ใช้ในการติดต่อเพิ่มข้อมูลในแบบของ

P. File List Box ใช้ในการติดต่อเพิ่มข้อมูลในแบบของ File List Box

Q. Shape ใช้สร้างรูปทรงต่างๆ ลงบนฟอร์ม






R. Line ใช้วาดเส้นต่างๆ ลงบนฟอร์ม
















S. Image เป็นคอนโทรล ที่ใช้ควบคุมภาพเหมือนกับ Picture แต่ความสามารถจะน้อยกว่า

T. Data Control ใช้เชื่อมต่อกับฐานข้อมูล

U. OLE เป็นคอนโทรลที่นำเอาโปรแกรมสำเร็จรูปที่มีความสามารถ OLE เข้ามาใช้เป็นแอปเจ็ทในโปรเจ็ท

คอนโทรลมาตรฐานที่ควรรู้จักได้แก่

ActiveX Controls	ไอคอน	คำอธิบาย
PictureBox		เป็นคอนโทรลที่ใช้อ่านไฟล์รูปภาพมาแสดงบนฟอร์ม
Label		เป็นแถบข้อความ มักใช้เขียนข้อความให้อ่านอย่างเดียว
TextBox		เป็นช่องให้ผู้ใช้งานกรอกข้อความ
Frame		เป็นกรอบที่จัดกลุ่มคอนโทรลต่างๆ ออกเป็นพวกๆ
CommandButton		เป็นปุ่มกดให้ผู้ใช้งานกด หรือคลิกที่ปุ่มนี้

CheckBox		เป็นปุ่มให้ผู้ใช้งานคลิกเลือก ซึ่งจะเลือกกี่ตัวก็ได้
OptionButton		เป็นปุ่มให้ผู้ใช้งานคลิกเลือก ซึ่งเลือกได้เพียง 1 ตัวเท่านั้น
ComboBox		เป็นรายการข้อมูลให้ผู้ใช้งานเลือก
ListBox		เป็นรายการข้อมูลให้ผู้ใช้งานเลือก
HScrollBar		เป็นแถบเลื่อนตามแนวนอน
VScrollBar		เป็นแถบเลื่อนตามแนวตั้ง
Timer		เป็นตัวจับเวลา
DriveListBox		เป็น ActiveX Control ที่ใช้ติดต่อกับฮาร์ดไดรฟ์ หรือซีดีรอม
DirListBox		เป็น ActiveX Control ที่ใช้ติดต่อกับโฟลเดอร์ หรือไดเรกทอรี
FileListBox		เป็น ActiveX Control ที่ใช้ติดต่อกับไฟล์
Shape		ใช้วาดรูปภาพทางเลขาคณิต
Line		เป็นคอนโทรลที่ใช้วาดเส้น
Image		เป็นคอนโทรลที่ใช้อ่านไฟล์รูปภาพมาแสดงบนฟอร์ม
Data Control		เป็นคอนโทรลที่ใช้ติดต่อกับฐานข้อมูล
OLE Control		ใช้สำหรับเรียกใช้งาน โปรแกรมอื่นที่ทำงานบนระบบ Windows

Properties

ชื่อของออปเจ็กที่กำลังแสดง

Properties อยู่

Object List Box

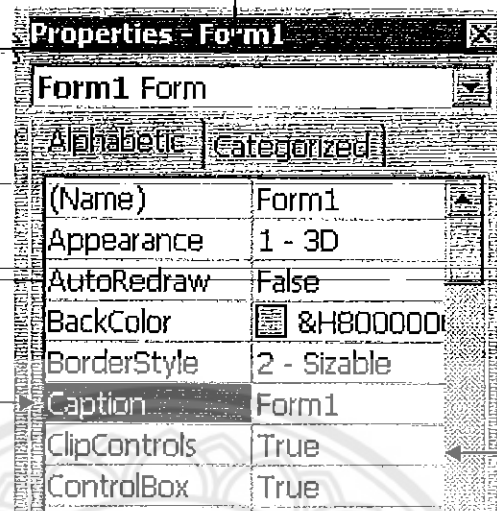
แสดงออปเจ็กที่ถูกเลือกอยู่

และจะแสดงออปเจ็ก

ทั้งหมดที่อยู่ในฟอร์ม

ชื่อ Properties ต่างๆ

ของออปเจ็ก



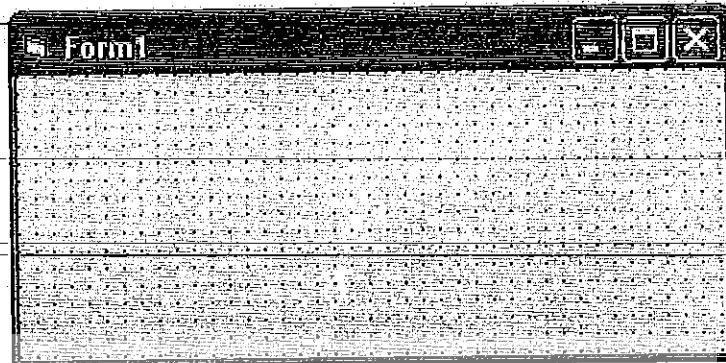
ค่าของ Properties ต่างๆ

- control ต่าง ๆ จะสามารถกำหนดรายละเอียดต่าง ๆ ได้ ทั้งในการแสดงผล และรูปแบบการทำงานของ control โดยการกำหนดในส่วน Properties ซึ่งจะเปลี่ยนแปลงรายละเอียดไปตาม control ที่เลือก การแสดง Properties สามารถแสดงได้ทั้งแบบเรียงตามลำดับตัวอักษร และแบบแบ่งเป็นหมวดหมู่

- การเปลี่ยนแปลงค่าของ Properties ต่าง ๆ จะทำให้การแสดงผล หรือการทำงานของ control เปลี่ยนไป โดยสามารถเปลี่ยนแปลง Properties ได้ทั้งในขณะการพัฒนา (design time) และเปลี่ยนแปลงได้โดย Program ในขณะที่ Run Application (run time)

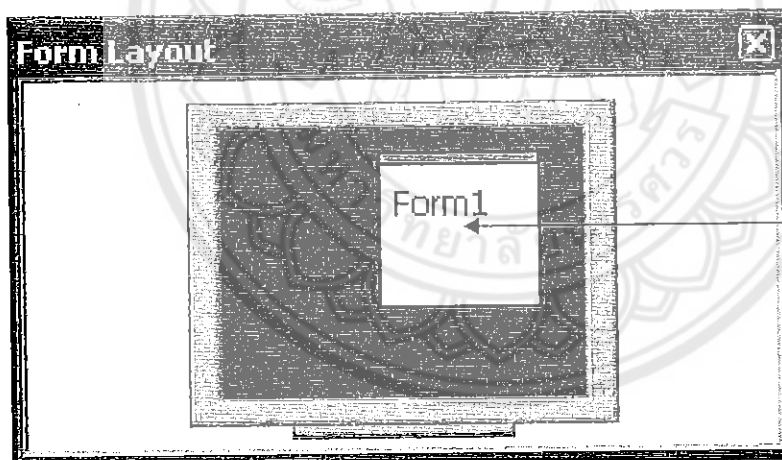
เมื่อเริ่มต้นใช้งาน VB จะเป็นการสร้าง Project ใหม่ ซึ่งสามารถเลือกได้หลายแบบตามการใช้งาน แต่โดยทั่วไปแล้วจะเป็นการพัฒนา Application แบบ Standard EXE

วินโดว์ Form



วินโดว์ Form Layout

จะเป็นตัวแสดงตำแหน่งฟอร์มของโปรเจกต์ที่เรา กำลังสร้างอยู่ ให้ดูบนจอภาพ เพื่อใช้ กำหนดตำแหน่งขณะที่โปรแกรมทำงานจริงๆ ลักษณะของ Form layout Windows ดูได้ดังรูป

