สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมคอมพิวเตอร์ลูกข่ายระยะไกล

Client Remote Control System



นายนาถรีย์ ยีตาหวี
นายประสิทธิ์ อุทาเลิศ

เลขหมู่.....
เลขทะเบียน 49990
วัน,เดือน,ปี 16 เม.ย. 2547

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมคอมพิวเตอร์ลูกข่ายระยะไกล

Client Remote Control System

โดย

นายนาถรีย์ ยีตาหวี

นายประสิทธิ์ อุทาเลิศ

อาจารย์ที่ปรึกษา

อาจารย์ ธนา หงษ์สุวรรณ

อาจารย์ อัครเดช วัชระภูพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมเครื่องลูกข่ายระยะไกล

Client Remote Control System

ผู้จัดทำ

1.นายนาถรีย์ ยีตาหวี รหัสประจำตัว 43015365

2.นายประสิทธิ์ อุทาเลิศ รหัสประจำตัว 43015370



_____ อาจารย์ที่ปรึกษา

(อาจารย์ ธนา หงษ์สุวรรณ)

_____ อาจารย์ที่ปรึกษา

(อาจารย์ อัครเดช วัชระภูพงษ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมคอมพิวเตอร์ลูกข่ายระยะไกล

นายนาถรีย์ ยีตาหวี
นายประสิทธิ์ อุทาเลิศ
อ.ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
อ.อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2545

บทคัดย่อ

ในปัจจุบันระบบเน็ตเวิร์กภายในองค์กรได้รับความนิยมอย่างแพร่หลาย ผู้ใช้งานคอมพิวเตอร์ต่างเริ่มมีการใช้งานที่หลากหลายมากขึ้น ซึ่งอาจมีผู้ใช้ส่วนหนึ่งที่ใช้งานเครื่องคอมพิวเตอร์ผิดเป้าหมายและผิดจุดประสงค์ขององค์กรนั้นๆซึ่งก่อให้เกิดปัญหาตามมามากมาย จึงเป็นปัญหาอย่างยิ่งของผู้ดูแลระบบในการดูแลการใช้งานเครื่องคอมพิวเตอร์ภายในองค์กร ให้มีการใช้งานตามระเบียบข้อบังคับที่ทางองค์กรได้กำหนดไว้ ในการควบคุมความเป็นระเบียบของเครือข่ายนั้นผู้ดูแลระบบต้องรู้ว่าผู้ใช้งานคอมพิวเตอร์อยู่นั้น ใช้งานคอมพิวเตอร์ขององค์กรผิดระเบียบการใช้งานขององค์กรหรือไม่หรือใช้คอมพิวเตอร์ทำอะไรอยู่มีความเสี่ยงหรือก่อให้เกิดปัญหาต่างๆต่อระบบหรือไม่ แล้วจึงตัดสินใจดำเนินการต่อไปว่าจะแจ้งเตือนหรือระงับการให้บริการ

โครงงานชิ้นนี้ถูกจัดทำขึ้นเพื่อเป็นเครื่องมือช่วยเหลือของผู้ดูแลระบบให้สามารถดูแลระบบเน็ตเวิร์กได้สะดวกยิ่งขึ้น พร้อมทั้งเครื่องมืออื่นๆสำหรับเพิ่มความสะดวกในการใช้งานให้การดูแลระบบง่ายขึ้นโดยมีประสิทธิภาพใกล้เคียงกับซอฟต์แวร์ที่ผลิตจากต่างประเทศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Client Remote Control System

Nattarie Yeetavee

Prasit Utalert

Thana Hongsuwan Advisor

Akkradach Watcharapupong Advisor

ABSTRACT

In these days network systems are used in places. There may be some users use computer not according to the organization purpose. That will be trouble for system administrators to maintain computers in organization. System administrators have to know how users use computer to decide if according to organization or not than. Than system administrator could alert users or stop that usage.

This project was made as a tool for system administrator to maintain network system easier as well as aboard software.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้คงไม่อาจสำเร็จลงได้ด้วยดีหากไม่ได้รับการช่วยเหลือและความร่วมมือจากหลายฝ่ายด้วยกัน บุคคลแรกที่มีส่วนสำคัญที่ทำให้โครงการนี้สำเร็จลงได้ด้วยดีคือ อาจารย์ ธนา หงษ์สุวรรณ และอาจารย์อัครเดช วัชระภูพงษ์ อาจารย์ประจำห้องวิจัย ISAG(Information Security Advisory Group) และเป็นอาจารย์ที่ปรึกษาโครงการ ที่ให้คำแนะนำช่วยเหลือเอาใจใส่ด้วยดีเสมอมา ขอบคุณเพื่อนๆทุกคนที่ให้ความช่วยเหลือในทุกเรื่อง ขอขอบคุณด้วยใจจริง

ขอขอบคุณศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติสำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีซึ่งได้ให้การสนับสนุนด้านเงินทุนแก่โครงงาน โปรแกรมควบคุมคอมพิวเตอร์เครื่องลูกข่ายระยะไกล โดยได้รับเงินทุนสนับสนุนจากโครงการแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย ครั้งที่ 5 ประเภทโปรแกรมเพื่องานการพัฒนาวิทยาศาสตร์และเทคโนโลยี และขอกราบขอบคุณมา ณ ที่นี้ด้วย

นาถรีย์ ยีตาหวี
ประสิทธิ์ อุทาเลิศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
ABSTRACT	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VII
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและความเป็นมา	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ขอบเขตของงานวิจัย	1
1.4 ปัญหาและประโยชน์ที่เป็นเหตุผลที่ควรพัฒนาโปรแกรม	2
บทที่ 2 การเขียนโปรแกรมเน็ตเวิร์กโดยใช้ วินซ็อก (Winsock)	3
2.1 ซ็อกเก็ตคืออะไร	3
2.2 ตัวอย่างการสื่อสารด้วยซ็อกเก็ต	3
2.3 ชนิดของซ็อกเก็ต	4
2.4 วินซ็อก	5
บทที่ 3 การเข้ารหัสข้อมูล	13
3.1 มาตรฐานการเข้ารหัสข้อมูล	13
3.2 การเข้ารหัสแบบ DES (Data Encryption Standard)	13
3.2.1 การเรียงบิตใหม่ (Initial Permutation)	14
3.2.2 การเข้าฟังก์ชันการคำนวณ	18
3.3 การเข้ารหัสแบบ 3DES (Triple Data Encryption Standard)	20
บทที่ 4 การจัดการกับวินโดวส์	22
4.1 เมสเซจ	22
4.1.1การสร้างและการประมวลผลเมสเซจ	22
4.1.2 การแปลและตีความเมสเซจ	23
4.1.3 การตรวจสอบเมสเซจ	24
4.1.4 การส่งและส่งผ่านเมสเซจ	24
4.1.5 ฟังก์ชั่นที่ใช้กับเมสเซจ	25
4.2 ช่องเกี่ยวโยง (Hook)	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้าที่

บทที่ 5 RFB Protocol 28

5.1 อาร์เอฟบีโพรโตคอล (RFB Protocol) 28

5.2 โพรโตคอลของการแสดงภาพ 29

5.3 อินพุตโพรโตคอล 29

5.4 การนำเสนอของพิกเซลเดทา 29

5.5 เมสเซจโพรโตคอล 32

บทที่ 6 การออกแบบและการพัฒนาซอฟต์แวร์ 45

6.1 การออกแบบซอฟต์แวร์ 45

6.2การพิสูจน์ตน 46

6.3การตรวจสอบสถานะของเครื่องลูกข่าย 47

6.4โปรแกรมควบคุมเครื่องลูกข่ายผ่านกราฟิกโหมด 47

6.4.1หลักการ 47

6.4.2การทำงาน 47

6.5โปรแกรมควบคุมเครื่องลูกข่ายผ่านเท็กซ์โหมด 48

6.5.1หลักการ 48

6.5.2การทำงาน 48

6.6 โปรแกรมสนทนา 49

6.6.1หลักการ 49

6.6.2การทำงาน 49

6.7 โปรแกรมโอนย้ายข้อมูล 50

6.7.1หลักการ 50

6.7.2การทำงาน 50

6.8 การออกแบบส่วนติดต่อผู้ใช้งาน 51

6.8.1 ส่วนติดต่อกับผู้ใช้งานฝั่งเครื่องแม่ข่าย 51

6.8.2ส่วนติดต่อกับผู้ใช้งานในฝั่งเครื่องลูกข่าย 51

บทที่ 7 ผลการทดสอบ 54

7.1 การทดสอบประสิทธิภาพของระบบ 54

7.2 เริ่มต้นการทดสอบโปรแกรม 54

7.2.1 ทดสอบการทำงานของโปแกรมควบคุมผ่านกราฟิก 56

7.2.2 ทดสอบการทำงานของโปรแกรมควบคุมผ่านเท็กซ์โหมด 57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้าที่

7.2.3 ทดสอบการทำงานของโปแกรมโอนย้ายข้อมูล 58
7.2.4 ทดสอบการทำงานของโปรแกรมสนทนา 58
7.3 ปัญหาที่เกิดขึ้นขณะทดสอบ 58
บทที่ 8 บทวิจารณ์และสรุป 59
8.1 วิจารณ์โครงงาน 59
8.2 สรุปผลโครงงาน 59
บรรณานุกรม 60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้าที่

รูปที่ 2-1 การสื่อสารโดยใช้ซ็อกเก็ต 4
รูปที่ 2-2 การเซตอัพการสร้างและการใช้ซ็อกเก็ต 6
รูปที่ 2- 3 แสดงการเพิ่ม components 7
รูปที่ 2-4 แสดง Properties Winsock Dialog 8
รูปที่ 3.1 กระบวนการเข้ารหัสด้วย DES 14
รูปที่ 3-2 การทำงานในส่วนของฟังก์ชัน F 18
รูปที่ 3-3 การคำนวณในส่วนของหมายเลขกุญแจ 19
รูปที่ 3-4 กระบวนการเข้าและถอดรหัสด้วย 3DES 21
รูปที่ 5-1 การเชื่อมต่อของโพรโตคอล 28
รูปที่ 6-1 สถานะการทำงานของโปรแกรม 45
รูปที่ 6-2 ส่วนประกอบหลักของโปรแกรม 46
รูปที่ 6-3 แสดงขั้นตอนการตรวจสอบรหัสผ่าน 46
รูปที่ 6-4 แสดงการทำงานของการควบคุมเครื่องลูกข่ายกราฟิกโหมด 48
รูปที่ 6-5 แสดงการทำงานของการควบคุมเครื่องลูกข่ายผ่านเท็กซ์โหมด 49
รูปที่ 6-6 แสดงการทำงานของโปรแกรมสนทนา 50
รูปที่ 6-7 แสดงการทำงานของการถ่ายโอนข้อมูล 50
รูปที่ 6-8 แสดงหน้าจอของส่วนแสดงสถานะ 51
รูปที่ 6-9 แสดงเมนูการจัดการเครื่องลูกข่าย 52
รูปที่ 6-10 แสดงเมนูการควบคุมเครื่องลูกข่าย 53
รูปที่ 6-11 แสดงส่วนติดต่อผู้ใช้ฝั่งเครื่องลูกข่าย 53
รูปที่ 7-1 แสดงสถานะเริ่มต้นและแสดงการเพิ่มเครื่องลูกข่าย 55
รูปที่ 7-2 การกำหนดรหัสผ่านและเซ็ตค่าต่างๆ 55
รูปที่ 7-3 แสดงการทำการควบคุม 56
รูปที่ 7-4 แสดงการใช้ทรัพยากรของโปรแกรมเมื่อมีการเรียกเครื่องลูกข่าย 20 เครื่อง 57
รูปที่ 7-5 แสดงคอมมานด์เชลล์ที่หลังจากเชื่อมต่อแล้วลิสต์
ที่แสดงเป็นรายการของเครื่องปลายทาง 57
รูปที่ 7-6 แสดงหน้าต่างสำหรับการการโอนย้ายไฟล์ 58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้าที่

ตารางที่ 1-1 ข้อแตกต่างระหว่างโครงงานชิ้นนี้และผลิตภัณฑ์ในตลาด 2
ตารางที่ 3-1 ค่าการสับเปลี่ยนตำแหน่งของบิตในตาราง IP (Initial Permutation) 15
ตารางที่ 3-2 ค่าการสับเปลี่ยนตำแหน่งของบิตในตาราง PC-1 (Permuted Choice 1) 15
ตารางที่ 3-3 จำนวนครั้งที่ทำการเลื่อนตำแหน่งในแต่ละรอบตาราง LS (Left Shift) 15
ตารางที่ 3-4 ค่าการสับเปลี่ยนตำแหน่งของบิตในตาราง PC-2 (Permuted Choice 2) 16
ตารางที่ 3-5 ค่าการสับเปลี่ยนตำแหน่งของบิตในตาราง E (Expansion) 16
ตารางที่ 3-6 ค่าการสับเปลี่ยนตำแหน่งของบิตในตาราง P (Permutation) 16
ตารางที่ 3-7 การหาค่าในตาราง S (Substitution) 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันระบบเน็ตเวิร์กภายในองค์กรได้รับความนิยมอย่างแพร่หลายผู้ใช้งาคอมพิวเตอร์ต่างมีการใช้งานที่หลากหลาย อาจมีผู้ใช้ส่วนหนึ่งที่ใช้งานผิดเป้าหมายซึ่งก่อให้เกิดปัญหาตามมามากมาย จึงเป็นปัญหาอย่างยิ่งของผู้ดูแลระบบในการดูแลความเป็นระเบียบของการใช้งานระบบในการควบคุมความเป็นระเบียบของเครือข่ายนั้นๆ ผู้ดูแลระบบต้องรู้ว่าผู้ใช้งานคอมพิวเตอร์อยู่นั้นผิดระเบียบการใช้งานขององค์กรหรือไม่หรือใช้คอมพิวเตอร์ทำอะไรที่มีความเสี่ยงต่อความมั่นคงของระบบ หรือก่อให้เกิดปัญหาต่อระบบหรือไม่แล้วจึงตัดสินใจดำเนินการต่อไปว่า จะแจ้งเตือนหรือระงับการให้บริการ

เราจึงนำเสนอโครงงานชิ้นนี้ เพื่อใช้เป็นเครื่องมือช่วยเหลือของผู้ดูแลระบบเพื่อให้สามารถดูแลระบบเน็ตเวิร์กได้สะดวกมากยิ่งขึ้นพร้อมทั้งเพิ่มเครื่องมืออื่นๆ สำหรับเพิ่มความสะดวกในการใช้งานทำให้การดูแลระบบง่ายขึ้นโดยมีประสิทธิภาพใกล้เคียงกับซอฟต์แวร์ที่ผลิตจากต่างประเทศ

1.2 วัตถุประสงค์ของงานวิจัย

1.เพื่อให้ผู้ดูแลระบบใช้เป็นเครื่องมือในการติดตามพฤติกรรมการใช้งานคอมพิวเตอร์ลูกข่าย

2.เพื่อเพิ่มความสะดวกแก่ผู้ดูแลระบบในการแก้ปัญหาของเครื่องลูกข่าย

3.เพื่อลดปัญหาการดูแลเครือข่ายไม่ทั่วถึงของผู้ดูแลระบบ

4.เพื่อเพิ่มทางเลือกแก่ผู้ดูแลระบบในการเลือกใช้เครื่องมือที่ช่วยให้การดูแลระบบเน็ตเวิร์กง่ายยิ่งขึ้น

1.3 ขอบเขตของงานวิจัย

1.พัฒนาโปรแกรมด้วยไมโครซอฟท์วิชวลเบสิก 6.0 บนระบบปฏิบัติการวินโดวส์เอ็กซ์พี

2.พัฒนาบนเครื่องคอมพิวเตอร์แบบ ไอบีเอ็ม คอมแพตทิเบิล

3.ทำการจับหน้าจอของเครื่องลูกข่ายที่ลงโปรแกรมฝั่งลูกข่ายไว้ได้

4.สามารถจับหน้าจอเครื่องลูกข่ายได้ครั้งละหลายเครื่อง

5.ทำการควบคุมเครื่องลูกข่ายที่อยู่ในการควบคุมได้ทั้งกราฟิกโหมดและเท็กซ์โหมด

6.ทำการรับส่งไฟล์ระหว่างเครื่องของผู้ดูแลระบบกับเครื่องลูกข่ายได้

7.ทำการรับส่งไฟล์ระหว่างเครื่องลูกข่ายกับเครื่องลูกข่ายได้

8.เครื่องลูกข่ายสามารถติดต่อกับผู้ดูแลระบบผ่านทางโปรแกรมสนทนา

9.เข้ารหัสข้อมูลแบบง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ปัญหาและประโยชน์ที่เป็นเหตุผลที่ควรพัฒนาโปรแกรม