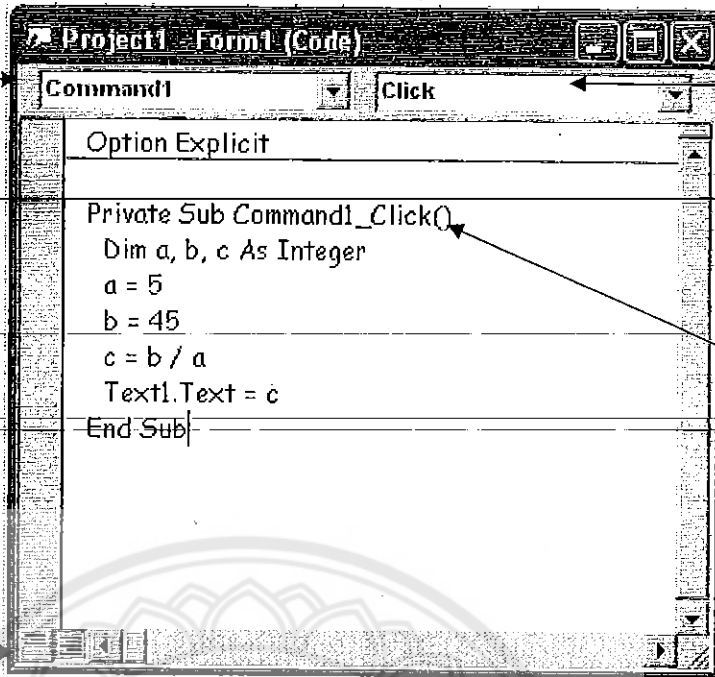


ใช้เมาส์ลาก (Drag) เพื่อกำหนด ตำแหน่งของฟอร์มบนจอภาพ

วินโดว์ Code Editor

เป็นเนื้อที่ใช้สำหรับเขียนโปรแกรม สามารถเรียกได้โดยใช้เมนู View/Code หรือดับเบิลคลิกที่ออบเจกต์ใดๆ บนฟอร์ม ซึ่งลักษณะของ Code Editor Windows ดูได้ดังรูป

List Box แสดง
รายชื่อของออบเจ็ค
และใช้เรียกออบเจ็ค
ของโปรแกรม

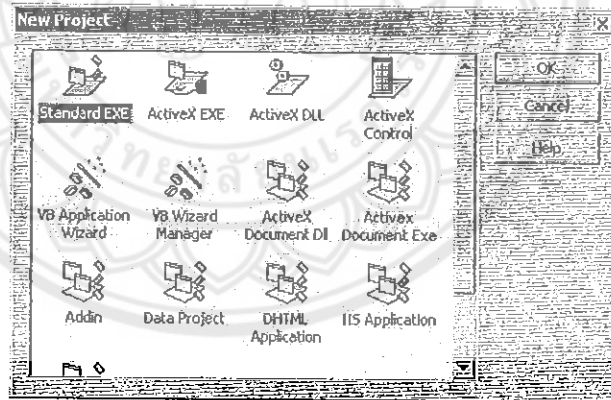


List Boxแสดง
รายชื่ออีเวนต์ของ
ออบเจ็คที่กำลังเขียน
โปรแกรมอยู่และใช้
เลือกเหตุการณ์ของ
ออบเจ็ค

View ดูอีเวนต์
ปัจจุบันของทุก
โปรแกรมที่เขียน

View ดูอีเวนต์ของทุก
โปรแกรมที่เขียน

ชื่อโพรซีเจอร์
ของเหตุการณ์



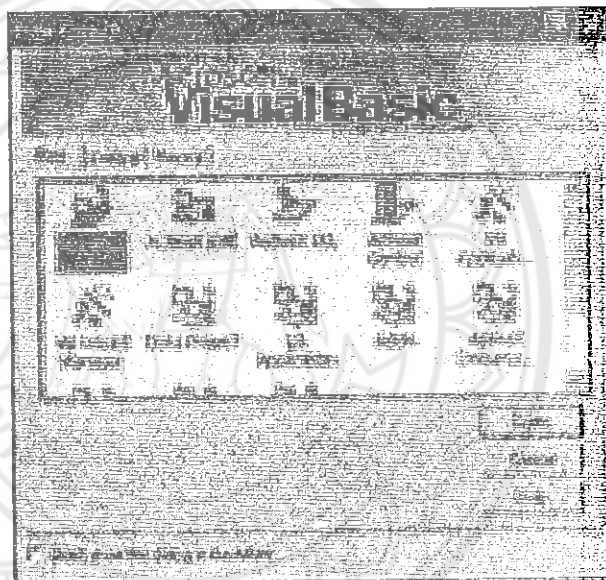
หน้าจอตั้งรูปร่างด้านนี้จะปรากฏขึ้นเสมอ เมื่อเรียกใช้คำสั่ง New Project (File → New Project)

Application Type	Description
Standard EXE	ใช้สร้าง program ในแบบ Graphic User Interface (GUI) ทั่วไป
Active EXE	ใช้สร้าง program ที่ใช้ติดต่อกับ program อื่น ในรูปแบบของ Object Linking and Embedding (OLE) แบบหนึ่งที่เรียกว่า OLE Automation Server
Active DLL	ใช้สร้าง program ชนิดเดียวกันกับ Active EXE แต่เก็บอยู่ใน file นามสกุล DLL แทน โดยไม่สามารถ run ได้ด้วยตัวเอง แต่จะถูกเรียกใช้โดย program อื่น
ActiveX Control	ใช้สร้าง control ขึ้นใช้เองใน program
VB Application Wizard	เป็นเครื่องมือที่ช่วยสร้าง program ตามโครงสร้างของ Database
VB Wizard Manager	ใช้สร้าง program ที่ควบคุมการทำงานของ Wizard
ActiveX Document Dll	ใช้สร้าง program ที่ run บน Internet Explorer แต่อยู่ในรูป file นามสกุล DLL
ActiveX Document Exe	ใช้สร้าง program ที่ run บน Internet Explorer แต่อยู่ในรูป file นามสกุล EXE
Adding	ใช้เพิ่ม Utility อื่น ๆ เข้าไว้ใน VB
Data Project	ใช้สร้าง program ที่ใช้ติดต่อกับฐานข้อมูลต่าง ๆ ผ่านทาง Open Database Connectivity (ODBC) หรือ Linking and Embedding Database (OLEDB)
DHTML Application	ใช้สร้าง program ที่ใช้งานบน Internet แบบ Dynamic HTML
IIS Application	ใช้สร้าง program ที่ใช้งานบน Internet แบบ IIS
VB Enterprise Edition Controls	ใช้สร้าง program ในแบบ Graphic User Interface (GUI) โดยทั่วไป จาก Control ต่าง ๆ ของ Enterprise Edition

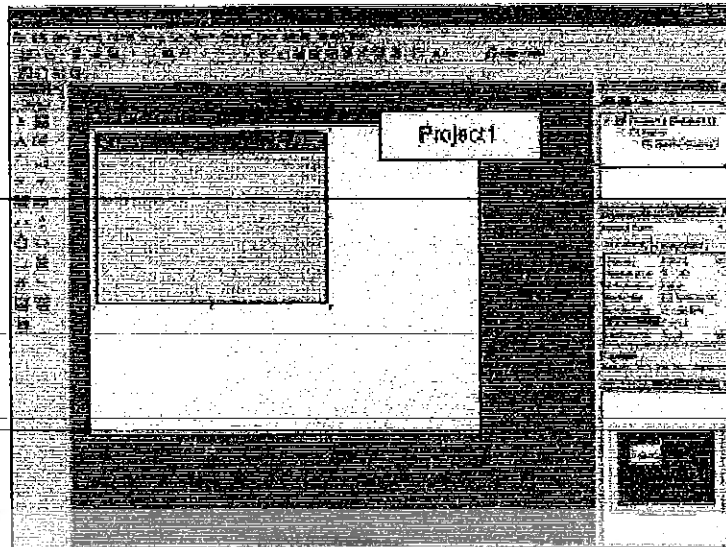
เริ่มต้นการสร้าง Application ด้วย Visual Basic 6

1. ขั้นตอนการเปิด Project

ขั้นตอนแรกในการเรียก Visual Basic ขึ้นมาใช้งานนั้นจะต้องทำการเปิด Project ขึ้นมาก่อน โดยที่ Visual Basic นั้นมี Project ให้เลือกหลายรูปแบบ ทั้งโปรแกรมธรรมดาที่รันบน Windows หรือเป็นโปรแกรมที่ทำงานบนเว็บ และ สร้างเป็น โปรแกรมที่ไม่ได้ทำงานเองแต่คอยให้คำสั่งหรือความสามารถอื่นๆ ให้โปรแกรมอะไรก็ได้เรียกใช้งาน ในการเขียน Application ทั่วๆ ไปนั้น จะต้องเลือกเปิด Project แบบ Standard EXE ซึ่งก็คือ Project ที่ใช้สร้าง โปรแกรมชนิดที่รันบน Windows

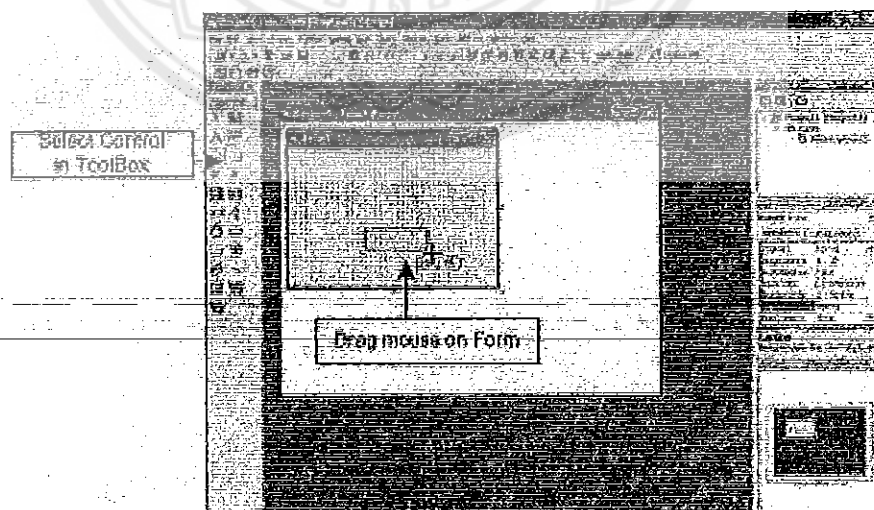


เมื่อเปิด Project แล้ว สิ่งที่จะปรากฏต่อสายตาเราก็คือ IDE ของ Visual Basic โดยจะสังเกตเห็นว่าตอนนี้ Project ถูกตั้งชื่อว่า Project1 ตามที่ทาง Visual Basic ตั้งไว้ให้



2. ขั้นตอนการสร้างคอนโทรลลงบนฟอร์ม

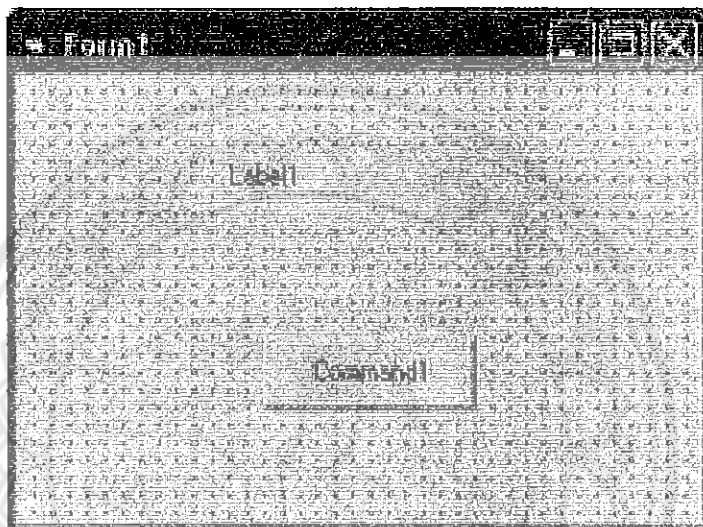
ในการสร้างคอนโทรลลงบนฟอร์มนั้นสามารถทำได้โดยการเลือกคลิกที่ตัวคอนโทรลในส่วนของ Toolbox จากนั้นก็ให้แดรกเมาส์ลงบนฟอร์มก็จะได้คอนโทรลที่ต้องการ เช่น ถ้าต้องการสร้างคอนโทรล CommandButton ลงบนฟอร์มก็ให้คลิกที่ตัวคอนโทรล CommandButton ใน Toolbox จากนั้นก็ให้ทำการแดรกเมาส์แล้วลากบนฟอร์ม เพื่อสร้างคอนโทรล CommandButton ลงบนฟอร์ม



3. ขั้นตอนการเขียนโค้ดควบคุมการทำงานของโปรแกรม

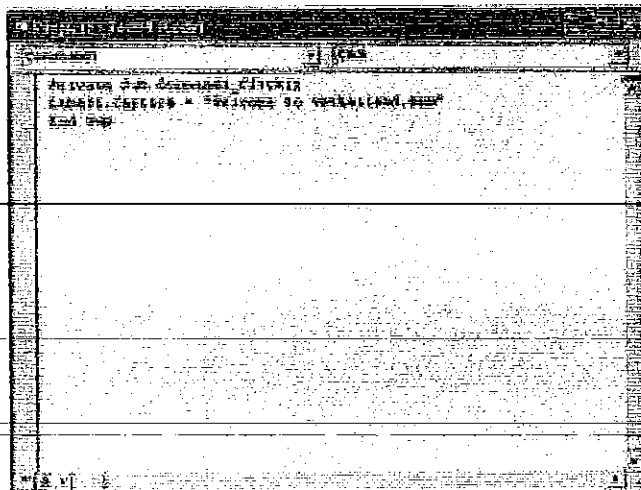
เมื่อได้ทำการออกแบบและสร้างรูปแบบหน้าต่างของ Application ของแล้ว ขั้นตอนต่อไปก็จะเป็นการเขียนโค้ด เพื่อควบคุมการทำงานของโปรแกรม ซึ่งสามารถทำได้ง่ายๆ โดยการดับเบิลคลิกที่ตัวคอนโทรล ที่ต้องการให้โปรแกรม ทำงานหลังจากที่คอนโทรลนั้นๆ Active โปรแกรมก็จะแสดงหน้าต่าง Code Window ซึ่งสามารถเขียนโค้ดควบคุม การทำงานของโปรแกรมได้ที่หน้าต่างนี้

ตัวอย่าง : ให้ทำการสร้างคอนโทรล Label และคอนโทรล CommandButton ดังรูป



จากนั้นให้ทำการดับเบิลคลิกที่คอนโทรล Command1 ก็จะปรากฏหน้าต่าง Code Window ขึ้นมา ให้ทำการพิมพ์คำสั่งต่างๆ ดังนี้

```
Private Sub Command1_Click()
Label1.Caption = "Welcome"
End Sub
```

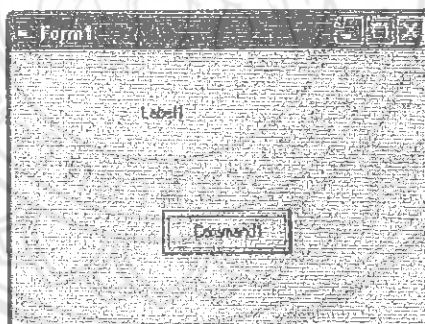



4. ขั้นตอนการรันโปรแกรม

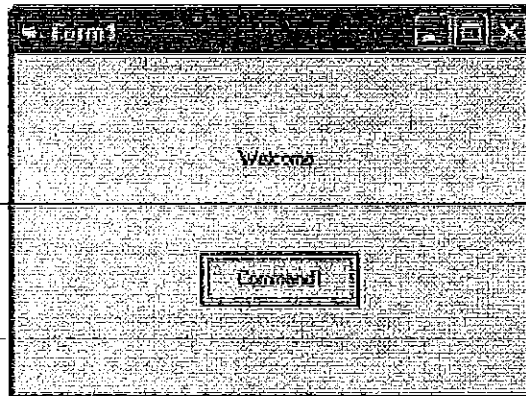
สามารถรันโปรแกรมที่ได้สร้างไว้ด้วยการกดปุ่ม F5 บนแป้นพิมพ์ หรือคลิกที่เมนู Run ->

Start ที่ Menu Bar หรือคลิกที่ไอคอน Start ที่ Tool Bar วิธีใดวิธีหนึ่ง

ตัวอย่าง : จาก Project ที่ได้สร้างค้างไว้ ให้ทดลองทำการรันโปรแกรม โดยการกดปุ่ม F5