เนื่องจากการใช้งานคอมพิวเตอร์ภายในองค์กรนั้น อาจมีผู้ใช้บางคนใช้ไปในทางที่ไม่ก่อให้เกิดประโยชน์ต่อองค์กร เช่นอาจใช้เวลาในช่วงเวลาทำงานเข้าดูเว็บไซต์ที่ไม่เหมาะสม หรือเมื่อเครื่องลูกข่ายอาจเกิดความสับสนในการใช้งานและต้องการความช่วยเหลือจากผู้ดูแลระบบ หากเป็นในกรณีของการตรวจสอบการใช้งานอาจจะไม่สะดวกนักในการที่จะทำการตรวจสอบโดยการเดินตรวจผู้ใช้ทีละเครื่องซึ่งบางครั้งอาจไม่ได้ข้อมูลตามความเป็นจริง เครื่องมือชิ้นนี้จึงเป็นทางเลือกหนึ่งในการดูแลระบบเครือข่ายของผู้ดูแลระบบให้สามารถดูการใช้งานเครื่องลูกข่ายได้ โดยไม่ให้ผู้ใช้สังเกตเห็นทำให้ได้ข้อมูลที่เป็นจริงมากขึ้น ในกรณีที่ต้องช่วยผู้ใช้งาน ในการแก้ปัญหาผู้ดูแลระบบก็สามารถแก้ปัญหาจากหน้าจอเครื่องควบคุมของผู้ดูแลระบบที่ได้ใช้โครงงานชิ้นนี้ได้เลย ไม่ต้องเดินไปทำหน้าเครื่องลูกข่ายที่มีปัญหาทำให้สะดวกมากยิ่งขึ้น

	RemoteAdmin2.1	PCAnywhere10.5	VNC3.3.4	CRCS
Monitoring 1:1	Y	Y	Y	Y
Monitoring 1:m	Y	N	Y	Y
GraphicControl	Y	Y	Y	Y
Port Changable	Y	Y	Y	Y
Command line Control	Y	N	Y	Y
File Transfer	Y	Y	N	Y
Chat	N	N	N	Y
Cost	$35.00	$24.95	FREE	FREE

ตารางที่ 1-1 ข้อแตกต่างระหว่างโครงงานชิ้นนี้และผลิตภัณฑ์ในตลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
การเขียนโปรแกรมเน็ตเวิร์กโดยใช้ วินซ็อก(Winsock)

การเขียนโปรแกรมเน็ตเวิร์ก (ทีซีพี/ไอพี) โดยใช้ Winsock

ในบทนี้เราจะนำเสนอการเขียนโปรแกรมผ่านเน็ตเวิร์กระหว่างฝั่งไคลเอนต์กับฝั่งเซิร์ฟเวอร์ด้วยการใช้โพรโตคอล ทีซีพี/ไอพี ใน Windows Socket API หรือเรียกสั้นๆว่า Winsock

2.1 ซ็อกเก็ต (socket) คืออะไร

ซ็อกเก็ตถูกนิยามหรือกำหนดไว้ว่าเป็นคู่ของการสื่อสารหรือคู่ของโพรเซส(หรือ เธรด)โดยที่การสื่อสารบนเน็ตเวิร์กใช้คู่ของซ็อกเก็ตสำหรับแต่ละโพรเซสสำหรับซ็อกเก็ตประกอบไปด้วยไอพีแอดเดรสกับหมายเลข Port (Port Number)

โดยทั่วๆไป ซ็อกเก็ตใช้สถาปัตยกรรมไคลเอนต์เซิร์ฟเวอร์ เซิร์ฟเวอร์จะรอการเข้ามาตามการร้องขอของไคลเอนต์โดยการฟังที่พอร์ตเฉพาะเมื่อการร้องขอได้รับเซิร์ฟเวอร์ก็จะยอมรับการเชื่อมต่อจากซ็อกเก็ตไคลเอนต์เพื่อให้สมบูรณ์ในการเชื่อมต่อ

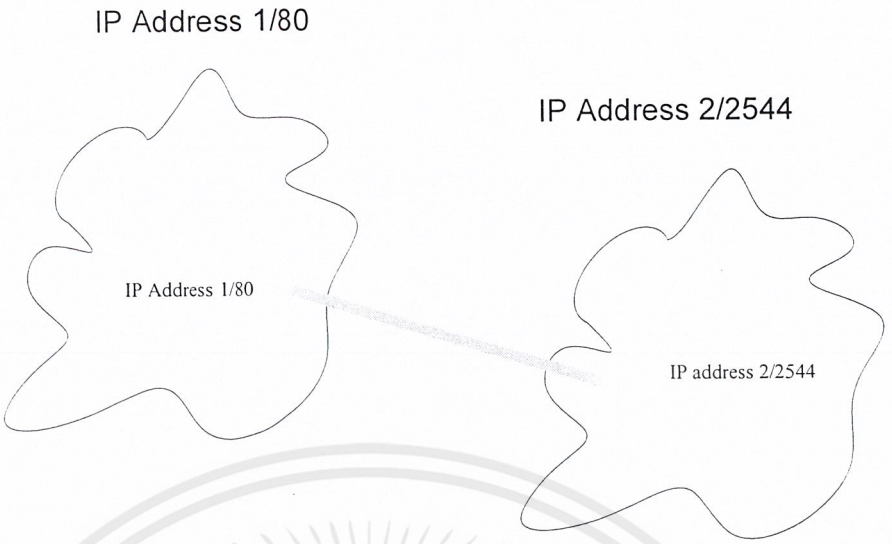
เซิร์ฟเวอร์ที่สร้างการบริการเฉพาะ เช่น เทลเน็ต, เอฟทีพี, เมล์ และ เอชทีทีพี จะฟัง(listen)ที่พอร์ตมีชื่อ เช่น เซิร์ฟเวอร์เทลเน็ตจะฟังที่พอร์ต 23, เซิร์ฟเวอร์เอฟทีพีจะฟังที่พอร์ต 21 หรือ เซิร์ฟเวอร์เอชทีทีพีจะฟังที่พอร์ต 80 เป็นต้น

หมายเลขพอร์ตทั้งหมดที่ต่ำกว่า 1024 จะถูกพิจารณาว่าเป็นพอร์ตมาตรฐานที่เราสามารถใช้พอร์ตเหล่านี้เพื่อการสร้างบริการตามมาตรฐานได้

2.2 ตัวอย่างการสื่อสารด้วยซ็อกเก็ต

เมื่อเธรดฝั่งไคลเอนต์เริ่มต้นการขอร้องสำหรับการเชื่อมต่อจะถูกกำหนดพอร์ตโดยโฮสต์คอมพิวเตอร์ (Host omputer) พอร์ตนี้เป็นหมายเลขใดๆก้ได้ที่มากว่า 1024 ตัวอย่าง เช่น ถ้าไคลเอ็นต์บนโฮสต์ A มีไอพีแอดเดรส 2 ต้องการที่จะสร้างการเชื่อมต่อกับเซิร์ฟเวอร์ เอชทีทีพี (ซึ่งฟังที่พอร์ต 80)ที่มีไอพีแอดเดรส 1 โฮสต์ A จะถูกกำหนดพอร์ต 2544 และที่ฝั่งเซิร์ฟเวอร์จะเป็นพอร์ต 80 สถานการณ์นั้นสามารถแสดงได้ดังรูป แพ็กเก็ตไปมาระหว่างโฮสต์ทั้งสองจะถูกส่งไปยังเธรดที่เหมาะสม ซึ่งขึ้นอยู่กับหมายเลขพอร์ตปลายทางด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-1 การสื่อสารโดยใช้ซ็อกเก็ต

การเชื่อมต่อทั้งหมดเป็นคุณสมบัติเฉพาะดังนั้นถ้าโพรเซสอื่นๆบนโฮสต์ A ต้องการสร้างการเชื่อมต่ออื่นๆกับเซิร์ฟเวอร์เอชทีทีพีเดียวกัน เซิร์ฟเวอร์จะกำหนดหมายเลขพอร์ตที่มากกว่า1024และต้องไม่เท่ากับพอร์ต 2544 (เนื่องจากถูกใช้ไปแล้ว) การทำอย่างนี้เพื่อให้แน่ใจว่าการเชื่อมต่อทั้งหมดประกอยด้วยคู่ที่เป็นยูนิค (unique) ของซ็อกเก็ตหรือเป็นสิ่งที่ไม่ซ้ำกับการเชื่อมต่ออื่นๆของซ็อกเก็ต

โดยปกติแล้วเซิร์ฟเวอร์จะมีหลายๆการร้องขอที่พร้อมกันจะต้องใช้ระยะเวลาหนึ่งที่ไคลเอ็นต์ต้องรอคอยเพื่อที่จะถูกบริการโดยเซิร์ฟเวอร์เธรดเดียว ซึ่งจะไม่สามารถรับได้

เพื่อแก้ไขสถานการณ์นี้เซิร์ฟเวอร์ต้องจัดการการขอร้องที่พร้อมๆกันโดยการกำหนดเธรดแยกออกมาเพื่อบริการแต่ละการร้องขอที่เข้ามา ตัวอย่างเช่น เซิร์ฟเวอร์ เอชทีทีพี ที่ไม่ว่างจะกำหนดเธรดแยกออกมาเพื่อบริการแต่ละการร้องขอสำหรับเว็บเพจ

2.3 ชนิดของซ็อกเก็ต

ชนิดของซ็อกเก็ตมีอยู่สามชนิด คือ

- Connection-Oriented Socket
- Connectionless Socket
- Raw Socket

2.3.1 Connection-Oriented Socket

เป็นซ็อกเก็ตสำหรับการเชื่อมต่อแบบต่อเนื่องที่จะยอมอนุญาตให้โพรเซสนั้นๆสามารถเชื่อมต่อกับโพรเซสจากระยะไกล (Remote) ซึ่งใช้โพรโตคอล ทีซีพี (Transmittion Control Protocol) ดังนั้นด้วยวิธีการนี้ทำให้ข้อมูลมีความเชื่อถือได้เมื่อการเชื่อมต่อได้เกิดขึ้นโพรเซสก็จะมีการส่งข้อมูลกลับไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จนกระทั่งฝั่งใดฝั่งหนึ่งหรือฝั่งอื่นๆปิดการเชื่อมต่อชนิดของซ็อกเก็ตแบบนี้บางครั้งเรียกว่า สตรีมซ็อกเก็ต (Stream socket) ทั้งเอฟทีพีและ เอชทีทีพีใช้ซ็อกเก็ตแบบนี้ในการเชื่อมต่อช่องการสื่อสาร

2.3.2 Connectionless Socket

ซ็อกเก็ตแบบนี้บางที่จะเรียกว่า เดทาแกรมซ็อกเก็ต ซึ่งซ็อกเก็ตแบบนี้เป็นการเชื่อมต่อแบบไม่ต่อเนื่อง และนำมาใช้ประโยชน์ในการส่งเมสเซจสั้นๆซึ่งไม่สนับสนุนส่วนหัว ดังนั้นจึงพิจารณาการเชื่อมต่อประเภทนี้เป็นแบบเชื่อถือไม่ได้ ซึ่งก็คือ การไม่รับประกันข้อมูลที่ส่งออกไป ไม่เหมือนกับซ็อกเก็ตการเชื่อมต่อแบบต่อเนื่องที่ซ็อกเก็ตปลายทางถูกตรวจสอบเมื่อแพ็กเก็ตถูกส่งออกไป

ซ็อกเก็ตแบบไม่ต่อเนื่อง เปรียบเสมือนกับการบริการของไปรษณีย์ที่ผู้ส่งจดหมายไปตามที่อยู่แล้วใส่ในกล่องรับจดหมาย ผู้ส่งจะไม่ทราบว่าผู้รับได้รับจดหมายหรือไม่ ซ็อกเก็ตแบบนี้นิยมใช้กันในเซิร์ฟเวอร์ DNS (Domain Name System) ที่ใช้ซ็อกเก็ตเดทาแกรมในการตอบสนองต่อการร้องขอที่เข้ามามากๆ

นอกจากนี้จะใช้เดทาแกรมซ็อกเก็ตในการกระจาย (Broadcast) เมสเซจ หรือ Multicast เพื่อไปยังปลายทางหลายๆแห่งพร้อมกัน ซึ่งเหมือนกับการกระจายเสียงวิทยุ หรือโทรทัศน์

2.3.3 Raw Socket

เป็นซ็อกเก็ตที่อนุญาตให้การเข้าถึงโพรโตคอล Transport Raw Socket ยังสามารถนำมาใช้เพื่อจัดการข้อมูลของ “ไอพีเฮดเดอร์” นอกจากนี้แล้วการใช้ซ็อกเก็ตชนิดนี้ยังต้องการความรู้อย่างมากเกี่ยวกับโครงสร้างโพรโตคอลพื้นฐาน

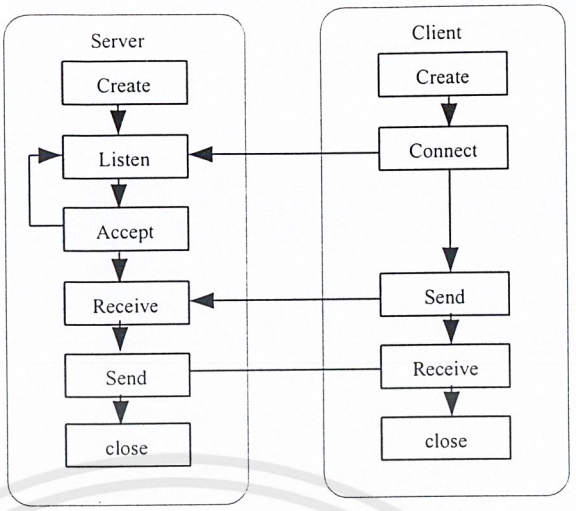
2.4 วินซ็อก (Winsock)

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

วินซ็อก เป็นเน็ตเวิร์กแอพพลิชันโปรแกรมมิ่งอินเตอร์เฟสไม่ใช่โพรโตคอล ซึ่งวินซ็อกนั้นมีรากฐานเดียวกับซ็อกเก็ตตระกูล บีเอสดี (Berkeley Software Distribution) เวอร์ชัน 4.3 จากมหาวิทยาลัยแคลิฟอร์เนีย วิทยาเขต เบอร์เกลีย์ รายละเอียดได้รวมเอาทั้งรูทีนซ็อกเก็ตสไตล์ บีเอสซี (BSC) และการขยายสเปกมาใช้กับวินโดวส์

การใช้วินซ็อกอนุญาตให้แอพพลิเคชันของเราทำการติดต่อสื่อสารข้ามเน็ตเวิร์กใดๆก็ได้ที่กระทำกับ วินซ็อก เอพีไอ แพลตฟอร์ม วิน32 ซึ่งวินซ็อกได้ให้เธรดที่ปลอดภัย (thread safty)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-2 การเซตอัพการสร้างและการใช้ซ็อกเก็ต

2.4.1 MS Winsock Control 6

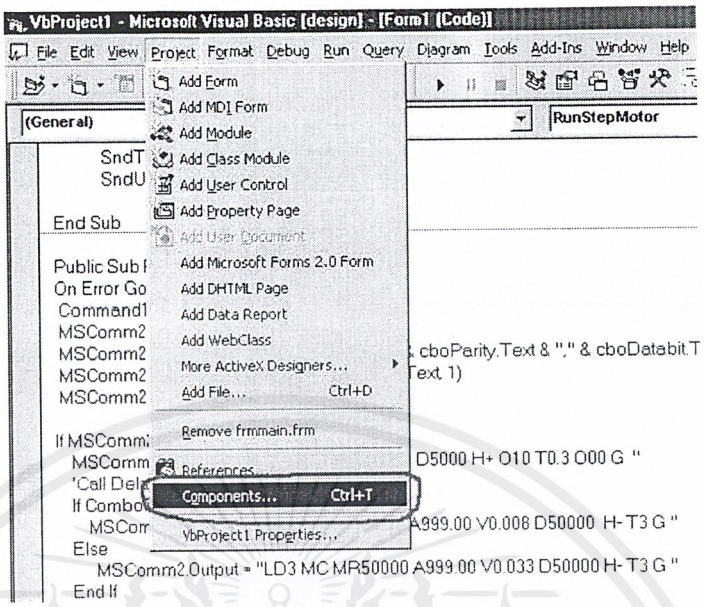
เขียนโปรแกรมบนระบบ Windwos ให้สามารถติดต่อสื่อสารบนระบบเครือข่ายได้ ถ้าเป็นเมื่อก่อนบนระบบปฏิบัติการ DOS นั้นคงจะยากทีเดียว แต่เดี๋ยวนี้เครื่องคอมพิวเตอร์อะไรๆก็ Windows กันแล้วเนื่องจากมีเครื่องมือ(Tool)ที่ช่วยให้เราสามารถเขียนโปรแกรมติดต่อผ่านระบบเครือข่าย ที่เราจะกล่าวถึงก็คือMS Winsock Control 6 เป็นเครื่องมือที่อำนวยความสะดวกสำหรับการเขียนโปรแกรมติดต่อผ่านระบบเครือข่าย โดยโปโตคอล TCP/IP ซึ่งเราจะทดลองเขียนด้วย Visual Basic หรือถ้าท่านใดถนัด C/C++ ก็ใช้ Visual C++ ท่านใดถนัด ปาสคาล ก็ใช้ Delphi ที่เลือก VB เพราะคิดว่าคงจะอธิบายได้ง่ายแล้วเข้าใจเร็วซึ่งก็ไม่ซับซ้อนมากนัก ส่วนนอกเหนือจากการที่กล่าวมาแล้วเกี่ยวกับ Winsock Control แล้วยังมี API ที่มีนักพัฒนาเขียนขึ้นมาหลายตัวทีเดียว แต่ดูๆแล้ว Winsock จะใช้งานง่ายกว่า