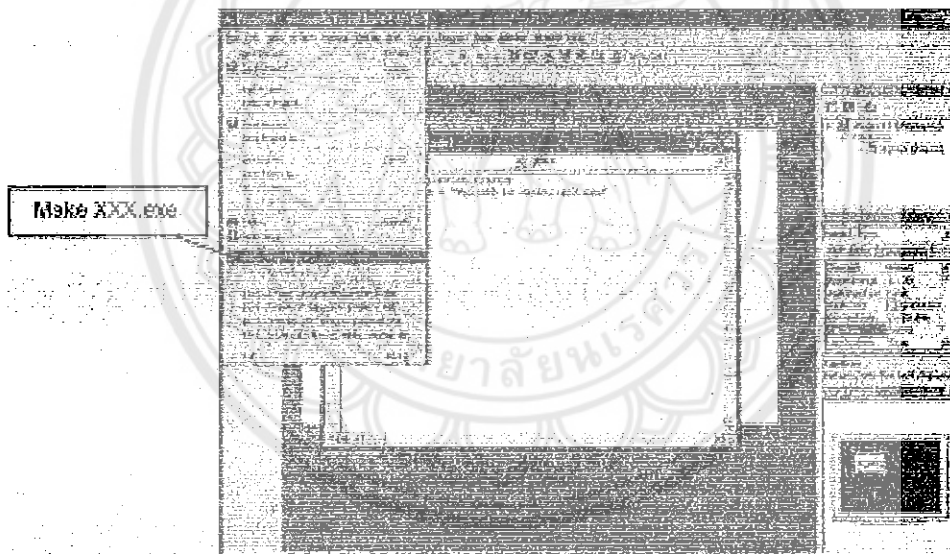


เมื่อทำการคลิกที่ปุ่ม Command1 หลังจากการรันโปรแกรมแล้ว โปรแกรมจะทำการ
แสดงผล ดังที่ได้เขียนโค้ดควบคุมโปรแกรมไว้ นั่นก็คือโปรแกรมจะแสดงข้อความ "Welcome"
บนตัวคอนโทรล Label1



5. ขั้นตอนการแปลงไฟล์ให้อยู่ในรูปแบบ EXE

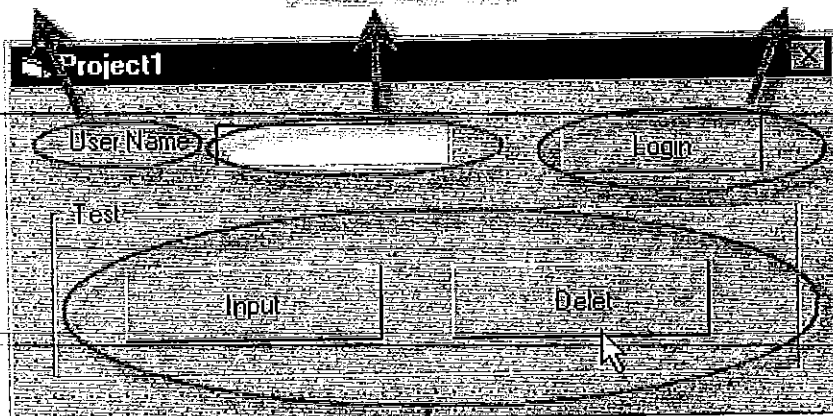
เมื่อสร้าง Application เสร็จแล้วสามารถแปลงไฟล์ให้อยู่ในรูปแบบ .EXE เพื่อให้สามารถเรียกใช้งานได้โดยไม่ต้อง เรียกผ่าน Visual Basic ซึ่งสามารถกระทำได้โดยการเลือกเมนู File -> Make ชื่อ Project... ดังนั้น จากนั้นก็ให้ระบุชื่อไฟล์ .EXE ที่ต้องการจะสร้าง



6. ขั้นตอนการใช้ Tool bar

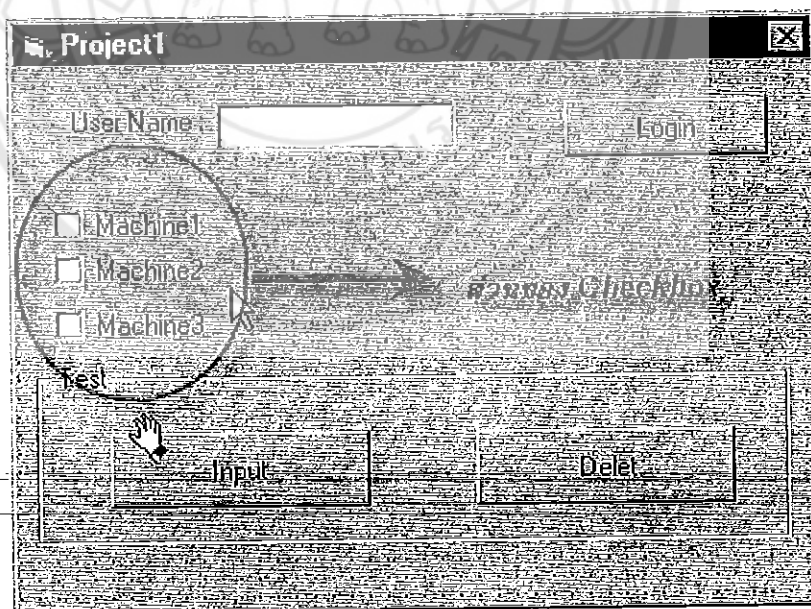
เครื่องมือในแถบ tool bar คือ checkbox option และ combo Box ในขั้นนี้จะสร้างปุ่มทั้งสามปุ่มในข้อมูลที่มีอยู่แล้ว โดยการเข้าไปคลิกที่ไฟล์ข้อมูลที่ทำกร save เอาไว้แล้ว โดยจะเริ่มที่หน้าต่างของ form ซึ่งมีการสร้างข้อมูลไว้แล้ว

ส่วนของ Label ส่วนของ Text box ส่วนของ Command Button




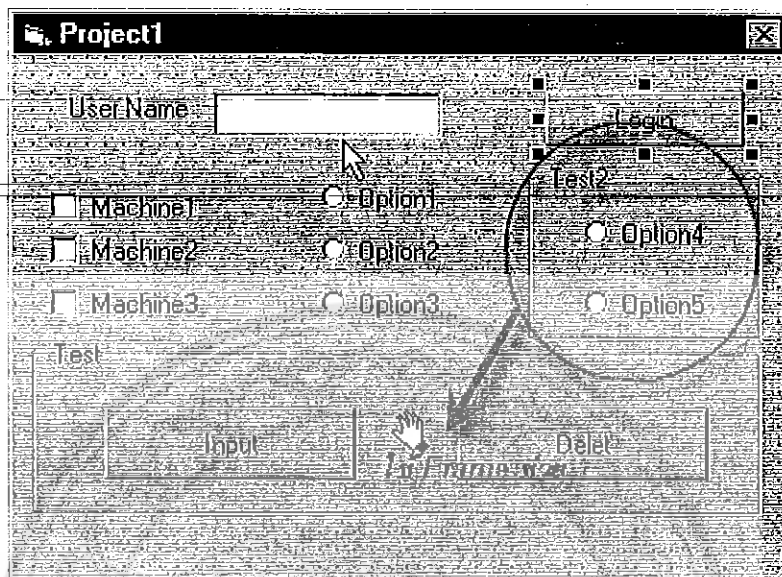
ส่วนของ Frame

จากนี้จะเป็นการสร้างข้อมูลต่อ โดยจะเพิ่มปุ่ม (checkbox) ดังรูปด้านล่างนี้

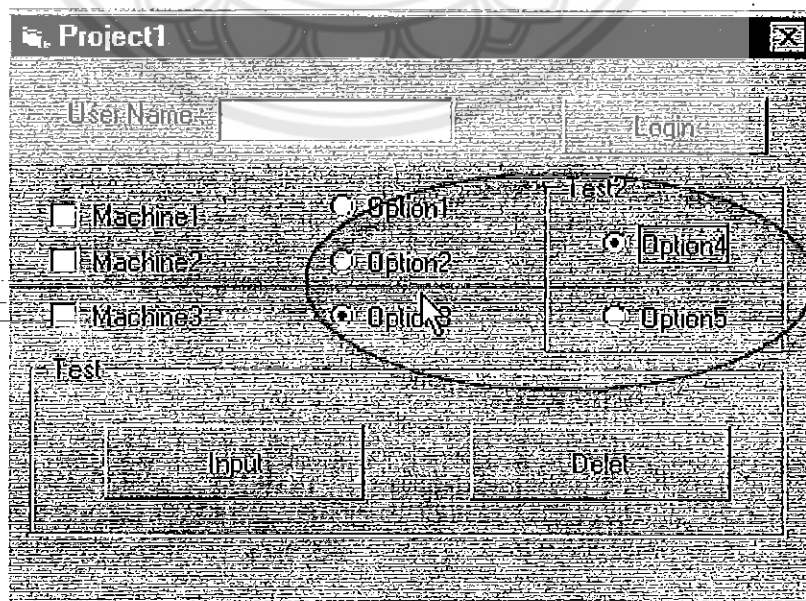


ที่ต้องสร้างปุ่ม (checkbox) ก็เพื่อความสะดวกหากมีข้อมูลหลายตัว ซึ่งปุ่มนี้สามารถที่จะเช็คข้อมูลได้พร้อมกัน และมีอีกหนึ่งทางเลือกหากต้องการที่จะให้เลือกข้อมูลอย่างไร


อย่างหนึ่งก็สามารถใช้ปุ่ม  (option) เป็นทางเลือกได้ซึ่งขึ้นอยู่กับข้อมูลที่ต้องการจะใช้และจากรูปเป็นการดึง frame เข้ามาช่วย

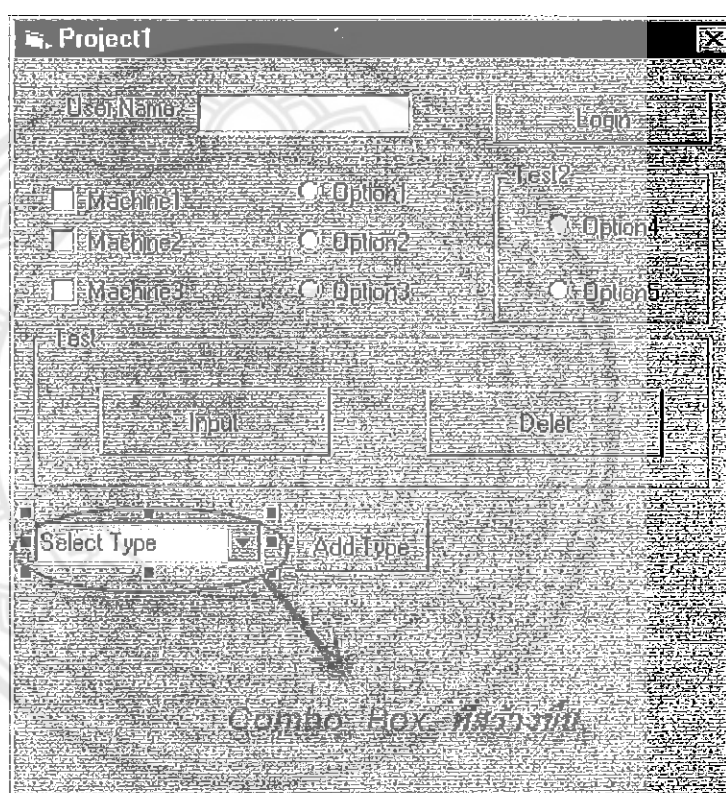


การที่ option ดึง frame เข้ามาช่วยก็เพื่อสามารถที่จะคลิก option ที่อยู่ทั้งใน frame และนอก frame พร้อมกันได้ ดังรูปที่แสดงให้เห็นจากผลการ run ด้านล่างนี้



จะเห็นได้ว่าจากผลการ run ภายในวงกลมนี้ สามารถเลือกคลิกที่ option ได้ถึงสองตัว เนื่องจากการใช้ frame เข้ามาช่วยในการจัดกลุ่มข้อมูลนั่นเอง เพราะ option ที่อยู่ใน frame กับนอก frame นั้นถือว่าเป็นคนละตัวกัน ซึ่งการที่จะกำหนดปุ่มต่าง ๆ เหล่านี้ได้มันจะต้องเข้าไปกำหนดที่ source code

ต่อมาจะกล่าวถึงตัว  (combo box) ซึ่งถ้าหากลากลงไปวางที่ตัว form จะมีลักษณะดังรูปด้านล่างนี้



ซึ่งภายใน combo box สามารถที่จะกำหนดค่าให้เป็นอะไรก็ได้ โดยให้กำหนดค่าที่ properties และในที่นี้จะยกตัวอย่างแต่จะใช้ปุ่ม command-button ช่วยกำหนดค่าให้กับ combo box ดังรูปด้านล่างนี้

```
cmdAddType Click  
Private Sub cmdAddType_Click()  
    chType.AddItem "5555"  
End Sub  
  
Private Sub cmdDelet_Click()  
    txtUn.Text = "" source code ที่ไม่ได้คิด 555  
End Sub  
  
Private Sub cmdInput_Click()  
    txtUn.Text = "Text Input"  
End Sub  
  
Private Sub cmdLogin_Click()  
    If optSelect1.Value = True Then  
        txtUn.Text = "Option1"  
    End If
```

Project1

UserName Login

Machine1 Option1

Machine2 Option2

Machine3 Option3

Test2

Option4

Option5

Option6

Test

Input Delet

5555
5555
5555

Add Type

วางรูปได้

จะเห็นได้ว่าผลการ run ที่ออกมานั้นมาจากที่กำหนดไว้คือค่าของ 5555 ซึ่งได้มาจากการคลิกที่ปุ่ม command button ที่มีชื่อว่า Add Type ถ้าคลิกที่ปุ่มไปกี่ครั้งผลก็จะออกมาเท่ากับที่คลิกนั่นเอง

คอนโทรลมาตรฐานของ VB6

ActiveX Controls นั้นทำให้สามารถสร้าง Application ได้อย่างสะดวกและง่ายดาย เพราะเป็นคอนโทรลสำเร็จรูปที่สามารถ ดึงนำมาใช้งานได้เลย ActiveX Controls ทุกตัวสามารถใช้งานได้ด้วย Development Program ที่สนับสนุนการทำงานกับ ActiveX Controls ทุกโปรแกรมเช่น Visual Basic, Visual C++ และ Delphi เป็นต้น

ActiveX Controls จะถูกเก็บไว้ที่หน้าต่าง Tool Box หรือจะกล่าวกลับกันก็คือ Tool Box เป็นหน้าต่างที่ใช้ในการเก็บ ActiveX Controls นั้นเอง ในการเขียนโปรแกรมด้วย Visual Basic นั้นจะเรียกใช้งานคอนโทรลชนิดต่างๆ เข้ามาประกอบเป็น Application ซึ่งคอนโทรลชนิดต่างๆ นั้น บางตัวก็เคยใช้งานผ่านโปรแกรมต่างๆ ของ Windows มาแล้วอาทิเช่น ปุ่มกด, Text Box และ List Box เป็นต้น แต่คอนโทรลบางตัวก็ไม่ได้ปรากฏให้เห็น แต่จะถูกเรียกใช้งานภายหลัง อาทิเช่น Common Dialog, MSChart เป็นต้น

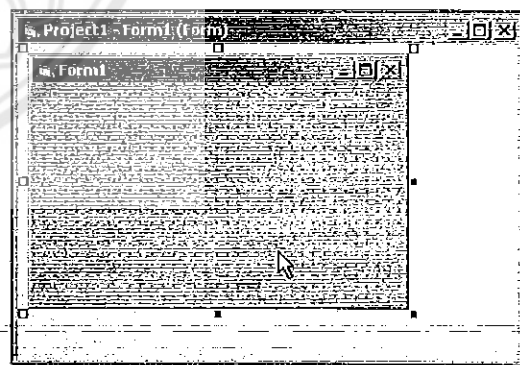
ตัวอย่าง Calendar

ในตัวอย่างนี้จะเป็นการสร้าง Application Calendar โดยมีคุณสมบัติดังนี้

- แสดงปฏิทินของเดือนปัจจุบัน และเวลาปัจจุบัน
- สามารถดูปฏิทินของเดือนอื่น ๆ ได้
- สามารถกำหนดโน้ตบันทึกของวันต่าง ๆ ได้

ขั้นตอน

เรียกใช้งาน VB และเลือกสร้าง Application แบบ



Standard EXE จากหน้าจอ New Project จะปรากฏ form ดังรูป

formจะเป็นพื้นที่ของ Application ที่สามารถวาง control ต่าง ๆ ลงบน form ได้ ซึ่งโดยทั่วไปแล้ว การสร้าง Application บน Windows จะอยู่ในรูปแบบของ form ทั้งสิ้น

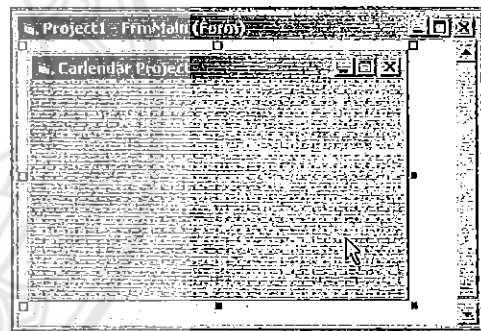
Form มี 2 แบบคือ SDI (Single Document Interface) คือ Application ที่มี Form เพียง Form เดียว หรือ อาจจะมีหลาย Form ที่แยกจากกันเป็น Form หน้าต่าง และ MDI (Multiple Document Interface) คือ Form ที่สามารถมี Form อื่นอยู่ภายใน Form ได้ ดังรูป