จากตอนที่แรกได้กล่าวเนื้อหาเบื้องต้นและโดยรวมเกี่ยวกับการอินเทอร์เฟสฮาร์ดแวร์ผ่านระบบเครือข่ายโดยมีโพรโตคอลTCP/IP เป็นตัวช่วยในการสื่อสารระหว่างคอมพิวเตอร์ด้วยกันได้ ทั้งนี้ในทางการเขียนโปรแกรมเพื่อติดต่อบนระบบ Windows นั้นก็ได้อำนวยความสะดวกให้เราสามารถที่จะเขียนโปรแกรมติดต่อกับระบบเครือข่ายได้ก็คือ MS Winsock Control ที่มีอยู่ในชุด Visual Studio ในเนื้อหานี้เราจะใช้เขียนโปรแกรมด้วย Visual Basic เพราะจะเข้าใจง่าย,สะดวก,และรวดเร็วต่อการทำความเข้าใจ

ก่อนอื่นก็เปิดโปรแกรม VB ขึ้นมาแล้วสร้าง Project Standard EXE ขึ้นใหม่ จากนั้นเราจะต้องเพิ่ม Winsock Control ก่อน โดยคลิกที่เมนู

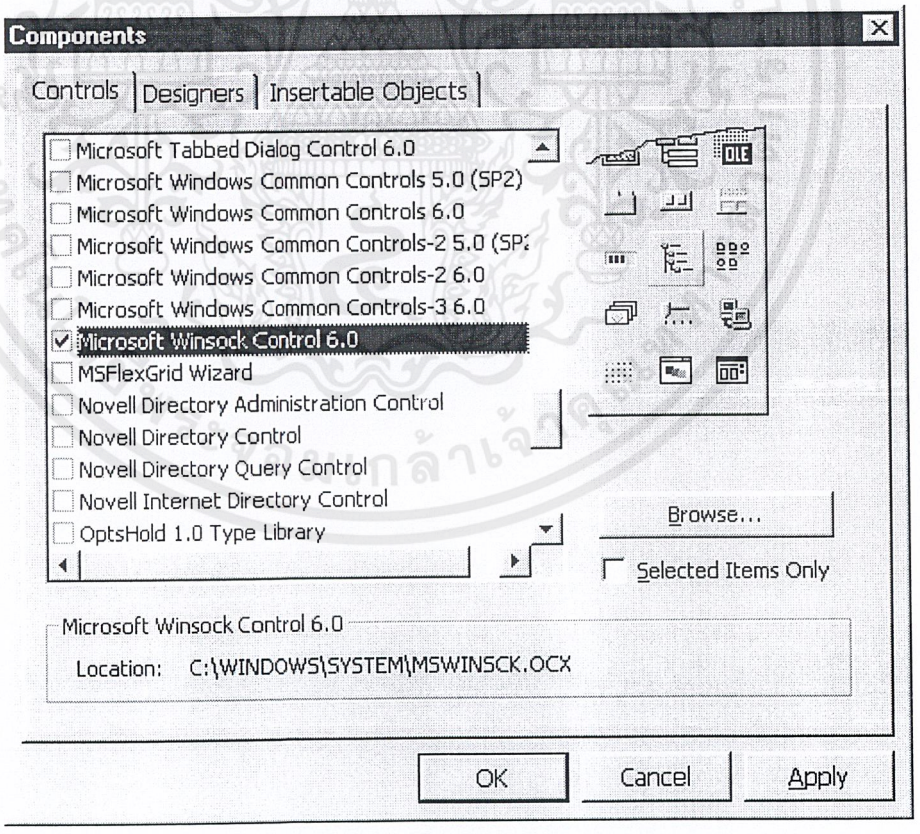
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Add Winsock Control



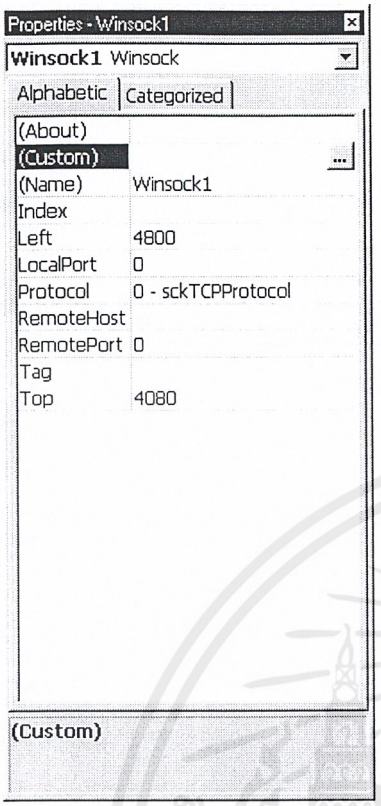
รูปที่ 2- 3 แสดงการเพิ่ม components

ให้เลือก Control ชื่อ Microsoft Winsock Control 6.0 (สำหรับ VB6)



รูปที่ 2-4 แสดง Properties Winsock Dialog

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Properties	Detail
Name	ชื่อของคอนโทรลที่เราสามารถตั้งได้
LocalPort	Portsที่เราจะกำหนดในการติดต่อ
Protocal	เลือกชนิดของโพรโตคอล ปกติเราใช้ 0-TCPProtocal
RemoteHost	IP Address ของเครื่องที่ใช้ติดต่อ
RemotePort	Port Address ของเครื่องที่ใช้ติดต่อ

Winsock Procedureในส่วนของ Windows

Code Object -->Winsock มีProcedure สำหรับกระทำติดต่อสื่อสารทั้งทางฝ่าย Server และ Client เราขอกล่าวอย่างนี้เพื่อความเข้าใจที่เหมือนกันดังจะกล่าวในที่นี้คือ

-Close คือ เหตุการณ์เมื่อมีหยุดหรือยกเลิกการติดต่อสื่อสารของฝ่าย Server หรือClient โดย Function Winsock.Close ซึ่งจะเราสามารถจะใช้ตรวจสอบฝ่ายตรงข้ามว่ามีการติดต่ออยู่หรือ ไม่ โดยอาจจะใส่ Massage เตือนเป็นต้น

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

-ConnectionRequest เป็นเหตุการณ์เมื่อฝ่าย Client ส่งสัญญาณติดต่อกลับมายัง Server Procedure ส่วนนี้ก็จะทำงานพร้อมกับค่า requestID As Long ซึ่งเป็นหมายที่ Gen ขึ้นมาในระบบค่านั้นจะไม่เหมือนเดิม โดยจะให้ฝ่าย Server รับรู้ว่าใช้ ID จากคอนโทรลตัวใดเพื่อจะได้สื่อสารถึงต้อง

-DataArrival เหตุการณ์นี้เกิดขึ้นเมื่อมีการส่งข้อมูลระหว่าง Server และ Client Porcedure นี้ก็จะทำงานขึ้นมา พร้อมกับค่าจำนวน bytesTotal As Long ที่รับเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

- SendProgress จะเกิดขึ้นในขณะที่มาการส่งข้อมูลอยู่เหตุการณ์นี้ก็จะทำงานเมื่อส่งข้อมูลหมดแล้วก็จะส่งผลทำให้เกิด Event SendComplete

-SendComplete เหตุการณ์เมื่อมีการส่งข้อมูลออกไปยังฝ่ายตรงข้ามเสร็จเรียบร้อยแล้ว

2.4.2 Winsock Properties & Events

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

Close เป็นการส่งสัญญาณยกเลิกการติดต่อระหว่างกัน จะเป็นฝ่าย Server หรือ Client ก็ได้ ที่จะใช้ Function นี้ จากนี้จะทำให้ Procedure close ในฝ่ายตรงข้ามทำงาน

Connect เป็นการส่งสัญญาณว่าตอนนี้ทำการติดต่อเรียบร้อยแล้ว ซึ่งจะส่งผลให้ Procedure ฝ่ายตรงข้ามทำงาน

Getdata เป็นการรับข้อมูลเมื่อฝ่ายตรงข้ามส่งมาโดยประโยคคำสั่งนี้จะอยู่ในส่วนของ Procedure DataArrival เนื่องจากเป็นเหตุการณ์ที่การกระทำขณะเมื่อฝ่ายตรงข้ามส่ง ข้อมูลเข้ามา

Listen การกระทำที่จะคอยตรวจสอบสัญญาณที่ส่งไปว่าฝ่ายตรงข้ามตอบรับการร้องขอการติดต่อ

LocalHostName คำสั่งนี้จะส่งชื่อของ Computer name ของเครื่องนั้นๆ

```
Debug.Print  Winsock1. LocalHostName
```

LocalIP คำสั่งนี้จะทำการส่งหมายเลข IP Address

```
Debug.Print  Winsock1.LocalIP
```

LocalPort คำสั่งที่จะส่งค่าของหมายเลขในการติดต่อ TCP/IP ของเครื่องนั้นๆ

```
Debug.Print  Winsock1.LocalPort
```

RemoteHost กำหนดหรือคืนค่าชื่อ Computer name ของเครื่องที่จะทำการติดต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 ขณะที่ใช้ติดต่อระหว่างอยู่ โดยอาจจะใช้ตรวจสถานะโดยค่าคงที่เหล่านี้เช่น sckClosed (มีค่า=0) Socket ปิดการใช้งาน, sckOpen (มีค่า= 1) Socket เปิดใช้งาน หรือ sckError(มีค่า = 9) Socket มีความผิดพลาดเกิดขึ้น เป็นต้น

2.4.3 Basic Winsock Process Connect

จากที่อธิบายรายละเอียดต่างๆเกี่ยวกับ Winsock control มาบ้างแล้วถึงคราวนี้เราจะมาทำการเริ่มการเขียนโปรแกรมเพื่อที่จะติดต่อกับโพรโตคอล TCP/IP กันล่ะซึ่งการติดต่อกันนั้นจะต้องมีทั้งฝั่ง Server และ Client ซึ่งคล้ายว่าจะมี 2 โปรแกรม แต่ความจริงแล้วก็คือโปรแกรมเดียวกันละครับแต่เปิด 2 หน้าต่าง โดยกำหนดว่าฝั่งไหนเป็น Server หรือ Client ก็ได้

เหตุการณ์แรกที่ต้องทำในฐานะที่เป็น ฝ่าย Server คือการตรวจสอบสัญญาณจากฝั่ง Client ในที่นี้ใช้ Event Click ของปุ่มชื่อ cmdListen

```
Private Sub cmdListen_Click()
   Winsock1.LocalPort = txtPortSvr.Text
   Winsock1.Listen
End Sub
```

เหตุการณ์ที่สองในฐานะClient ก็จะตอบรับโดยส่งสัญญาณไปให้กับ Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub cmdConnect_Click()
   Winsock1.RemoteHost ="10.10.0.25"
    Winsock1.RemotePort = "5000"  ' กำหนดหมายเลขอื่นก็ได้ แต่ต้องให้ตรงกันทั้ง 2 ฝ่าย
   Winsock1.Connect
End Sub
```

เมื่อคลิกปุ่ม cmdConnect ตอนนี้ทาง Winsock Control ฝั่ง Sever จะเกิด Event Connect ซึ่งใน Event นี้เราอาจจะใส่ Massage เตือนก็ได้ดังนี้

```
        Private Sub Winsock1_Connect()
   MsgBox "ตอบรับการติดต่อกลับมาแล้ว", vbExclamation, "Chat by MS Winsock Control 6"
End Sub
```

ขึ้นก็เป็นอันว่าทั้ง Server กับ Client สามารถติดต่อสื่อสารกันได้แล้ว เอ้าลองส่ง Data กันหน่อยสิ

```
Private Sub cmdSend_Click()
    Winsock1.SendData "Hi! How are you ? "
End Sub
```

เมื่อฝ่าย Server ส่งData ออกไปที่ Winsock control ฝ่าย Client จะเกิด Event DataArrival เราก็จะรับข้อมูลนั้นด้วยคำสั่ง Getdata โดยเขียนโปรแกรมไว้ใน Event นี้

```
Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
    Dim StrValue As String
    Winsock1.GetData StrValue
End Sub
```

ถ้าหากต้องการยกเลิกการติดต่อก็ใช้คำสั่ง Close เมื่อ Server หรือ Client ใช้คำสั่งนี้ จะทำให้เกิด Event Close ขึ้นกับ Winsock control ของฝั่งนั้น ดังเช่นเราสร้างปุ่ม Close ไว้

```
Private Sub cmdClose_Click()
   Winsock1.SendData "ยกเลิกการติดต่อแล้วครับ"
   Winsock1.Close
End Sub
```

สมมุติว่าเมื่อฝั่ง Server มีการคลิกปุ่มเกิดขึ้นก็จะทำให้ Winsock Control ฝั่ง Client เกิด Event Close โดยอาจจะเขียน code Messageเตือนไว้ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Winsock1_Close()
   MsgBox "ยกเลิกการติดต่อแล้วครับ", vbExclamation, "Chat by MS Winsock     Control 6"
End Sub
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การเข้ารหัสข้อมูล

3.1 มาตรฐานการเข้ารหัสข้อมูล (Data Cryptography)

ไม่ว่าคอมพิวเตอร์จะมีระบบความปลอดภัยอยู่ในระดับ C, B หรือ A ก็ตาม หากว่าข้อมูลที่เก็บอยู่ในเครื่องยังคงอยู่ในรูปข้อมูลปกติ หรือตัวอักษรธรรมดา (Plain Text) ความปลอดภัยของข้อมูลก็ยังต่ำมาก เนื่องจากเมื่อผู้อื่นรู้ชื่อผู้ใช้และรหัสผ่านที่ถูกต้องแล้ว ก็สามารถลักลอบนำข้อมูลที่เก็บไว้นี้ไปใช้ได้ทันที ซึ่งเท่ากับว่าความปลอดภัยของข้อมูลนี้ขึ้นอยู่กับรหัสผ่านเพียงอย่างเดียว ดังนั้นการเข้ารหัสข้อมูล หรือ Data Cryptography จึงเป็นอีกวิธีหนึ่งที่จะป้องกันไม่ให้ข้อมูลอันมีค่าของเราถูกผู้อื่นลักลอบนำไปใช้

การเข้ารหัสข้อมูล คือการนำข้อมูลปกติที่อยู่ในรูปของตัวอักษรธรรมดา มาผ่านการเข้ารหัสทำให้ได้ข้อมูลออกมาในอีกรูปแบบหนึ่ง ซึ่งจะไม่มีความหมาย อ่านไม่ออก และแปลกลับไปหาข้อมูลเดิมไม่ได้หากไม่มีรหัสลับที่ถูกต้อง ดังนั้นถึงแม้จะมีใครแอบก๊อปปี้ข้อมูลที่เข้ารหัสนี้ไปได้ แต่ถ้าไม่รู้รหัสลับก็จะไม่สามารถใช้ประโยชน์จากข้อมูลนั้นได้ ดังนั้นการเข้ารหัสข้อมูลจึงเป็นมาตรการสำหรับการรักษาความปลอดภัยของข้อมูล ที่เพิ่มขึ้นจากระบบความปลอดภัยของคอมพิวเตอร์ ซึ่งการเข้ารหัสข้อมูลจะแบ่งออกเป็น 2 ประเภท คือ Symmetric Cryptography และ Asymmetric Cryptography

3.2 การเข้ารหัสแบบ DES (Data Encryption Standard)

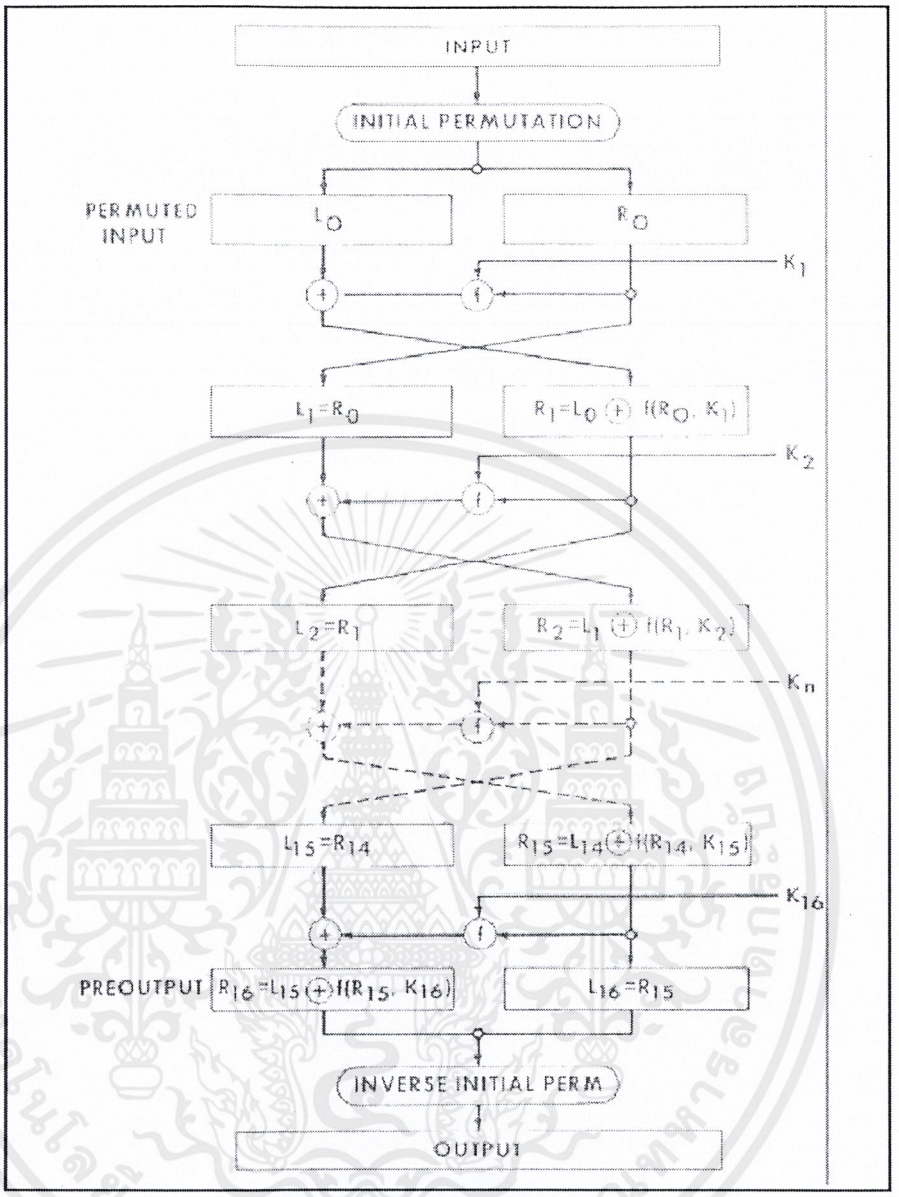
DES คือ การเข้ารหัสข้อมูลโดยใช้อัลกอริทึมทางคณิตศาสตร์ โดยจะใช้ข้อมูลขนาด 64 บิต และกุญแจขนาด 64 บิต (โดย 8 บิตที่เพิ่มเข้ามา เพื่อนำมาใช้ตรวจสอบความผิดพลาด) ซึ่งในการเข้าและถอดรหัสนั้น ได้ใช้ระบบการเข้ารหัสแบบสมมาตร (Symmetric Cryptosystem) คือ ใช้กุญแจตัวเดียวกันในการเข้าและถอดรหัส สำหรับข้อมูลที่นำมาเข้ารหัสเรียกว่า เพลนเท็กซ์ (Plain Text) เมื่อผ่านการเข้ารหัสแล้ว จะเรียกข้อมูลชุดนั้นว่า ไซเฟอร์เท็กซ์ (Cipher Text) ซึ่งเป็นข้อมูลที่ไม่สามารถอ่านหรือตีความได้

ในกระบวนการถอดรหัส จะใช้กุญแจตัวเดียวกันกับที่ใช้ในการเข้ารหัส โดยจะทำกระบวนการย้อนกลับกับกระบวนการเข้ารหัส

การทำงานของอัลกอริทึม DES ได้แบ่งการทำงานออกเป็น 3 ส่วนที่สำคัญ คือ

1. การเรียงบิตใหม่ (Initial Permutation) เพื่อผลิต “permuted input” ซึ่งเป็นข้อมูลที่มีการสลับตำแหน่งจากเดิมไป
2. การเข้าฟังก์ชันการคำนวณ โดยมีกระบวนการทำงานแบบเดิมซ้ำกัน 16 ครั้ง ซึ่งรวมถึงการทำ และ ซึ่งผลลัพธ์ที่ได้ จะได้เป็นข้อมูลขนาด 64 บิตอยู่ชุดหนึ่งที่เรียกว่า PREOUTPUT
3. การเรียงบิตกลับ (Inverse Initial Permutation) ซึ่งจะได้ผลลัพธ์ออกมาเป็นข้อมูลที่ทำการเข้ารหัสแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-1 กระบวนการเข้ารหัสด้วย DES

3.2.1.การเรียงบิตใหม่ (Initial Permutation)

สำหรับขั้นตอนแรกของการเข้ารหัสข้อมูลแบบ DES นั้น เริ่มต้นโดยการสับเปลี่ยนตำแหน่งของบิต ซึ่งจะสับเปลี่ยนตำแหน่งของบิตตามตาราง

ในการใช้ตารางนั้น ตำแหน่งของตารางที่ปรากฏจะแทนตำแหน่งที่บิตนั้นๆอยู่หลังจากที่ทำการสลับตำแหน่งแล้วนั่นเอง โดยที่หมายเลขข้างในคือ หมายเลขที่แทนตำแหน่งข้อมูลก่อนการสลับตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	42	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

ตารางที่ 3-1 ค่าการสับเปลี่ยนตำแหน่งของบิตในตาราง IP (Initial Permutation)

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	1

ตารางที่ 3-2 ค่าการสับเปลี่ยนตำแหน่งของบิตในตาราง PC-1 (Permuted Choice 1)

ลำดับ	จำนวนครั้ง	ลำดับ	จำนวนครั้ง
1	1	9	1
2	1	10	2
3	2	11	2
4	2	12	2
5	2	13	2
6	2	14	2
7	2	15	2
8	2	16	1

ตารางที่ 3-3 จำนวนครั้งที่ทำการเลื่อนตำแหน่งในแต่ละรอบตาราง LS (Left Shift)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

ตารางที่ 3-4 ค่าการสับเปลี่ยนตำแหน่งของบิตในตาราง PC-2 (Permuted Choice 2)

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

ตารางที่ 3-5 ค่าการสับเปลี่ยนตำแหน่งของบิตในตาราง E (Expansion)

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

ตารางที่ 3-6 ค่าการสับเปลี่ยนตำแหน่งของบิตในตาราง P (Permutation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

S_1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

ตารางที่ 3-7 การหาค่าในตาราง S (Substitution)

ตาราง S นั้น จะมีอยู่ 8 ตาราง โดยข้อมูลที่นำมาเข้าในแต่ละตาราง S นั้น จะมีขนาด 6 บิต วิธีการใช้ตาราง S คือ นำค่าของบิตในตำแหน่งที่ 1 และ 6 มาเป็นค่าของส่วนแถว (Row) และค่าของบิตในตำแหน่งที่เหลือคือ 2, 3, 4 และ 5 นั้น นำมาเป็นค่าของส่วนคอลัมน์ (Column) แล้วนำค่าแถวและคอลัมน์มาเทียบในตาราง S ซึ่งจะได้ออกมาเป็นผลลัพธ์ของแต่ละตาราง S ที่มีขนาด 4 บิตออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

49990

3.2.2. การเข้าฟังก์ชันการคำนวณ

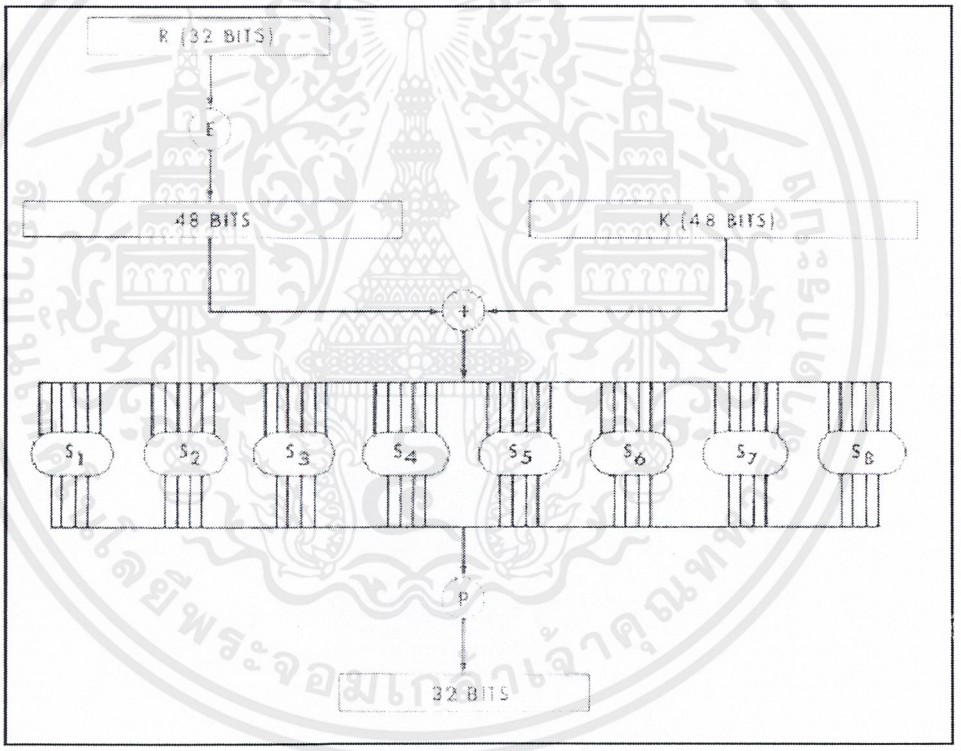
ในการเข้าฟังก์ชันการคำนวณ มีขั้นตอนดังนี้

1. ข้อมูลขนาด 64 บิต จะถูกแบ่งออกเป็น 2 ส่วน คือ ส่วนซ้าย L และส่วนขวา R โดยแต่ละส่วนจะมีขนาดเท่ากันคือ 32 บิต ซึ่งจะกำหนดค่าเริ่มต้นในแต่ละส่วนเป็น L0 และ R0
2. การทำงานของส่วน L และ R จะทำงานทั้งหมด 16 รอบการทำงาน โดยแต่ละรอบการทำงาน จะแทนด้วยตัว n ซึ่ง n จะมีค่าตั้งแต่ 1 ถึง 16 ซึ่งมีสูตรการคำนวณดังนี้

$$L0 = Rn\text{-}1$$

$$Rn = Ln \text{ XOR } f(Rn\text{-}1, Kn)$$

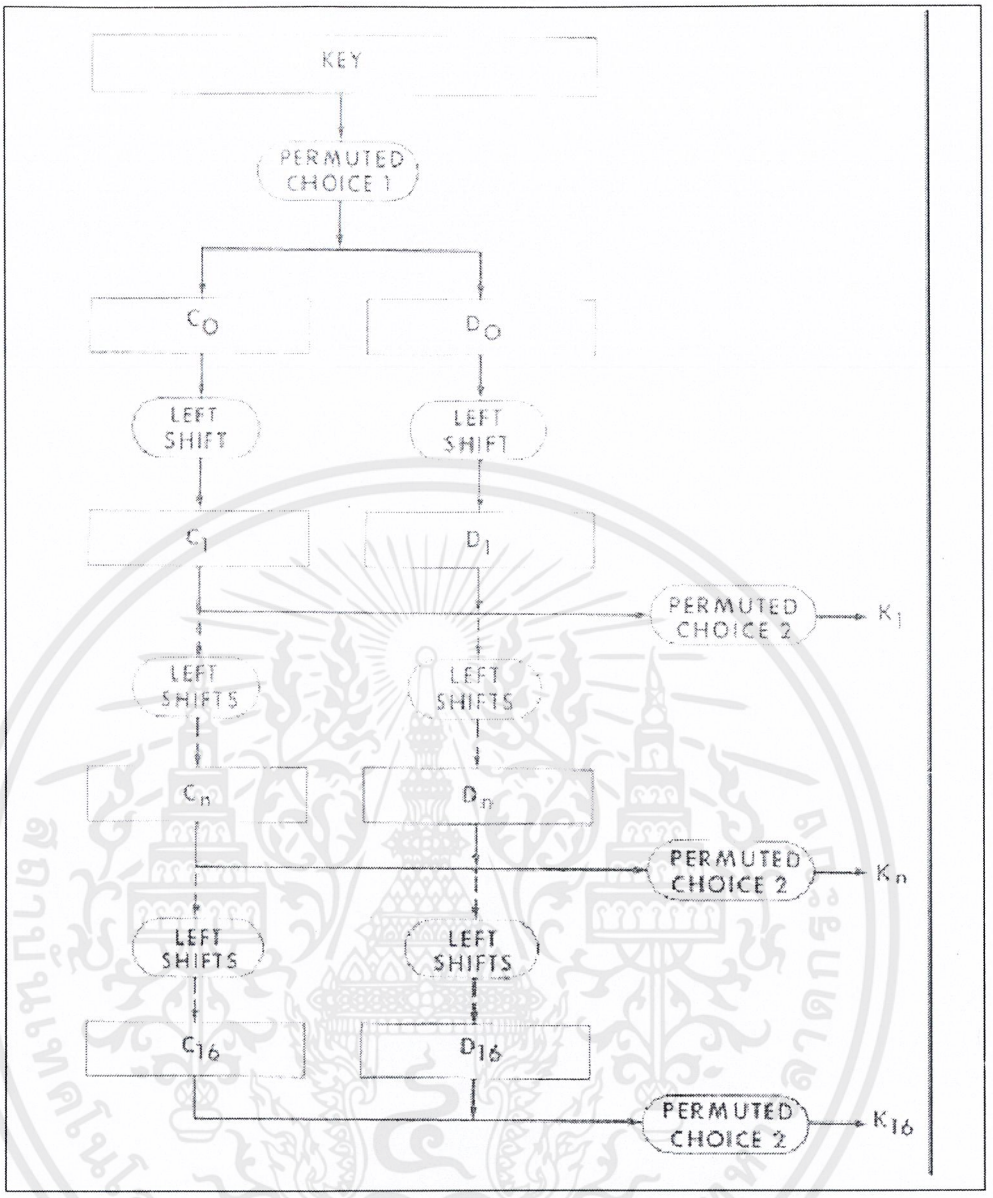
ในส่วนของฟังก์ชัน f จะเป็นการคำนวณเพื่อเพิ่มความยากของการเข้าและถอดรหัส ซึ่งจะถูกเรียกใช้คำนวณทั้งหมด 16 รอบ โดยในแต่ละรอบจะมีวิธีการคำนวณดังรูปที่ 2-8



รูปที่ 3-2 การทำงานในส่วนของฟังก์ชัน F

ในแต่ละรอบของกระบวนการเข้ารหัสข้อมูลด้วยกุญแจจำนวน 16 ชุดนั้น จะใช้ค่าของกุญแจแต่ละชุดที่มีขนาด 48 บิต โดยที่ค่าของกุญแจทั้ง 16 ชุดได้มาจากกุญแจที่เป็นข้อมูลขนาด 64 บิต โดยบิตตำแหน่งที่ 8, 16, ..., 64 เป็นบิตที่ใช้ในการตรวจสอบความผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-3 การคำนวณในส่วนของหมายเลขกุญแจ

ในการทำการสับเปลี่ยนตำแหน่ง Permuted Choice 1 ตามตาราง PC-1 (ตารางที่ ก-2 ในภาคผนวก ก) จะทำการตัดในส่วนของบิตตรวจสอบความผิดพลาดออกแล้ว โดยมีการใช้ข้อมูลเพียงแค่ 56 บิตที่เหลือ