แต่ในตัวอย่างนี้จะเป็น Form แบบ SDI

ทำการเปลี่ยนแปลง properties ของ Form ดังนี้

- (Name) - FrmMain
- Caption - Calendar Project
- MaxButton - False
- StartUpPosition - 2 - CenterScreen




เพิ่ม control ใน ToolBox โดยใช้ Menu

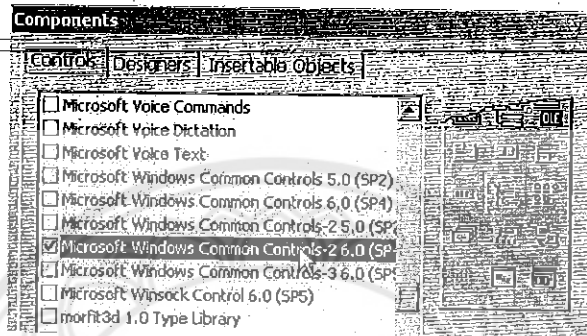
Project → Components แล้วคลิกที่ช่องสี่เหลี่ยมหน้า

Microsoft Windows Common Control-2 6.0 ดังรูป และคลิกปุ่ม OK จะสังเกตเห็นว่ามี control เพิ่มขึ้นใน ToolBox

เพิ่ม control MonthView ใน form


การเพิ่ม control ลงใน form สามารถทำได้ 2 วิธีคือใช้เมาส์ดับเบิลคลิกที่ control ที่ต้องการใน ToolBox หรือคลิกที่ control ที่ต้องการ และเลือกตำแหน่งที่ต้องการวาง control นั้นใน form คลิกเมาส์ปุ่มซ้ายค้างไว้ แล้วลากเมาส์เพื่อวาง control โดยถ้าใช้วิธีที่ 2 ขนาดของ control จะมีขนาดเท่ากับบริเวณที่ลากเมาส์

ใช้เมาส์ดับเบิลคลิกที่ control MonthView  control จะถูกวางลงใน form ซึ่งสามารถปรับขนาดของ form ได้ โดยคลิกบริเวณที่ว่างใน form แล้วเลื่อนเมาส์ไปยังบริเวณขอบของ form จนกระทั่งเมาส์เปลี่ยนเป็นเครื่องหมายลูกศร แล้วคลิกเมาส์ปุ่มซ้ายค้างไว้แล้วเลื่อนเมาส์จนได้ขนาดของ form ที่ต้องการ ซึ่งสามารถใช้วิธีนี้ได้กับ control ต่างๆ ด้วย นอกจากนี้ยังสามารถเลื่อน control เพื่อจัดวางเรียงให้มีความสวยงามได้ โดยคลิกปุ่มซ้ายของเมาส์ค้างภายใน control ที่ต้องการ ย้าย แล้วเลื่อนเมาส์ย้ายไปยังบริเวณที่ต้องการใน form




เพิ่ม control Timer

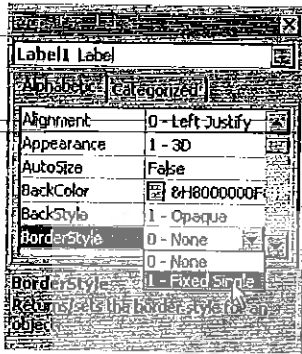


- เลื่อน control MonthView ไปทางขวา แล้วดับเบิลคลิกที่ control Timer  ซึ่ง control Timer จะเป็น control แบบไม่แสดงผล ดังนั้นจึงสามารถวางไว้ที่ตำแหน่งใด ๆ ใน form ก็ได้

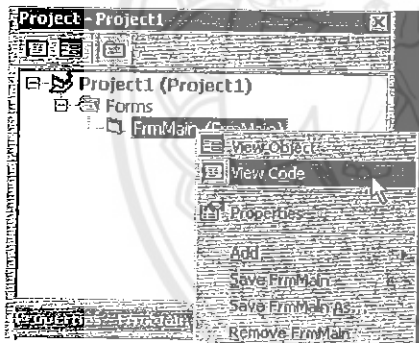
- คลิกเมาส์ที่ control Timer ใน form แล้วกำหนด properties Interval ให้มีค่า 500 ซึ่งจะเป็นการกำหนดช่วงเวลาของ Timer ซึ่ง control Timer สามารถนับช่วงเวลาได้ละเอียดถึง 1/1000 วินาที ดังนั้นค่า 500 ที่กำหนดจึงเป็นการกำหนดช่วงเวลาทุก ๆ ครึ่งวินาที

ใช้ control Label ในการแสดงเวลา

เลือก control Label  แล้วใช้วิธีวาง control วิธีที่ 2 โดยกำหนดขนาดของของ control ให้เหมาะสม แล้วเปลี่ยน properties BorderStyle ให้เป็น 1 - Fixed Single ดังรูป



การทำงานของโปรแกรมใน VB จะอยู่ในรูปแบบของ event-driven หรือทำงานตามเหตุการณ์ที่เกิดขึ้น เหตุการณ์ในที่นี้หมายถึงสิ่งที่เกิดจากผู้ใช้ เช่น การเลื่อนเมาส์ การคลิกเมาส์ การกดแป้นพิมพ์ เป็นต้น หรือเหตุการณ์ที่เกิดการ control หรือ form เอง เช่น เมื่อ form เริ่มทำงาน หรือเมื่อมีการเปลี่ยนแปลงข้อความใน TextBox เป็นต้น ซึ่งสามารถใช้การเกิดเหตุการณ์นี้ในการกำหนดให้โปรแกรมทำงาน เช่น เมื่อคลิกเมาส์ที่ปุ่ม ให้เปลี่ยนข้อความใน TextBox เป็นต้น



เขียนโปรแกรมเพื่อให้ control Label (Label1) แสดงเวลาที่ได้จาก control Timer (Timer1) โดยการเข้าสู่โหมดการเขียนโปรแกรมทำได้โดยคลิกขวาที่รายการฟอร์มใน Project Explorer แล้วเลือก View Code ดังรูป

จะปรากฏหน้าจอสำหรับเขียนprogram

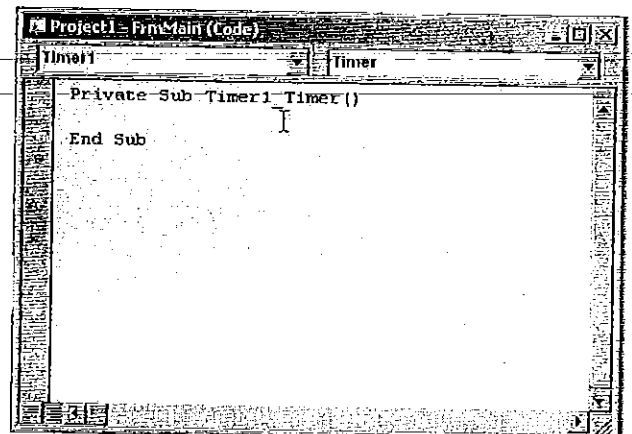
(Code) โดยส่วนบนจะแสดง-control

ที่ต้องการควบคุม (ช่องซ้าย) และเหตุการณ์

ที่เกี่ยวข้องกับ control นั้น (ช่องขวา)

ในการกำหนดให้ Label1 แสดงเวลาของ


Timer1 จะใช้เหตุการณ์ Timer ของ Timer1



โดยเลือก control Timer1 และ เหตุการณ์ Timer ดังรูป

โดยจะปรากฏ sub-routine Timer1_Timer () เปล่า ๆ ขึ้น ซึ่ง program ที่จะเขียนลงใน routine นี้จะถูกทำงานเมื่อเกิดเหตุการณ์ Timer ขึ้นใน control Timer1 โดยเหตุการณ์ Timer จะเกิดขึ้นเมื่อช่วงเวลาทำงานของ program ถึงช่วงเวลาที่กำหนดใน properties Interval (ขึ้นตอนที่ 5) ซึ่งกำหนดไว้ทุก ๆ ครึ่งนาที โดยให้เขียน program คำสั่งดังนี้

```
Label1.Caption = Time
```



ซึ่งเป็นคำสั่งกำหนดค่าเวลาให้กับ properties Caption ของ Label1 โดย properties นี้เป็น properties ที่แสดงผลบนหน้าจอ ทดลองให้ program ทำงานโดยเลือกเมนู Run → Start หรือกด F5 หรือคลิกที่ปุ่ม  บน Toolbar จะปรากฏการทำงานของ program ดังรูป