ผลลัพธ์จากการทำตามตาราง PC-1 แล้ว จะนำมาแบ่งออกเป็น 2 ส่วน ส่วนละ 28 บิต เรียกว่า C และ D ซึ่งนำมาใช้นำการหาค่าของกุญแจย่อยแต่ละชุด โดยที่

Ci = LSi(Ci-1)

Di = Lsi(Di-1)

ซึ่ง LS คือ การเลื่อนบิตไปทางซ้าย (Left Shift หรือ Circular Left Shift หรือ Rotation) โดยเลื่อนบิตไปเป็นจำนวนเท่ากับจำนวนครั้งที่กำหนดไว้ในตารางที่ ก-3 (ในภาคผนวก ก) โดยมี C0 และ D0 เป็นค่าเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และค่าของกุญแจย่อยในแต่ละชุด จะได้จากการนำเอา Ci และ Di มาต่อกัน แล้วนำค่านั้นไปเข้าตาราง PC-2 (ตารางที่ ก-4 ในภาคผนวก ก)

Ki = PC_2(CiDi)

ในการทำงานนั้น เนื่องจากขนาดของกุญแจมี 48 บิต แต่ R มีขนาด 32 บิต ดังนั้น จึงต้องทำการขยายขนาดของ R ให้เป็น 48 บิต เพื่อให้สามารถนำ R และกุญแจมา exclusive OR กันได้ โดยนำ R ไปเข้าตาราง E (ตารางที่ ก-5 ในภาคผนวก ก)

หลังจากทำการขยายขนาดของ R แล้ว จึงนำเอา R และกุญแจที่มีขนาดเท่ากันมาทำการ XOR กัน หลังจากนั้น นำผลลัพธ์ที่ได้มาทำการแบ่งออกเป็น 8 ส่วน ส่วนละ 6 บิต แล้วนำไปเข้าตาราง S (ตารางที่ ก-7 ในภาคผนวก ก) ซึ่งจะได้ผลลัพธ์ออกมาส่วนละ 4 บิต รวมกันได้ 32 บิต

จากนั้น นำข้อมูลขนาด 32 บิตนี้ ไปทำการสับเปลี่ยนตำแหน่งในตาราง P (ตารางที่ ก-6 ในภาคผนวก ก)

เมื่อทำขั้นตอนในส่วนของการเข้าฟังก์ชันครบ 16 ครั้งแล้ว จะได้ข้อมูลชุดหนึ่งที่มีขนาด 64 บิต โดยเรียกว่า PREOUTPUT หลังจากนั้น จึงนำข้อมูลนี้ไปทำการเรียงบิตกลับ

3. การเรียงบิตกลับ (Inverse Initial Permutation)

เมื่อได้ข้อมูลที่เป็น PREOUTPUT ออกมาแล้ว จะนำข้อมูลนั้น มาทำการสับเปลี่ยนตำแหน่งกลับจากในตอนแรก ซึ่งจะได้เป็นข้อมูลที่ผ่านการเข้ารหัสแล้ว (Cipher Text)

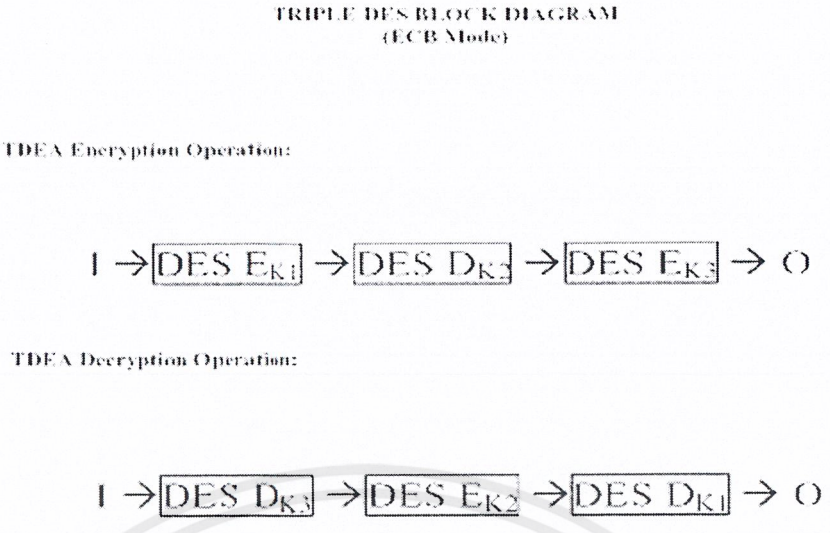
สำหรับกระบวนการถอดรหัสนั้น ทำเช่นเดียวกันกับกระบวนการเข้ารหัส คือ ใส่ข้อมูลที่ถูกเข้ารหัสข้อมูลมาแล้วที่มีขนาด 64 บิต พร้อมกับกุญแจตัวเดิมที่ใช้ในการเข้ารหัส แล้วทำตามขั้นตอนต่างๆ เหมือนกับการเข้ารหัสข้อมูล แต่ต้องทำการเปลี่ยนลำดับของกุญแจใหม่ โดยใช้กุญแจจากลำดับท้ายสุดเรียงมาจนถึงลำดับแรกสุด คือ K16, K15,..., K1

3.3 การเข้ารหัสแบบ 3DES (Triple Data Encryption Standard)

และการใช้งานจริงหากเราต้องการใช้ข้อมูลมีความปลอดภัยมากขึ้นสามารถทำได้โดยเราสามารถเอาอัลกอริทึม DES มาประยุกต์ใช้งาน

การเข้ารหัสแบบ 3DES นั้น เป็นการเข้ารหัสอีกแบบหนึ่ง โดยอิงมาตรฐานมาจากการเข้ารหัสแบบ DES ซึ่งได้ทำการเข้าและถอดรหัสแบบ DES เป็นจำนวน 3 ครั้ง และใช้กุญแจเป็นจำนวน 3 ตัวในการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-4 กระบวนการเข้าและถอดรหัสด้วย 3DES

โดยมีขั้นตอนในการเข้ารหัสข้อมูล ดังนี้คือ

1. ทำการเข้ารหัสข้อมูลโดยใช้กุญแจตัวแรก
2. ทำการถอดรหัสข้อมูลโดยใช้กุญแจตัวที่สอง และ
3. ทำการเข้ารหัสข้อมูลโดยใช้กุญแจตัวที่สาม

ในส่วนของขั้นตอนในการถอดรหัสข้อมูล มีดังนี้คือ

1. ทำการถอดรหัสข้อมูลโดยใช้กุญแจตัวแรก
2. ทำการเข้ารหัสข้อมูลโดยใช้กุญแจตัวที่สอง และ
3. ทำการถอดรหัสข้อมูลโดยใช้กุญแจตัวที่สาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การจัดการกับวินโดวส์

การจัดการกับวินโดวส์

4.1 เมสเซจ

เมสเซจเป็นอินพุตหรือข้อมูลเข้าเพียงทางเดียวของแอพพลิเคชั่น วินโดวส์เป็นตัวแทนต่อทุก ๆ เหตุการณ์ทั้งหมดที่ต้องการตอบสนอง เมสเซจเป็นตัวแปรโครงสร้างชนิดหนึ่งที่ประกอบไปด้วยส่วนค่าอ้างอิงของเมสเซจนั้น กับส่วนพารามิเตอร์ โดยที่พารามิเตอร์จะขึ้นกับเมสเซจชนิดนั้น ๆ

4.1.1 การสร้างและการประมวลผลเมสเซจ

วินโดวส์จะมีการสร้างเมสเซจขึ้นทุกครั้งที่มีอินพุตเข้ามายังระบบไม่ว่าจะเป็นการเลื่อนเมาส์ การกดคีย์ หรืออื่น ๆ เพื่อบอกแก่แอพพลิเคชั่นหรือตัววินโดวส์เองถึงเหตุการณ์ (อินพุต) ที่เกิดขึ้น โดยวินโดวส์จะนำเมสเซจที่สร้างขึ้นเหล่านี้ใส่เข้าไปในคิวของระบบจากนั้นก็ส่งผ่านไปยังคิวของแอพพลิเคชั่นที่เหมาะสมต่อไป ลักษณะของคิวของแอพพลิเคชั่นนี้เป็นแบบเข้าก่อนออกก่อน โดยที่การเข้าก็คือวินโดวส์ดึงเมสเซจจากคิวของระบบมาใส่ให้แก่คิวของแอพพลิเคชั่นส่วนการออกคือแอพพลิเคชั่นเป็นตัวดึงออกไปเอง โดยเรียกใช้ฟังก์ชั่น Getmessage จากนั้นก็แจกแจงและส่งไปยังฟังก์ชั่นประจำวินโดวส์ต่าง ๆ ด้วยฟังก์ชั่น DispatchMessage

มีเมสเซจบางอันที่วินโดวส์ต้องการส่งไปยังฟังก์ชั่นประจำวินโดวส์โดยตรง แทนที่จะวางลงในคิว อาจเรียกเมสเซจนี้ว่าเป็น "เมสเซจลัดคิว" เมสเซจลัดคิวนี้จะเป็นเมสเซจที่มีผลกับวินโดวส์นั้น ๆ โดยเฉพาะวินโดวส์และแอพพลิเคชั่นประจำวินโดวส์โดยตรงอีกทั้งฟังก์ชั่นประจำวินโดวส์นั้นยังสามารถส่งค่าการประมวลผลเมสเซจกลับมาได้อีกด้วย

ตัวอย่างของเมสเซจลัดคิวอย่างหนึ่งก็คือ การใช้ฟังก์ชั่น CreateWindow นอกจากจะเป็นการสร้างวินโดวส์แล้ว ยังเป็นการบังคับให้วินโดวส์ส่งเมสเซจ WM_C REATE ซึ่งเป็นเมสเซจลัดคิวไปยังฟังก์ชันประจำวินโดวส์ แล้วรอจนกว่าฟังก์ชั่นนั่นจะจัดการกับเมสเซจเสร็จ เมสเซจ WM_CREATE จะไม่ผ่านแอพพลิเคชั่นเลย

แต่ก็ไม่ใช่ว่ามีเพียงวินโดวส์เท่านั้นที่สามารถสร้างเมสเซจขึ้นมาได้ตัวแอพพลิเคชั่นเองก็สามารถสร้างเมสเซจแล้วส่งไปยังคิวของตนเอง และคิวของแอพพลิเคชั่นอื่น ๆ ได้เช่นกัน

แอพพลิเคชั่นจะใช้ฟังก์ชั่น GetMessage ในลูปฟังก์ชั่น WinMain เพื่อดึงเมสเซจออกจากคิวของตัวแอพพลิชั่นเอง ฟังก์ชั่น GetMessage จะทำงานโดยเริ่มจากดูว่าในคิวมีเมสเซจหรือไม่หากมีก็จะดึงเมสเซจที่อยู่แรกสุดไป แต่หากไม่มีเมสเซจอยู่ในคิว ก็จะเกิดการรอเมสเซจขึ้น และปล่อยการควบคุมกลับคืนไปให้วินโดวส์ เพื่อแบ่งการควบคุมแอพพลิเคชั่นอื่นสามารถทำงานได้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อฟังก์ชั่นWinMainของแอพพลิเคชั่นได้รับเมสเซจจากคิวแล้วก็ต้องใช้เรียกฟังก์ชั่น DispatchMessage เพื่อแจกแจงเมสเซจและการส่งเมสเซจนั้นไปให้ฟังก์ชั่นประจำวินโดวส์ต่างๆการทำเช่นนี้จะเป็นการถ่ายการควบคุมให้กับฟังก์ชั่นประจำวินโดวส์ที่จะประมวลผลเมสเซจนั้นหลังจากที่ทำงานเกี่ยวข้องกับเมสเซจนี้เรียบร้อยแล้ว ก็ถ่ายการควบคุมกลับไปให้ฟังก์ชั่นหลักในแอพพลิเคชั่นเพื่อดึงเมสเซจต่อไปในคิวมาประมวลผลต่อไป

ข้อควรระวัง ฟังก์ชั่นประจำวินโดวส์ไม่ควรคาดว่าเมสเซจใดจะเข้ามาก่อน เพราะวินโดวส์สามารถส่งเมสเซจมาในลำดับอย่างไรก็ได้

ส่วนการรับข้อมูลจากคีย์บอร์ดนั้นวินโดวส์จะรับเข้ามาในลักษณะของรหัสของคีย์ที่กด ดังนั้นจึงต้องมีการแปลงรหัสของคีย์เหล่านั้นให้เป็นรหัสของตัวอักษรเมื่อแปลงได้แล้วฟังก์ชั่นก็จะส่งเมสเซจของตัวเองเข้าไปในคิว พร้อมทั้งรหัสตัวอักษรที่ผู้ใช้กดเข้ามาเป็นพารามิเตอร์ของเมสเซจนั้น

4.1.2 การแปลและตีความเมสเซจ

โดยทั่วไปแล้วจะมีการใช้ฟังก์ชั่น TranslateMessage นี้กับทุกๆเมสเซจที่เข้ามา โดยที่หากไม่ใช่เมสเซจที่เกี่ยวกับคีย์บอร์ดก็จะไม่มีผลอันใด

ตัวอย่างต่อไปนี้ก็จะเป็นการแสดงถึงหน้าตาของเมสเซจลูปที่อยู่ฟังก์ชั่น WinMain ของแอพพลิเคชั่น เมสเซจลูปจะดึงเมสเซจออกจาคิวและแจกจ่ายดังนี้

```
int PASCAL WinMain ( hInt, hPrevInst, lpCmdLine, ShowCmd )
HINSTANCE hInst ;
HINSTANCE hPrevnst ;
LPSTR lpCmdLine ;
int ShowCmd ;
{
        MSG msg ;
        .
        .
        .
        While ( GetMessage ( &msg, NULL, 0,0 ) ) {
                TranslateMessage ( &msg ) ;
                DispatchMessage ( &msg ) ;
        }
        return msg.wParam;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
```

ส่วนแอพพลิเคชั่นที่มีการใช้คีย์ทันใจ (accelerator key) ของเมนู ก็ต้องมีการสร้างตารางคีย์ทันใจที่ใช้ไว้ในไฟล์รีซอร์สจากนั้นก็ต้องมีการโหลดตารางนี้เข้ามาโดยใช้ฟังก์ชั่น LoadAccelerator โดยจะทำให้ลักษณะของลูปในการรับส่งเมสเซจเป็นดังนี้

```
While ( GetMessage ( ( ( LPMSG ) &msg, ( HWND ) NULL, 0, 0 ) ) )
{
    if ( TranslateAccelerator ( hWindow, hAccel, ( ( LPMSG ) &msg ) == 0 )
    {
        TranslateMesage ( ( LPMSG ) &msg ) ) ;
        DispatchMesage ( ( LPMSG ) &msg ) ) ;
    }
}
```

ที่ต้องการตรวจสอบว่าค่าที่ได้กลับมาจากการเรียกใช้ฟังก์ชั่น TranslateAccerator นั้นก็เพราะว่าจากเมสเซจนั้นเป็นการกดคีย์ทันใจ ฟังก์ชั่น TranslateAccerator ก็จะทำการแปลงและส่งเมสเซจที่แปลงแล้วนั้น ไปยังฟังก์ชั่นประจำวินโดวส์เอง ดังนั้นจึงไม่ต้องมาเรียกใช้ฟังก์ชั่น TranslateMessage และฟังก์ชั่น DispatchMessage อีก

4.1.3 การตรวจสอบเมสเซจ

เราใช้ฟังก์ชั่น PeekMessage เพื่อมองหาและตรวจสอบเมสเซจบางเมสเซจในระหว่างการทำงานที่กินเวลา โดยไม่ต้องการมีผลกระทบกระเทือนต่อการเป็นอยู่ของเมสเซจ เมื่อเรียกใช้ฟังก์ชั่นนี้ ถ้ามีเมสเซจในคิวฟังก์ชั่นนี้จะให้ค่าที่ไม่เป็น 0 กลับมา ทำให้สามารถประมวลเมสเซจได้โดยไม่ต้องกลับไปที่ลูปของฟังก์ชั่น WinMain ประโยชน์ของฟังก์ชั่นนี้คือ เมื่อมีการทำงานบางอย่างที่ต้องกินเวลามาก (ใช้เวลานานกว่าจะคืนการควบคุมไปให้ฟังก์ชั่นหลัก) เช่น การอ่านเขียนดิสก์ ก็สามารถใช้ฟังก์ชั่นนี้ในการตรวจสอบว่า ขณะนี้ผู้ใช้มีการกดคีย์เพื่อยกเลิกการทำงานหรือไม่

4.1.3 การส่งและส่งผ่านเมสเซจ

ฟังก์ชั่น Postmessage และ SendMessage มีหน้าที่ในการส่งเมสเซจไปยังฟังก์ชั่นประจำวินโดวส์ของตัวเองหรือฟังก์ชั่นประจำวินโดวส์ของแอพพลิเคชั่นอื่น นอกจากนี้ยังมีฟังก์ชั่น PostAppMessage ที่ทำงานแบบเดียวกับฟังก์ชั่น PostMessage แต่เป็นการส่งผ่านโดยอาศัยแฮนเดิลโมดูลของแอพพลิเคชั่นแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชั่น PostMessage จะส่งเมสเซจผ่านไปทางคิวของแอพพลิเคชั่น (เราจะขอเรียกการส่งแบบนี้ว่าการส่งผ่าน) โดยที่ผู้ส่งจะยังไม่เสียการควบคุม และผลของเมสเซจที่ส่งไปนั้นก็จะยังไม่เกิดขึ้นจนกว่าแอพพลิเคชั่นนั้น ๆ จะดึงเมสเซจที่ส่งไปนั้นขึ้นมา ส่วน SendMessage จะส่งเมสเซจลัดคิวไปยังฟังก์ชั่นประจำวินโดวส์ที่ต้องการทันที (โดยเราจะขอเรียกการส่งแบบนี้ว่าการส่ง) โดยที่ไม่ต้องผ่านคิวของแอพพลิเคชั่น ซึ่งการส่งเมสเซจไปแบบนี้ ผู้ที่ส่งจะสูญเสียการควบคุมไปให้ฟังก์ชั่นประจำวินโดวส์ที่ส่งเมสเซจให้ และเมื่อฟังก์ชั่นทำงานเสร็จแล้วก็จะคืนการควบคุมกลับมา ส่วนค่าที่ได้กลับมาของการเรียกใช้ SendMessage นั้นคือค่าที่ฟังก์ชั่นประจำวินโดวส์นั้นส่งค่ากลับมา แต่ถ้าเป็นค่าที่ได้รับกลับมาจากฟังก์ชั่น PostMessage เป็นเพียงว่าสามารถส่งเมสเซจนั้นไปได้สำเร็จหรือไม่

4.1.4 ฟังก์ชั่นที่ใช้กับเมสเซจ

ฟังก์ชั่นในกลุ่มนี้มีหน้าที่ในการอ่านเมสเซจขึ้นมาจากคิวและประมวลผลเมสเซจในวินโดวส์ ฟังก์ชั่นที่ใช้กับเมสเซจก็จะมีตารางดังนี้

ฟังก์ชั่น	หน้าที่
CallWindowProc	ส่งผ่านข้อมูลของเมสเซจไปยังฟังก์ชั่น
DispatchMessage	แจกแจงเมสเซจและส่งเมสเซจไปยังฟังก์ชั่นประจำวินโดวส์ใด ๆ
GetMessage	รับเมสเซจในช่วงของเมสเซจที่กำหนดไว้
GetMessagePost	ให้ค่าตำแหน่งของเมาส์ในขณะที่มีการรับเมสเซจล่าสุด
GetMessageTime	ให้ค่าเวลาในขณะที่มีการรับเมสเซจล่าสุด
PeekMessage	ตรวจสอบว่ามีเมสเซจในคิวหรือไม่ไม่โดยไม่ได้ดึงออกมา
PostAppMessage	ส่งเมสเซจไปยังแอพพลิเคชั่น
PostMessage	ส่งเมสเซจไปยังคิวของแอพพลิเคชั่น
PostQuitmesage	ส่ง WM_QUIT ไปยังแอพพลิเคชั่น
ReplyMessage	ตอบรับเมสเซจ
SendMessage	ส่งเมสเซจไปยังวินโดวส์ต่าง ๆ
TranslateAccelerator	ประมวลผลคีย์ทันใจของเมนูให้เป็นเมสเซจและส่งเมสเซจให้แก่วินโดวส์
Translatemessage	แปลค่ารหัสของคีย์ที่กดให้เป็นรหัสตัวอักษร
WaitMessage	ส่งการควบคุมไปให้แอพพลิเคชั่นอื่น
Winmain	เป็นจุดที่เริ่มเข้าไปทำงานของวินโดวส์แอพพลิเคชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ช่องเกี่ยวโยง (Hook)

ช่องเกี่ยวโยงนับเป็นกลไกการจัดการเมสเซจอย่างหนึ่งของวินโดวส์ ที่สามารถให้แอพพลิเคชั่นสามารถเข้าถึงเมสเจที่ส่งกันไปมาวินโดวส์มีช่องเกี่ยวโยงให้ใช้กันหลายแบบซึ่งแต่ละแบบก็จะมีความสามารถในการเข้าถึงเมสเซจต่างกันไป การใช้ประโยชน์จากช่องเกี่ยวโยงคือการติดตั้งฟังก์ชั่นกรองเมสเซจ (filter function) ซึ่งจะจัดการกับเมสเซจจากช่องเกี่ยวโยงก่อนที่เมสเซจนั้นประจำวินโดวส์ที่อยู่ปลายทาง

ลูกโซ่ของฟังก์ชั่นการกรองเมสเซจ

ลูกโซ่ของฟังก์ชั่นกรองเมสเซจเกิดจากการต่อฟังก์ชั่นกรองเมสเซจหลายๆฟังก์ชั่นเข้ากับช่องเกี่ยวโยงเดียวกันของระบบ โดยที่ตำแหน่งแรกสุดของลูกโซ่คือ ช่องเกี่ยวโยงนั่นเอง และปลายของลูกโซ่ก็จะไปอยู่ที่ฟังก์ชั่นประจำวินโดวส์ เมื่อเราติดตั้งฟังก์ชั่นกรองเมสเซจ ฟังก์ชั่นดังกล่าวจะเกี่ยวข้องกับช่องเกี่ยวโยงอันถือว่าเป็นจุดเริ่มต้นขอลูกโซ่รับเมเสเสจแล้วส่งเมสเซจที่กรองแล้วโยงเข้ากับฟังก์ชันกรองเมสเซจที่เคยติดตั้งอยู่ ซึ่งตอนนี้อยู่ในตำแหน่งถัดไป ฟังก์ชั่นกรองถัดไปนั้นก็จะทำเช่นนี้ต่อไปเรื่อย ๆ เป็นลูกโซ่ที่จะโยงเข้าหาฟังก์ชั่นประจำวินโดวส์ในที่สุด

ฟังก์ชั่นกรองเมสเซจจะสามารถโยงตัวเองเข้ากับช่องเกี่ยวโยงด้วยฟังก์ชัน SetWindowHook ในแต่ละครั้งที่เรียก ฟังก์ชั่นกรองเมสเซจเหล่านั้นก็จะถูกโยงไปยังจุดเริ่มต้นของลูกโซ่ (ซึ่งอาจมีฟังก์ชั่นกรองเมสเซจเกี่ยวอยู่) นั่นคือโยงต่อเข้ากับช่องเกี่ยวโยง และทุกครั้งที่แอพพลิเคชั่นติดตั้งฟังก์ชันกรองเมสเซจเข้ากับช่องเกี่ยว ก็จะต้องสำรองพ้นที่สำหรับเก็บตำแหน่งแอดเดรสของฟังก์ชั่นกรองเมสเซจที่เคยอยู่เดิม เพื่อส่งเมสเซจที่กรองแล้วโยงเขข้ากับฟังก์ชั่นกรองที่อยู่ถัดไป

การจะถอดฟังก์ชั่นกรองเมสเซจออกจากลูกโซ่คือแอพพลิเคชั่นเรียกใช้ฟังก์ชั่น UnHookWindowHook ด้วยชนิดของช่องเกี่ยวโยงและตัวชี้ไปยังฟังก์ชั่นกรองนั้น

ช่องเกี่ยวโยงวินโดวส์และช่องเกี่ยวโยงดีบักมีตารางนี้

ชนิด	จุดประสงค์
WH_CALLWNDPROC	ติดตั้งฟังก์ชั่นกรองเมสเซจของวินโดวส์
WH_CBT	ติดตั้งฟังก์ชั่นกรองเมสเซจสำหรับช่วยการสอน
WH_DEBUG	ติดตั้งฟังก์ชั่นกรองเมสเซจสำหรับการดีบัก
WH_GETMESSAGE	ติดตั้งฟังก์ชั่นกรองเมสเซจ (เฉพาะวินโดวส์รุ่น ดีบัก)
WH_HARDWARE	ติดตั้งฟังก์ชั่นกรองเมสเซจสำหรับฮาร์ดแวร์ที่ไม่เป็นมาตรฐานของวินโดวส์
WH_JOURNALPLAYBACK	ติดตั้งฟังก์ชั่นกรองเมสเซจเล่นกลับ
WH_JOURNALRECORD	ติดตั้งฟังก์ชั่นกรองเมสเซจบันทึก
WH_KEYBOARD	ติดตั้งฟังก์ชั่นกรองเมสเซจคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WH_MOUSE	ติดตั้งฟังก์ชั่นกรองเมสเซจเมาส์
WH_MSGFILTER	ติดตั้งฟังก์ชั่นกรองเมสเซจทั่วไป
WH_SYSMSGFILTER	ติดตั้งฟังก์ชั่นกรองเมสเซจทั้งหมด

เพิ่มเติม ช่องเกี่ยวโยงระบบ WH_CALLWNDPROC และ WH_GETMESSAGE จะมีผลกระทบต่อประสิทธิภาพของระบบ จึงควรใช้เฉพาะในการดีบักเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

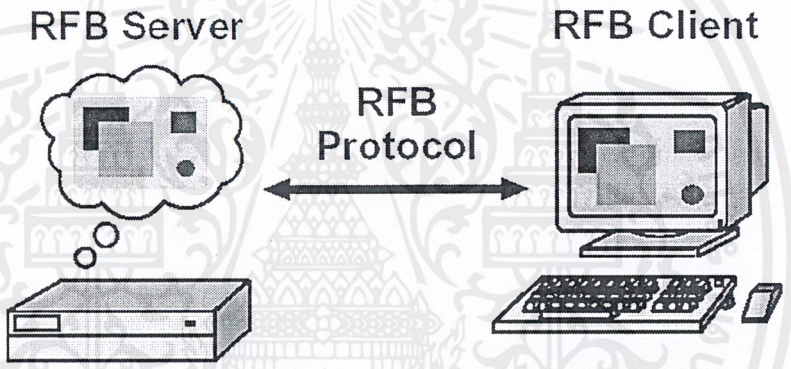
บทที่ 5

RFB Protocol

5.1 อาร์เอฟบีโพรโตคอล (RFB Protocol)

อาร์เอฟบี (RFB : Remote Frame Buffer) เป็นโปรโตคอลสำคัญที่เราจะนำมาใช้เป็นโพรโตคอลที่จะถูกนำมาใช้ในการเข้าควบคุมเครื่องคอมพิวเตอร์จากระยะไกลผ่านยูสเซอร์อินเตอร์เฟสเนื่องจากว่ามันจะทำงานในระดับของเฟรมบัฟเฟอร์ มันจึงสามารถใช้ได้กับระบบและแอพพลิเคชันทั้งหมดของวินโดวส์ รวมไปถึงระบบ เอ็กซ์11 (X11), วินโดวส์ 3.1/95/เอ็นที และแมคอินทอช

ในโพรโตคอลนี้จะเรียกส่วนต่างๆดังนี้ รีโมทเอ็นด์พ้อยน์คือจุดที่ติดต่อกับผู้ใช้(เช่น จอ คีย์บอร์ด เม้าส์) จะเรียกว่า "อาร์เอฟบีไคลเอนต์" และจุดเอ็นท์พ้อยน์ที่เกิดการเปลี่ยนแปลงของเฟรมบัฟเฟอร์ จะเรียกว่า "อาร์เอฟบีเซิร์ฟเวอร์"



รูปที่ 5-1 การเชื่อมต่อของโพรโตคอล

อาร์เอฟบี โดยทั่วไปจะถูกเรียกว่า " ตินไคลเอนต์ (thin client) โพรโตคอล " ส่วนที่เป็นหลักสำคัญในการออกแบบอาร์เอฟบีโพรโตคอล คือการทำให้ไคลเอนต์มีข้อกำหนดต่างๆน้อยที่สุด นั่นก็คือไคลเอนต์สามารถที่จะทำงานบนฮาร์ดแวร์ได้หลากหลายไม่จำกัดอยู่แค่แบบหนึ่งแบบใดเท่านั้นและงานหนักของการนำโพรโตคอลไปใช้งานกับไคลเอนต์ก็คือการทำให้มันทำงานได้ง่ายที่สุดเท่าที่จะเป็นไปได้

โพรโตคอลนี้จะทำให้ไคลเอนต์ทำงานแบบสแตทเลส (stateless) คือ ถ้าไคลเอนต์หลุดการเชื่อมต่อจากเซิร์ฟเวอร์และต่อมาได้ทำการเชื่อมต่อเข้าไปใหม่ยังเซิร์ฟเวอร์ตัวเดิม สถานะของส่วนที่ติดต่อกับผู้ใช้จะต้องคงสถานะเดิมไว้ยิ่งกว่านั้นตัวโพรโตคอลยังสามารถยอมให้ไคลเอนต์หลายๆตัวทำการเชื่อมต่อไปยังอาร์เอฟบีเซิร์ฟเวอร์ได้ที่ฝั่งไคลเอนต์ผู้ใช้สามรถที่จะเห็นส่วนติดต่อกับผู้ใช้เหมือนกับที่เกิดขึ้นในฝั่งของเซิร์ฟเวอร์ผลจากข้อกำหนดนี้จะทำให้การติดต่อกับโปรแกรมใช้งานของผู้ใช้โปรแกรมจากที่อื่น (remote) กลายเป็นสิ่งที่ง่ายดายทำให้สามารถใช้งานเครื่องที่ฝังโปรแกรมส่วนเซิร์ฟเวอร์ที่ไหนก็ตามที่มีการเชื่อมต่อกับเน็ตเวิร์กและผู้ใช้งานยังสามารถกระทำการใดๆกับโปรแกรมใช้งานที่อยู่ในเครื่องส่วนตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่อยู่ห่างออกไปได้นอกจากนั้นสถานะของการเข้ากระทำนี้จะถูกเก็บรักษาไว้ระหว่างที่มีการกระทำจากที่อื่น

5.2 โพรโตคอลของการแสดงภาพ

ในส่วนแสดงภาพของโพรโตคอลนี้ ใช้พื้นฐานการแสดงรูปภาพอย่างหยาบๆ คือ ใช้การวางจุดของพิกเซลที่ตำแหน่ง เอ็กซ์,วาย

ในขั้นตอนแรกนี้อาจจะมองวิธีที่จะวาดส่วนประกอบหลายๆส่วน ของส่วนติดต่อผู้ใช้ อย่างไรก็ตามหากรูปแบบการเข้ารหัสของเดทาพิกเซลมีความแตกต่างกันก็จะทำให้เกิดความยุ่งยากจากค่าพารามิเตอร์ต่างๆที่เกิดขึ้น อย่างเช่น แบนด์วิดธ์ของเน็ตเวิร์ก ความเร็วในการวาดของไคลเอ็นต์และความเร็วในการประมวลผลของเซิร์ฟเวอร์

ลำดับการในการจัดวางรูปสี่เหลี่ยมเหล่านี้ที่ประกอบกันขึ้นทำให้เกิดการเปลี่ยนแปลงของเฟรมบัฟเฟอร์ (หรือเรียกว่า อัพเดท) การอัพเดทเป็นการเปลี่ยนแปลงจากสถานะการใช้งานเฟรมบัฟเฟอร์หนึ่งไปยังสถานะอื่น ในอีกนัยหนึ่งก็คือมันจะคล้ายกับเฟรมของวิดีโอที่ส่งออกมาเป็นลำดับ ในกรณีทั่วๆไปแล้วเฟรมเหล่านี้จะต้องติดกันแต่ก็ไม่จำเป็นในกรณีของเรื่องนี้