- จะเห็นว่า Timer1 จะไม่แสดงในขณะที่ program ทำงาน และ Label1 จะแสดงเวลาที่ได้จาก Timer1 ส่วน Month View ก็จะแสดงปฏิทินของเดือนปัจจุบันโดยอัตโนมัติ และสามารถเปลี่ยนไปดูเดือนอื่น ๆ ได้

- เหตุการณ์ทำงานของ program ได้โดยคลิกที่ปุ่ม  ด้านบนของ program หรือที่ปุ่ม  บน Toolbar ของ VB

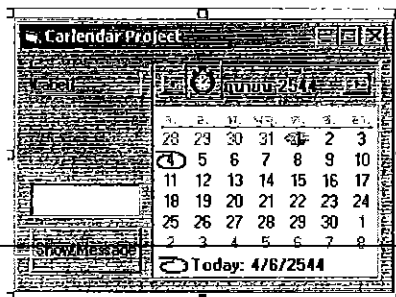
แสดงข้อความเมื่อคลิกที่ปุ่ม

เพิ่ม control TextBox  และ CommandButton  ลงใน form จัดตำแหน่งและขนาดให้เหมาะสม และกำหนด properties ของ Command1 ดังนี้

Caption – Show Message

ส่วนใน control Text1 ให้ลบตัวอักษรใน properties Text ออกทั้งหมด จะได้ดังรูป

ดับเบิลคลิกที่ CommandButton1 เพื่อสร้าง sub-routine Command1_Click() ซึ่งจะทำงานเมื่อเกิดเหตุการณ์ปุ่มถูกคลิก และพิมพ์ Program ดังนี้ (ไม่ต้องพิมพ์หมายเลขบรรทัด)



01 Dim strMsg As String

02 strMsg = WeekdayName(Weekday(Date), False, _
vbUseSystemDayOfWeek)

03 strMsg = strMsg & " is the first day of the rest!"

04 strMsg = strMsg & " of your life!"

05 Text1.Text = strMsg

ในการเขียน โปรแกรมใน VB ตัวอักษรพิมพ์เล็กหรือพิมพ์ใหญ่จะมีความหมายเดียวกัน เช่น dim จะมีความหมายเดียวกับ Dim หรือ DIM และส่วนของโปรแกรมที่เป็นคีย์เวิร์ด เช่น Dim, As, Private, Sub ชนิดข้อมูล เช่น String, Integer ชื่อฟังก์ชัน เช่น WeekdayName, vbUseSystemDayOfWeek และคำสั่งควบคุมการทำงานของโปรแกรม เช่น For, If จะถูกเปลี่ยนให้เป็นตามรูปแบบของ VB โดยอัตโนมัติ Dim เป็นคีย์เวิร์ดที่ใช้ในการประกาศตัวแปร โดยระบุชนิดข้อมูลของตัวแปรนั้นหลังคีย์เวิร์ด As เช่น Dim I As Integer Dim st1, st2 As String WeekdayName เป็นฟังก์ชันของ VB ใช้ในการหาวันในสัปดาห์ (อาทิตย์, จันทร์, ...) & เป็นเครื่องหมายที่ใช้ในการต่อเชื่อม string โดยอาจจะเป็นได้ทั้งตัวแปรและค่าคงที่

ในการเขียน โปรแกรม ถ้าเป็นประโยคคำสั่งเดียวกัน แต่ต้องการขึ้นบรรทัดใหม่ สามารถใช้เครื่องหมาย _ ในการต่อเชื่อมได้ เช่น

strMsg = WeekdayName(Weekday(Date), False, _
vbUseSystemDayOfWeek)

Programข้างต้น จะเป็นการหาค่าวันปัจจุบันในสัปดาห์ด้วยฟังก์ชัน WeekdayName และนำไปสร้างประโยคโดยการเชื่อมต่อกับประโยค " is the first day of the rest" (บรรทัดที่ 03) และประโยค " of your life!" (บรรทัดที่ 04) และนำออกแสดงใน Text1 โดยกำหนดที่ properties Text (บรรทัดที่ 05)

Save Project

เก็บ project ที่ทำทั้งหมดใน disk โดยเรียก Menu: File → Save Project แล้วตั้งชื่อ project และชื่อ form ที่ต้องการเก็บ (ให้ใช้ชื่อ project และชื่อ form ที่ VB กำหนดให้) และทดลอง run program

Properties, Methods and Events

ทุก ๆ Object (form และ control) ใน VB จะมี properties ซึ่งเป็นรายละเอียดของ Object นั้น เมื่อกำหนด properties ที่แตกต่างกันใน Object ชนิดเดียวกัน จะมีผลให้ Object นั้นแสดงผลหรือมีการทำงานที่แตกต่างกัน properties สามารถเปลี่ยนแปลงค่าได้ทั้งในขณะการพัฒนา program (design time) และในขณะที่ program ทำงาน (run time) แต่จะมี properties บางอย่างที่จะเปลี่ยนแปลงค่าได้เฉพาะในขณะที่ program ทำงานเท่านั้น

Object บางชนิดจะสามารถมี Object อื่นในตัวมันเองได้ เช่น form หรือ frame เรียก Object ประเภทนี้ว่า container

Properties ทั่วไปของ control ใน VB คือ

Property	Description
Left	ตำแหน่งด้านซ้ายของ control เทียบกับ container ที่ control นั้นอยู่
Top	ตำแหน่งด้านบนของ control เทียบกับ container ที่ control นั้นอยู่
Height	ความสูงของ control
Width	ความกว้างของ control
Name	ชื่อของ control
Enabled	ค่า Boolean (True/False) ที่กำหนดว่าจะให้ผู้ใช้สามารถใช้งาน control นั้นได้หรือไม่
Visible	ค่า Boolean (True/False) ที่กำหนดว่าจะให้ control นั้นแสดงขึ้นหรือไม่ ในขณะที่ program ทำงาน

ชื่อของ control (Name property) จะถูกกำหนดให้กับ control โดยอัตโนมัติ เมื่อ control นั้นถูกใช้ใน form เช่น Text1, Text2 หรือ Label1, Label2 ซึ่งผู้เขียน program ควรจะตั้งชื่อใหม่ให้กับ control ที่ใช้ โดยมีหลักการในการตั้งชื่อดังนี้

- นำหน้าชื่อด้วยประเภทของ control โดยอาจจะใช้แบบย่อได้ เช่น Form ใช้ Frm, TextBox ใช้ Txt, Label ใช้ Lbl, CommandButton ใช้ Cmd เป็นต้น

- ตามด้วยชื่อของ control นั้น ซึ่งสื่อถึงการใช้งานและใช้ตัวอักษรพิมพ์ใหญ่ในการแยกคำ เช่น TextBox สำหรับแสดงเวลาอาจจะมี ชื่อว่า TxtShowTime หรือ CommandButton สำหรับบอกรอกจาก program อาจจะใช้ชื่อว่า CmdExit เป็นต้น

- ชื่อของ Object จะต้องขึ้นต้นด้วยตัวอักษรเท่านั้น (a..z, A..Z) (สามารถใช้ภาษาไทยได้แต่ไม่ควรใช้) และตามด้วย ตัวอักษร, ตัวเลข หรือเครื่องหมาย “_” (underscore) เท่านั้น ตัวอักษรพิมพ์เล็กพิมพ์ใหญ่จะมีความหมายเดียวกัน และการแก้ไขชื่อของ control จะไม่มีผลกับ program ที่เขียนไว้แล้ว ซึ่งผู้พัฒนาจะต้องตามไปแก้ไขในทุก ๆ ที่ที่มีการอ้างอิง control นั้น

การตั้งชื่อที่สามารถระบุชนิดของ control และสื่อความหมาย จะช่วยให้สามารถเขียน program ได้เร็วขึ้น และสามารถทำความเข้าใจ program ได้เร็วขึ้น เมื่อนำมาแก้ไขในภายหลัง

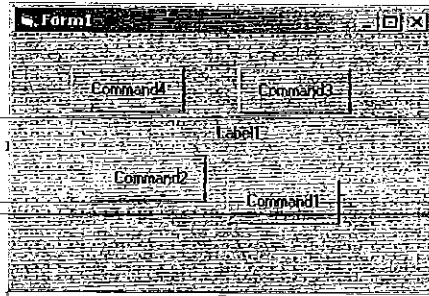
แมธอด (method) คือการกำหนดให้ Object ทำงานบางอย่าง โดยจะอยู่ในรูปแบบของ code program ซึ่งแต่ละ control ก็จะมี method ที่เหมือนและแตกต่างกัน โดย method ที่กำหนดมาแล้วของ แต่ละ control จะถูกเรียกใช้ตามเหตุการณ์ที่เกิดขึ้น หรือสามารถเรียกได้โดยตรงจาก method อื่น method อาจจะมี parameter ที่ส่งมาด้วย เช่น method KeyPress ของ TextBox จะมี parameter KeyAscii ซึ่งเป็นค่า ascii ของแป้นพิมพ์ที่กด ซึ่งสามารถนำไปใช้ใน program ได้