โพรโตคอลในการอัพเดท จะถูกกระตุ้นโดยไคลเอนต์ นั่นคือการอัพเดทจะถูกส่งจากเซิร์ฟเวอร์ไปยังไคลเอนต์เท่านั้นในการตอบสนองความต้องการของไคลเอนต์ ในระบบที่ไคลเอนต์และเน็ตเวิร์กมีความช้ามากการอัพเดทก็จะช้าลงด้วย ในแอพพลิเคชันปกติทั่วๆไปการเปลี่ยนแปลงที่พื้นที่เดียวกันของเฟรมบัฟเฟอร์จะทำให้มีการอัพเดทเร็วกว่ากรณีอื่นๆ ในระบบที่คอมพิวเตอร์หรือ/และระบบเครือข่ายมีความช้าสถานะทรานส์เชี้ยนอาจถูกยกเลิกได้ซึ่งจะทำให้เกิดทราฟฟิกและการวาดภาพน้อยลงในไคลเอนต์

5.3 อินพุทโพรโตคอล

ในส่วนอินพุทของโพรโตคอลจะขึ้นอยู่กับมาตรฐานของอุปกรณ์คีย์บอร์ดและอุปกรณ์จำพวกชี้ตำแหน่ง อีเวนต์ของอินพุทจะถูกส่งไปยังเซิร์ฟเวอร์โดยผ่านทางไคลเอนต์ เมื่อไหร่ก็ตามที่ผู้ใช้กดปุ่มหรือกดเมาท์ หรือมีการเคลื่อนที่ของเมาท์เกิดขึ้นก็จะถูกส่งออกไปด้วย อีเวนต์ของอินพุทเหล่านี้สามารถที่จะถูกสังเคราะห์จากอุปกรณ์อื่นที่ไม่เป็นอุปกรณ์มาตรฐาน ตัวอย่างเช่น อุปกรณ์ที่ใช่เขียนแทนปากกาจะมีตัวจดจำที่ทำหน้าที่เปลี่ยนให้เป็นอีเวนต์ที่คล้ายกับอีเวนต์ที่มาจากคีย์บอร์ด

5.4 การนำเสนอของพิกเซลเดทา

ในการเริ่มต้นของการโต้ตอบกันระหว่าง อาร์เอฟบีไคลเอนต์และเซิร์ฟเวอร์จะต้องมีการตกลงกันในเรื่องของรูปแบบ (format) และการเข้ารหัส (encode) ของข้อมูลที่จะทำการส่งออกไปก่อน ข้อตกลงเหล่านี้จะถูกออกแบบมาเพื่อให้ไคลเอนต์ทำงานง่ายที่สุดเท่าที่เป็นไปได้ อย่างเช่นเส้นล่างสุดของเฟรมบนเครื่องเซิร์ฟเวอร์จะต้องสามารถส่งเดทาพิกเซลให้อยู่ในรูปแบบที่ไคลเอนต์ต้องการได้ อย่างไรก็ตามไคลเอนต์จะต้องสามารถจัดการกับรูปแบบและการเข้ารหัสที่หลากหลายได้เท่าๆกันโดยมันจะทำการเลือกวิธีที่ง่ายสำหรับการจัดการของเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบของจุดภาพ (Pixel format) มาจากการนำเสนอจุดของสีเดี่ยวๆด้วยค่าพิกเซลโดยทั่วๆไป แล้วรูปแบบของพิกเซลจะมีขนาดอยู่ที่ 24 บิต หรือ 16บิต ค่าภายในบิตฟิลด์จะถูกแปลออกโดยตรงเป็นสีแดง สีเขียวและสีน้ำเงินเข้ม และที่ความละเอียดขนาด 8 บิต เรียกว่า"ชุดสี (color map) " จะเป็นสีที่ไม่มีการกำหนดในชุดสีมาตรฐานสามารถทำการแปลงจากค่าของพิกเซลเป็นค่าของสีต่างๆได้เลย

การเข้ารหัส(Encoding)หมายถึงการทำให้กรอบของข้อมูลพิกเซลที่ถูกส่งออกไปบนสายสัญญาณสามารถทำการส่งได้อย่างรวดเร็ว ทุกๆชุดพิกเซลที่ถูกส่งออกไปจะต้องถูกปะหัวด้วยด้วยเฮดเดอร์ที่บอกตำแหน่งเอ็กซ์, ตำแหน่งวายบนหน้าจอ ความกว้างยาวของกรอบสี่เหลี่ยมและชนิดของการเข้ารหัสจะเป็นตัวกำหนดรูปแบบการเข้ารหัสของข้อมูลของพิกเซลซึ่งโดยตัวของข้อมูลเองแล้ว มันจะอนุญาตให้มีการกำหนดการเข้ารหัสได้

โพรโตคอลนี้สามารถทำขยายออกได้โดยการเพิ่มชนิดของการเข้ารหัสใหม่ ชนิดของการเข้ารหัสที่ใช้อยู่ในปัจจุบัน คือ การเข้ารหัสแบบหยาบ (raw encoding) การเข้ารหัสแบบการคัดลอกกรอบ (copy rectangle encoding) การเข้ารหัสแบบอาร์อาร์อี (RRE : rise-and-run-length) การเข้ารหัสแบบซีโออาร์อาร์อี (CoRRE : Compact RRE) และการเข้ารหัสแบบเฮ็กซ์ไทล์ (hextile encoding) แต่ในเรื่องนี้เราจะใช้ การเข้ารหัสแบบ เฮกซ์ไทล์และแบบการคัดลอกกรอบเทานั้นเนื่องจากว่าการเข้ารหัสแบบนี้นั้นได้ถูกทำให้เป็นวิธีที่ดีที่สุดในการบีบอัดข้อมูลสำหรับเครื่องคอมพิวเตอร์ที่ใช้งานทั่วไป ตัวอย่างอื่นๆของการเข้ารหัสที่เป็นไปได้ เช่น การเข้ารหัสแบบ JPEG ที่ใช้ส่งข้อมูลภาพและ MPEG สำหรับการส่งภาพเคลื่อนไหว

5.4.1 การเข้ารหัสแบบหยาบ (Raw encoding)

เป็นแบบที่ง่ายที่สุดของการเข้ารหัสโดยข้อมูลจะมีลักษณะเป็นข้อมูลพิกเซลแบบหยาบๆ เท่านั้น ในตัวอย่างนี้ให้ข้อมูลประกอบด้วย ค่า n พิกเซล โดยที่ n เป็นความกว้างคูณกับความสูงของกรอบ ค่าที่ได้จะถูกนำมาใช้แทนแต่ละพิกเซล จากซ้ายไปขวาตามเส้นแสกน อาร์เอฟบีไคลเอนต์ทุกตัวจะต้องสามารถที่จะจัดการกับข้อมูลพิกเซลในการเข้ารหัสแบบนี้ได้ และอาร์เอฟบีเซิร์ฟเวอร์ควรจะสร้างการเข้ารหัสแบบนี้เท่านั้นถ้าไคลเอนต์ไม่กำหนดการเข้ารหัสแบบอื่นๆเป็นพิเศษ

5.4.2 การเข้ารหัสแบบคัดลอกกรอบ (Copy Rectangle encoding)

การเข้ารหัสแบบคัดลอกกรอบนี้เป็นวิธีที่ง่ายและมีประสิทธิภาพมาก จะถูกใช้เมื่อไคลเอนต์มีข้อมูลพิกเซลเดียวกันภายในบัฟเฟอร์ซ้ำกัน ในการเข้ารหัสแบบนี้จะประกอบด้วยพิกัดของเอ็กซ์ วาย โดยตำแหน่งนี้เป็นตำแหน่งในเฟรมบัฟเฟอร์ เมื่อถึงเวลา ไคลเอนต์ก็สามารถทำการคัดลอกกรอบของข้อมูลพิกเซลนี้ไปได้เลย จึงสามารถที่จะนำไปใช้ในหลายๆรณีได้ ที่เห็นได้ชัดที่สุดคือ เมื่อผู้ใช้งานทำการเคลื่อนย้ายหน้าต่างผ่านไปมาและเมื่อวินโดวส์มีการเลื่อนขึ้นหรือลงของหน้าต่าง จุดที่เห็นชัดน้อยที่สุดคือการใช้สำหรับการวาดข้อความหรือการทำรูปแบบซ้ำๆอื่นๆการที่จะให้ข้อมูลมีความกระชับสมบูรณ์ในตัวเซิร์ฟเวอร์ที่ฉลาดอาจสามารถส่งข้อมูลออกมาในรูปแบบไดรูปแบบหนึ่งออกมาตรงๆได้เลยและการที่รู้ตำแหน่งของรูปแบบของข้อมูลก่อนหน้าในเฟรมบัฟเฟอร์ จะทำให้การส่งข้อมูลตามลำดับของรูปแบบที่คล้ายกันจะเกิดขึ้นโดยใช้การเข้ารหัสแบบคัดลอกกรอบได้ง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.3 การเข้ารหัสแบบอาร์อาร์อี (RRE : rise-and-run-length encoding)

การเข้ารหัสแบบนี้จะแบ่งการทำงานออกเป็นสองส่วนสำคัญคือการรัน-ลองและการเข้ารหัสในระดับสองมิติ กรอบข้อมูลที่มีการเข้ารหัสแบบอาร์อาร์อี จะไม่เพียงแต่ถูกย่อในระดับเดียวกันหรือดีกว่าด้วยการเข้ารหัสแบบ รัน-ลอง เท่านั้นหากแต่กรอบที่ถูกส่งออกไปจากไคลเอนต์นั้นจะถูกเข้ารหัสให้อยู่ในรูปแบบที่สามารถที่จะนำไปแสดงผลได้ทันทีและเพิ่มประสิทธิภาพโดยการเรียกใช้กราฟิกเอ็นท์จิน (graphic engines) แบบที่ง่ายที่สุด อาร์อาร์อีมีจุดมุ่งหมายอยู่ที่การทำให้การบีบอัดขนาดเป็นไปตามความต้องการของเซิร์ฟเวอร์ แต่อาร์เอฟบีไคลเอนต์ไม่มีประสิทธิภาพเพียงพอในการบีบอัดข้อมูลให้มีความเร็วเพียงพอที่จะรักษาประสิทธิภาพการใช้งาน ได้อย่างทันท่วงที

ความคิดพื้นฐานของ อาร์อาร์อี คือ การแบ่งส่วนของกรอบของข้อมูลพิกเซลให้อยู่ในรูปของสี่เหลี่ยมย่อยๆ แต่ละส่วนย่อยๆนั้นจะประกอบด้วยพิกเซลของพิกเซลค่าเดี่ยวและการรวมของแต่ละชุดของกรอบข้อมูลในตอนแรก การที่เราทำการแบ่งจากกรอบใหญ่ออกเป็นกรอบย่อยๆนั้นเพื่อให้ง่ายต่อการคำนวณ

ผลที่ได้จากการเข้ารหัสจะประกอบด้วย ค่าของพิกเซลที่เป็นแบ็คกราวน์ Vb (โดยปกติจะเป็นพิกเซลที่มีค่ามากที่สุดในกรอบนั้นๆ) และเลขจำนวนนับ N ซึ่งจะใช้เป็นตัวแสดงจำนวนของกรอบย่อยๆ แต่ละอันจะประกอบเข้าด้วยกัน ทัปเปิ้ลของ <v,i,y,w,h> ที่ v (Vb) เป็นค่าของพิกเซล (x,y) เป็นพิกัดของกรอบสี่เหลี่ยมย่อยที่สัมพันธ์กับมุมด้านบนซ้ายของกรอบและ (w,h) เป็นความกว้างความสูงของกรอบสี่เหลี่ยมย่อย ไคลเอนต์สามารถแสดงกรอบที่มาจากต้นฉบับได้โดยการวาดเติมกรอบของค่าพิกเซลพื้นหลังให้เต็มและเมื่อการวาดเติมกรอบนั้นถูกเติมเต็มและตรงกับกรอบสี่เหลี่ยมย่อย

5.4.4 การเข้ารหัสแบบโคอาร์อาร์อี (CoRRE encoding)

การเข้ารหัสแบบโคอาร์อาร์อีเป็นเป็นส่วนที่เพิ่มขึ้นมาจากการเข้ารหัสแบบอาร์อาร์อี ที่สามารถรับประกันได้ว่ากรอบที่ส่งออกไปจะมีขนาดใหญ่ที่สุดไม่เกิน 255 X 255 พิกเซล ถ้าเซิร์ฟเวอร์ต้องการส่งกรอบที่มีขนาดใหญ่กว่านี้ก็จะทำการแยกมันออกเป็นชิ้นๆและส่งออกมาเป็นกรอบอาร์เอฟบีเล็กๆจำนวนหลายๆครั้ง ภายในแต่ละส่วนของกรอบสี่เหลี่ยมเล็กๆเหล่านั้นจะมีหนึ่งไบต์ที่สามารถจะเอาไปใช้ในการแทนมิติของกรอบสี่เหลี่ยมย่อยๆเหล่านี้ สำหรับในคอมพิวเตอร์ทั่วไปผลลัพธ์จะดีกว่าการบีบอัดข้อมูลด้วยวิธีอาร์อาร์อี ในความเป็นจริงแล้วการบีบอัดข้อมูลที่ดีที่สุดเมื่อเรากำหนดขนาดของกล่องสี่เหลี่ยมให้มีขนาดสูงสุด 48 X 48 นั่นเป็นเพราะว่ากรอบข้อมูลไม่ได้ทำการเข้ารหัสไว้ (โดยปกติแล้วข้อมูลภายในจะเป็นข้อมูลประเภทรูปภาพ) จะถูกส่งออกเป็นแบบหยาบ ถ้าอันไหนทำการเข้ารหัสไว้ดีจะถูกส่งออกไปเป็นแบบโคอาร์อาร์อี ตัวที่เล็กที่สุดของขนาดกรอบที่ใหญ่ที่สุดเป็นตัวกำหนดการตัดสินใจ ถ้าเป็น อาร์อาร์อี กรอบเดิมทั้งหมดจะต้องถูกส่งทั้งในแบบ อาร์อาร์อี และแบบหยาบ อย่างไรก็ตามเนื่องจากว่าในการส่งแต่ละครั้งก็จะมีค่าโสหุ้ย ดังนั้นการส่งแบบให้กรอบมีขนาดใหญ่ก็จะจึงมีขนาดของข้อมูลโดยรวมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ดังนั้นจึงสามารถเพิ่มจำนวนของกรอบอาร์เอฟบีเพิ่มขึ้นได้) ทั้งหมดนี้เป็นการคิดในกรณีที่เลวร้ายที่สุดก่อน

5.4.5 การเข้ารหัสแบบเฮ็กซ์ไทล์ (hextile encoding)

เฮ็กซ์ไทล์เป็นการปรับปรุงแนวความคิดของ โคอาร์อาร์อี ใหม่ จากกรอบข้อมูลเดิมจะถูกแบ่งออกเป็น 16X16 ชิ้น และใช้ข้อมูล 4 บิตสำหรับแทนแต่ละกรอบสี่เหลี่ยมย่อย และ16 บิตในการแทนข้อมูลของกรอบรวมจะไม่เหมือนกับแบบ โคอาร์อาร์อี แต่ละชิ้นจะไม่อยู่บนระดับบนของกรอบอาร์เอฟบีเมื่อทำการแตกกรอบตัวเดิมออกเป็นแผ่นย่อยๆวิธีนี้เรียกว่า การทำพรีดีเทอร์มาย นั่นหมายความว่าตำแหน่งและขนาดของแต่ละชิ้นจะไม่มีรูปแบบกำหนดที่แน่นอนตายตัว องค์ประกอบของการเข้ารหัสของแผ่นข้อมูลย่อยที่แตกออกมาเป็นชิ้นนี้จะทำให้ง่ายในลำดับของการทำพรีดีเทอร์มาย ลำดับของชิ้นส่วนย่อยจะเริ่มต้นจากด้านซ้ายมุมบนแล้วไปทางขวาจากบนลงล่างตามลำดับ ถ้าความกว้างของกรอบหารด้วย 16 ไม่ลงตัวให้เอาความกว้างของชิ้นส่วนสุดท้ายในแต่ละแถวแทนจะทำให้ได้เล็กที่สุด ในทางของคอลัมน์ก็ทำเช่นเดียวกันหากความสูงของมันหารด้วย 16 ไม่ลงตัวก็ให้เอาแถวสุดท้ายมาทำเช่นเดียวกัน

แต่ละชิ้นส่วนย่อยจะถูกเข้ารหัสข้อมูลพิกเซลเป็นแบบหยาบหรืออาจจะเป็นแบบอาร์อาร์อีแล้วแต่ข้อกำหนดโดยจะถูกกำหนดโดยไบต์ที่บอกชนิดของแต่ละชิ้นส่วนย่อย แต่ละชิ้นส่วนย่อยจะเก็บค่าพิกเซลของพื้นหลังไว้อย่างไรก็ตามค่าของข้อมูลพิกเซลจะไม่มีความจำเป็นมากนักหากมันมีค่าพื้นหลังเช่นเดียวกับชิ้นก่อนหน้าและค่าพิกเซลของกรอบย่อยมีค่าเหมือนหมด เราสามารถกำหนดค่าหนึ่งแทนจำนวนชิ้นย่อยทั้งหมดที่ซ้ำกันแทนเพื่อลดจำนวนข้อมูลลงได้ ส่วนข้อมูลที่เป็นภาพพื้นหน้าสามารถนำมาจากชิ้นส่วนก่อนหน้าได้

5.5 เมสเซจโพรโตคอล (Protocol Messages)

อาร์เอฟบีโพรโตคอลสามารถทำงานบนระดับชั้นทราสพอร์ตที่เชื่อถือไดๆก็ได้ทั้งที่เป็นแบบไบต์สตรีม (byte-stream) หรือแบบเมสเซจ (Messages base) ในโพรโตคอลนี้มีขั้นตอนใหญ่ๆคือ ในขั้นตอนการเริ่มต้นการเชื่อมต่อ ที่เรียกว่า “ อินนิเชียลแฮนด์เชคคิ่ง (initial handshaking) “ ซึ่งกำหนดโดยโพรโตคอลที่ใช้ในการโต้ตอบปกติทั่วไป

อินนิเชียลแฮนด์เชคคิ่งประกอบด้วย โพรโตคอลเวอร์ชัน (ProtocolVersion), การรับรองสิทธิ (Authentication), การเริ่มต้นการเชื่อมต่อของไคลเอนต์ (ClientInnitialisation), การเริ่มต้นการเชื่อมต่อของเซิร์ฟเวอร์(ServerInitialisation) ควรจำไว้ว่าทั้งไคลเอนต์และเซิร์ฟเวอร์จะส่งเมสเซจ ProtocolVersion มาด้วย