เหตุการณ์ (event) คือสิ่งที่เกิดขึ้นกับ program ซึ่งอาจจะเกิดขึ้นจากผู้ใช้ program เช่น การเลื่อนเมาส์ การคลิกเมาส์ การกดแป้นพิมพ์ เป็นต้น หรืออาจจะเกิดจากระบบเองก็ได้ เช่น เมื่อมีข้อมูลส่งมาจาก port สื่อสาร หรือเมื่อ form ถูกload เป็นต้น ซึ่งการเหตุการณ์หนึ่งอาจจะเกิดจากเหตุการณ์อื่น ๆ อีกหลายเหตุการณ์ก็ได้ เช่น เมื่อ CommandButton ถูกคลิก จะมีเหตุการณ์เกิดขึ้น 4 เหตุการณ์ คือ GotFocus, MouseDown, Click และ MouseUp ซึ่งในการเขียน program สามารถเลือกเหตุการณ์ที่เหมาะสมสำหรับ control นั้น ๆ ได้

ตัวอย่าง program MoveIt

User Interface

Program ในการทดลองนี้จะมี ส่วนติดต่อกับผู้ใช้งานรูป โดยจะประกอบด้วย CommandButton 4 control และ Label 1 control ซึ่งจากรูปจะเห็นว่า control ต่าง ๆ ยังไม่ถูกจัดเรียงให้เป็นระเบียบและ ไม่มีการกำหนด properties แต่อย่างใด ซึ่งทั้งหมดจะทำในprogramในขั้นตอนการ load form



เนื่องจากชื่อของ form และ control ไม่สามารถเปลี่ยนแปลงได้ในขณะที่programทำงาน ดังนั้นจึงเป็น properties ที่จะต้องกำหนดไว้ก่อนดังนี้

- ตั้งชื่อ form เป็น FrmMoveIt
- ตั้งชื่อ CommandButton เป็น CmdTopLeft, CmdTopRight, CmdBottomLeft, CmdBottomRight
- ตั้งชื่อ Label เป็น LblNotify

Event Form_Load

ในการเรียก program ทำงาน เหตุการณ์หนึ่งที่จะเกิดขึ้นคือ Form_Load ซึ่งจะมีชื่อ sub-routine นี้เสมอ ไม่ว่าจะมีการเปลี่ยนชื่อform หรือไม่ ใน sub-routine Form_Load จะทำการเปลี่ยน การแสดงผลด้วย program ดังนี้ โดยสิ่งที่อยู่หลังเครื่องหมาย ‘ จะเรียกว่าคอมเมนต์ (comment) ซึ่ง จะไม่มีผลในการทำงานของ program แต่จะใช้ประโยชน์ในการอธิบายการทำงานของ program (ส่วนที่เป็น comment ไม่ต้องพิมพ์ program ก็ได้)

01 ‘ กำหนด properties Caption ของ CommandButtons

02 CmdTopLeft.Caption = "Top Left"

03 CmdTopRight.Caption = "Top Right"

04 CmdBottomLeft.Caption = "Bottom Left"

05 CmdBottomRight.Caption = "Bottom Right"

06

07 ' ลบข้อความใน Label

08 LblNotify.Caption = ""

09

10 ' กำหนดข้อความใน title bar ของform

11 FrmMoveIt.Caption = "MoveIt"

12

13 ' จัดให้ form อยู่กึ่งกลางหน้าจอ

14 ' โดยกำหนดproperties Left ให้ มีค่าเท่ากับ(ความกว้างของหน้าจอ – ความกว้างของform) / 2

15 ' และ กำหนด properties Top ให้เท่ากับ (ความสูงของหน้าจอ – ความสูงของform) / 2

16 FrmMoveIt.Left = (Screen.Width - FrmMoveIt.Width) / 2

17 FrmMoveIt.Top = (Screen.Height - FrmMoveIt.Height) / 2

18

19 ' จัดตำแหน่งของ CommandButton

20 ' โดยกำหนด properties Left และ Top โดยให้อยู่ห่างจากขอบของform 200 pixels

21 CmdTopLeft.Left = 200

22 CmdBottomLeft.Left = 200

23 CmdTopRight.Left = FrmMoveIt.Width - CmdTopRight.Width - 300

24 CmdBottomRight.Left = FrmMoveIt.Width - CmdBottomRight.Width - 300

25

26 CmdTopLeft.Top = 200

27 CmdBottomLeft.Top = FrmMoveIt.Height - CmdBottomLeft.Height - 500

28 CmdTopRight.Top = 200

29 CmdBottomRight.Top = FrmMoveIt.Height - CmdBottomRight.Height - 500

30

31 ' กำหนดขนาดของ Label และ จัดให้อยู่กึ่งกลางของ form

32 LblNotify.Height = 360

33 LblNotify.Width = 3000

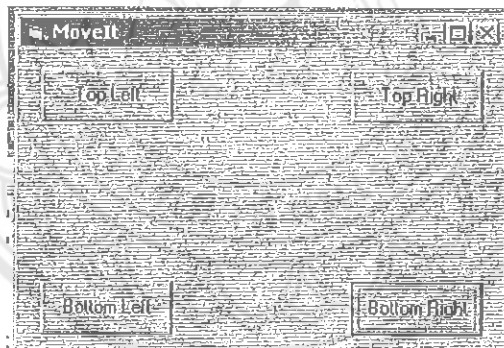
34

35 LblNotify.Left = (FrmMoveIt.Width - LblNotify.Width) / 2

36 LblNotify.Top = (FrmMoveIt.Height - LblNotify.Height) / 2 - 200

37 LblNotify.Alignment = 2

ทดลองrun program จะได้ผลของ program ดังรูป



ย้าย form ไปยังตำแหน่งต่าง ๆ ของหน้าจอเมื่อคลิกที่ปุ่ม

วิธีการจะคล้าย ๆ กับการจัดตำแหน่งของ form ให้อยู่กลางหน้าจอโดยกำหนดค่าให้กับ FrmMoveIt.Top และ FrmMoveIt.Left แต่จะเป็นการทำงานของ program เมื่อเกิดเหตุการณ์ Click ที่ CommandButton ต่างๆ ดังนี้

```
Private Sub CmdTopLeft_Click()
```

```
    FrmMoveIt.Top = 0
```

```
    FrmMoveIt.Left = 0
```

End Sub

Private Sub CmdTopRight_Click()

FrmMoveIt.Top = 0

FrmMoveIt.Left = Screen.Width - FrmMoveIt.Width

End Sub

Event notification

ขั้นตอนนี้จะกำหนดให้ control LblNotify (Label) แสดงเหตุการณ์ต่างๆ ที่เกิดขึ้น เช่น เหตุการณ์การกดปุ่มเมาส์ เหตุการณ์การปล่อยปุ่มเมาส์ เป็นต้น ดังตัวอย่าง

Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)

LblNotify.Caption = "MouseDown Event"

End Sub

Private Sub Form_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)

LblNotify.Caption = "MouseUp Event"

End Sub

Run program และ ทดลอง โดยการคลิกเมาส์บริเวณที่ว่างของ form

ประวัติผู้เขียนโครงการ



ชื่อ นายคทาฐ ยันตระภูต

ภูมิลำเนา 78/4-5 ถ.ศิริมิ่งคลาจารย์ ต.สุเทพ อ.เมือง จ.เชียงใหม่ 50200

ประวัติการศึกษา

- จบการศึกษาจากโรงเรียนสาธิตมหาวิทยาลัยเชียงใหม่

- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 5 สาขาวิศวกรรม

คอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail Ong_nu@hotmail.com



ประวัติผู้เขียนโครงการ



ชื่อ นายคทาวุธ ฮั่นตระกูล
 ภูมิลำเนา 78/4-5 ถ.ศิริมิ่งกลางจารย์ ต.สุเทพ อ.เมือง จ.เชียงใหม่ 50200
 ประวัติการศึกษา

- จบการศึกษาจากโรงเรียนสาธิตมหาวิทยาลัยเชียงใหม่
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 5 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E – mail Ong_nu@hotmail.com