โพรโตคอลจะทำงานในสถานะปกติหลังจากมีการส่งอินนิเชียลไลเซชัน (initialisation) เมสเซจหลังจากขั้นตอนนี้ไคลเอนต์ก็จะสามารถส่งเมสเซจอะไรก็ได้ตามที่มันต้องการหรืออาจรับเมสเซจจากเซิร์ฟเวอร์ก็ได้ เมสเซจเหล่านี้จะต้องเริ่มต้นด้วยไบต์ที่บอกชนิดของเมสเซจ ตามด้วยข้อมูลชนิดพิเศษอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.1 เมสเซจเริ่มต้นการเชื่อมต่อ (Initail Handshaking Messages)

5.5.1.1 โพรโตคอลเวอร์ชัน (ProtocolVersion)

การเริ่มต้นการเชื่อมต่อจะเริ่มต้นด้วยการที่เซิร์ฟเวอร์ส่งโพรโตคอลเวอร์ชันไปยัง ไคลเอนต์ ในการส่งเมสเซจนี้เครื่องไคลเอ็นจะต้องรู้เลขเวอร์ชันล่าสุดของอาร์เอฟบีโพรโตคอลที่ตัวเซิร์ฟเวอร์นั้นก่อน ซึ่งมันจะสนับสนุนการทำงานนั้นอยู่แล้วโดยฝั่งไคลเอนต์จะส่งเมสเซจที่ให้เลขเวอร์ชันที่มันใช้อยู่กลับมายังเซิร์ฟเวอร์ (ซึ่งอาจจะแตกต่างกับเวอร์ชันของเซิร์ฟเวอร์ก็ได้)

อาจจะมีการตั้งใจทำให้ทั้งฝั่งไคลเอนต์และฝั่งเซิร์ฟเวอร์มีการเตรียมระดับของการเข้ากันได้ของทั้งสองฝั่งกับเวอร์ชันเก่าและควรเข้ากันได้กับเวอร์ชันใหม่ด้วย ตัวอย่างเช่น ถ้าไคลเอนต์ต้องการใช้โพรโตคอลเวอร์ชัน 3.1 เซิร์ฟเวอร์ที่ใช้เวอร์ชัน 3.0 ต้องสามารถจัดการได้โดยการไม่สนใจในสิ่งที่ไม่เข้าใจในโพรโตคอลที่ต่างกันแต่มันยังสามารถทำงานได้ในกรณีเช่นนี้ไม่เกี่ยวกับกรณีที่มีการเปลี่ยนแปลงเลขเวอร์ชันครั้งใหญ่ๆ

โพรโตคอลเวอร์ชันเมสเซจจะประกอบด้วย 12 ไบต์ ที่สามารถแปลงเป็นสตริงในแอสกีเป็นรูปแบบ “RFB xxx.yyy\n” ที่ xxx และ yyy เป็นเลขตัวหลักและตัวรองของเลขเวอร์ชัน ตามด้วยส่วนเติมเต็มด้วย 0

No. of bytes	Value
12	"RFB 003.003\n" (hex 52 46 42 20 30 30 33 2e 30 30 33 0a)

5.5.1.2 การกำหนดขอใช้สิทธิการใช้งาน (Authentication)

เมื่อจะใช้โพรโตคอลนี้เซิร์ฟเวอร์จะต้องส่งเมสเซจที่แสดงว่ามีการรับรองสิทธิที่ใช้ในการเชื่อมต่อด้วย

No. of bytes	Type	[Value]	Description
4	CARD32		*authentication-scheme:*
		0	*connection failed*
		1	*no authentication*
		2	*VNC authentication*

- Connection failed หากมีเหตุผลบางอย่างที่ทำให้การเชื่อมต่อผิดพลาด จะต้องตามด้วยสตริงที่อธิบายเหตุผลของการผิดพลาดนั้น
- No authentication ไม่มีการกำหนดการขอใช้สิทธิการใช้งาน
- VNC authentication มีการกำหนดการขอใช้สิทธิ จะตามด้วยเลข 16 ไบต์ที่ทำการสุ่มมา

No. of bytes	Type	[Value]	Description
16	CARD8		*challenge*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไคลเอนต์จะต้องเข้ารหัสการร้องขอด้วยการเข้ารหัสแบบ DES โดยการใช้รหัสผ่านที่ผู้ใช้ป้อนเป็นคีย์และส่งผลลัพธ์เป็น 16 ไบต์

No. of bytes	Type [Value]	Description
16	CARD8	*response*

เซิร์ฟเวอร์จะส่งเมสเซจที่แจ้งให้ไคลเอนต์ทราบว่าการขอใช้สิทธินั้นสำเร็จ ถ้าการเชื่อมต่อสำเร็จโพรโตคอลจะต้องตามด้วยเมสเซจ Initialisation ถ้าหากไม่สำเร็จก็จะตัดการเชื่อมต่อ

No. of bytes	Type [Value]	Description
4	CARD32	*status:*
	0	*OK*
	1	*failed*
	2	*too-many*

ถ้าหากในการขอใช้สิทธินั้นผิดพลาดหลายครั้งติดต่อกันโพรโตคอลไม่ควรจะอนุญาตให้มีการเชื่อมต่อจากไคลเอนต์ตัวนั้นอีก

5.5.1.3 ClientInitailisation

ในการเชื่อมต่อนั้นไคลเอนต์จะต้องส่งเมสเซจที่แสดงการเริ่มต้นการเชื่อมต่อ

No. of bytes	Type [Value]	Description
1	CARD8	*shared-flag*

5.5.1.4 ServerInitialisation

หลังจากมีการรับเมสเซจClientInitialisationจากไคลเอนต์แล้วเซิร์ฟเวอร์จะต้องส่ง ServerInitialisation ออกไปด้วยซึ่งเมสเซจตัวนี้จะบอกความกว้าง ความยาวของเฟรมบัฟเฟอร์,รูปแบบของพิกเซลและชื่อที่เกี่ยวพันกับเดสก์ท็อปของมัน

No. of bytes	Type [Value]	Description
2	CARD16	*framebuffer-width*
2	CARD16	*framebuffer-height*
16	PIXEL_FORMAT	*server-pixel-format*
4	CARD32	*name-length*
name-length	CARD8 array	*name-string*

ที่ pixel-format เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No. of bytes	Type [Value]	Description
1	CARD8	*bits-per-pixel*
1	CARD8	*depth*
1	CARD8	*big-endian-flag*
1	CARD8	*true-colour-flag*
2	CARD16	*red-max*
2	CARD16	*green-max*
2	CARD16	*blue-max*
1	CARD8	*red-shift*
1	CARD8	*green-shift*
1	CARD8	*blue-shift*
3		*padding*

server-pixel-format จะเป็นตัวกำหนดรูปแบบพิกเซลตามปกติของเซิร์ฟเวอร์ รูปแบบพิกเซลจะถูกนำมาใช้ก็ต่อเมื่อไคลเอนต์มีการเรียกร้องรูปแบบที่แปลกออกไป โดยการส่งเมสเซจ SetPixelFormat

จำนวนบิตต่อพิกเซล (bit-per-pixel) เป็นจำนวนบิตที่ใช้บนพิกเซลหนึ่งๆซึ่งมันจะต้องมากกว่าหรือเท่ากับความกว้างของกรอบที่ถูกส่งมา ซึ่งบิตเหล่านี้จะเป็นตัวบอกค่าความละเอียดของพิกเซล โดยค่านี้จะต้องมีค่า 8,16 หรือ 32 และระบบจะไม่สนับสนุนการทำงานหากมีค่าน้อยกว่า 8 บิตลงมา

ในกรณีที่บิ๊กเอนเดียนแฟล็ก (Big-endian-flag) ไม่เป็น 0 (เป็นจริง) ถ้าพิกเซลที่ใช้หลายๆไบต์เป็นบิ๊กเอนเดียน อาจแปลได้ว่า ใช้ 8 บิตต่อพิกเซล

ถ้า ทรู-คัลเลอร์-แฟล็ก (true-colour-flag) ไม่เป็น 0 (เป็นจริง) ให้เอาหกตัวสุดท้ายเป็นตัวกำหนดความคมชัดของสีต่างๆในแต่ละพิกเซล

เรดแมกซ์ (Red-max) เป็นค่าสูงสุดของสีแดง (= $2^n - 1$ ที่ n เป็นจำนวนบิตของสีแดง) ข้อควรจำ ค่าเหล่านี้จะเก็บในรูปแบบบิ๊กเอนเดียน

เรดชิพ (Red-shift) เป็นจำนวนครั้งในการเลื่อนของสีแดง กรีนแมกซ์ (Green-max) กรีนชิพ (Green-shift) บลูแมกซ์ (Blue-max) บลูชิพ (Blue-shift) ก็เป็นแบบเดียวกับสีแดง ตัวอย่าง หาค่าของสีแดง (ระหว่าง 0 ถึง ค่าสูงสุดของสีแดง) จากพิกเซลที่ให้ วิธีทำดังนี้

- สลับค่าของพิกเซลให้สอดคล้องกับบิ๊กเอนเดียนแฟล็ก (ตัวอย่างเช่น ถ้าบิ๊กเอนเดียนเป็น 0 และลำดับไบต์ของโฮสต์เป็นบิ๊กเอนเดียน ให้ทำการสลับค่าได้)
- ให้ทำการเลื่อนเรด-ชิพ ไปทางขวา
- ทำการ AND ด้วย เรด-แมกซ์ (ในลำดับไบต์ของโฮสต์)

ในตอนนี้มีจำนวนน้อยหรือเกือบไม่มีเลยที่สนับสนุนการทำงานของตารางสี บางงานอาจจะทำได้แต่จะไม่สมบูรณ์ มันจะต้องเพิ่มข้อกำหนดบางอย่างดังนี้

ถ้า ทรู-คัลเลอร์-แฟล็ก เป็น 0 (เป็นเท็จ) ให้เซิร์ฟเวอร์ใช้ค่าพิกเซลที่มันจะไม่รวมเอาความเข้มของสีแดง น้ำเงิน และเขียวไว้ แต่มันจะทำการชี้ไปยังแผนที่สีแทน ซึ่งค่าที่อยู่ในแผนที่สีสามารถที่จะเซตโดยไคลเอนต์โดยใช้ FixColourMapEntries เมสเซจ หรือโดยเซิร์ฟเวอร์ โดยใช้ SetColourMapEntries

แต่ FixColourMapEntries เมสเซจนั้นไม่สนับสนุนการทำงานโดยเซิร์ฟเวอร์ใดๆที่ใช้งานอยู่ในปัจจุบันนี้และแผนที่สีจะสนับสนุนการทำงานทั้งหมดโดยเซิร์ฟเวอร์แบบ X-base เท่านั้น ในความเป็นจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วการสนับสนุนการทำงานของแผนที่สีนั้นไคลเอนต์จำเป็นจะต้องมีความสามารถที่จะกำหนดค่าพิกเซลเฉพาะอันที่ไม่ต้องการให้เซิร์ฟเวอร์ใช้ได้ซึ่งคุณสมบัตินี้อาจถูกเพิ่มเข้าไปในเวอร์ชันถัดไปขอโพรโตคอล แต่ตอนนี้ยังไม่ได้เพิ่มคุณสมบัติข้อนี้เข้าไป

5.5.2 เมสเซจที่ส่งจากไคลเอนต์ไปยังเซิร์ฟเวอร์ (Client to server message)

5.5.2.1 SetPixelFormat

เป็นการกำหนดรูปแบบค่าของพิกเซลที่ควรจะทำการส่งไปในเมสเซจ FrameBufferUpdate ถ้าไคลเอนต์ไม่สามารถส่งเมสเซจ SetPixelFormat ได้ ให้เซิร์ฟเวอร์ใช้รูปแบบค่าของพิกเซลตามที่เคยใช้อยู่ก่อนหน้าโดยจะถูกกำหนดในเมสเซจ ServerInitialisation

ณ ตอนนี้ (1998) มีคอมพิวเตอร์จำนวนน้อยที่ไม่สนับสนุนแผนที่สีหรือทำงานแต่ทำได้ไม่สมบูรณ์จึงต้องมีการเพิ่มบางส่วนดังนี้

ถ้า true-colour-flag เป็น 0 ให้นำเอาแผนที่สีนั้นมาใช้งานได้ ตัวไคลเอนต์เองสามารถกำหนดตัวที่จะให้คงที่ภายในแผนที่สีได้โดยใช้เมสเซจ FixColourMapEntries ตัวอื่นๆที่ไม่ได้ถูกทำให้มีค่าคงที่โดยไคลเอนต์สามารถที่จะถูกปรับเปลี่ยนค่าได้โดยเซิร์ฟเวอร์ได้โดยใช้เมสเซจ SetColourMapEntries

ในทันทีที่ไคลเอนต์มีการส่งเมสเซจนี้ออกไปแผนที่สีก็จะว่างเปล่าถ้าก่อนหน้านี้มีการกำหนดให้มันมีค่าคงที่โดยไคลเอนต์หรือเซิร์ฟเวอร์ตัวใดตัวหนึ่ง

No. of bytes	Type	[Value]	Description
1	CARD8	0	*message-type*
3			*padding*
16	PIXEL_FORMAT		*pixel-format*

ที่ PIXEL_FORMAT ถูกอธิบายไว้ในหัวข้อ เรื่อง Pixel Format แล้ว

No. of bytes	Type [Value]	Description
1	CARD8	*bits-per-pixel*
1	CARD8	*depth*
1	CARD8	*big-endian-flag*
1	CARD8	*true-colour-flag*
2	CARD16	*red-max*
2	CARD16	*green-max*
2	CARD16	*blue-max*
1	CARD8	*red-shift*
1	CARD8	*green-shift*
1	CARD8	*blue-shift*
3		*padding*

5.5.2.2 FixColourMapEntries

ในเวลานี้มีคอมพิวเตอร์จำนวนน้อยที่ไม่สนับสนุนแผนที่สี หรือทำงานได้แต่ทำได้ไม่สมบูรณ์จึงต้องมีการเพิ่มบางส่วนดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อรูปแบบพิกเซลใช้แผนที่สีนี้ เมสเซจตัวนี้จะบอกเซิร์ฟเวอร์ให้กำหนดค่าพิกเซลให้เพื่อทำการแมพกับค่าในแผนที่สีสำหรับกำหนดความสว่างของRGBนั่นหมายความว่าเซิร์ฟเวอร์อาจจะไม่ได้กำหนดความสว่างของค่าRGBเองสำหรับค่าความสว่างของพิกเซลเหล่านี้เราสามารถทำการกำหนดได้โดยใช้เมสเซจ SetColourMapEntries ซึ่งถ้าไคลเอนต์มีการกำหนดแผนที่สีตายตัวมันจะต้องทำการส่งเมสเซจ FixColourMapEntries ซึ่งจะอธิบายแผนที่สีของมันทั้งหมดที่มันต้องการและเซิร์ฟเวอร์จะแปลงค่าพิกเซลทั้งหมดไปเป็นค่าต่างๆ ที่ต้องใช้ในการเข้ารหัส สำหรับแผนที่สีของไคลเอนต์ในกรณีเช่นนี้เซิร์ฟเวอร์จะไม่สามารถส่งเมสเซจ SetColourMapEntries ออกไปได้เลยเพราะไคลเอนต์ได้กำหนดไว้ก่อนแล้วเมสเซจ FixcolourMapEntries จะไม่มีการสนับสนุนการใช้งานโดยเซิร์ฟเวอร์ที่มีใช้อยู่ในตอนนี้เลย

No. of bytes	Type	[Value]	Description
1	CARD8	1	*message-type*
1			*padding*
2	CARD16		*first-colour*
2	CARD16		*number-of-colours*

number-of-colours มีลักษณะดังนี้

No. of bytes	Type	[Value]	Description
2	CARD16		*red*
2	CARD16		*green*
2	CARD16		*blue*

5.5.2.3 SetEncodings

เป็นการกำหนดชนิดของการเข้ารหัสของพิกเซลเดทาที่ส่งมาโดยเซิร์ฟเวอร์ลำดับของชนิดการเข้ารหัสจะถูกเก็บอยู่ในเมสเซจนี้โดยจะบอกโดยคร่าวๆเท่านั้นโดยไคลเอนต์จะนำค่านี้ไปใช้ในการอ้างอิงถึง(โดยการเข้ารหัสที่มาเป็นตัวแรกจะถูกกำหนดให้มีความสำคัญในการอ้างอิงถึงก่อนมากที่สุด)โดยเซิร์ฟเวอร์อาจจะใช้หรือไม่ใช้คำบอกใบ้นี้ก็ได้ ข้อมูลของพิกเซลจะถูกเข้ารหัสเป็นการเข้ารหัสแบบหยาบถ้าหากว่าไม่มีการกำหนดไว้ก่อน

No. of bytes	Type	[Value]	Description
1	CARD8	2	*message-type*
1			*padding*
2	CARD16		*number-of-encodings*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

number-of-encodings จะถูกกำหนดดังข้างล่าง

No. of bytes	Type [Value]	Description
4	CARD32	*encoding-type*
	0	*raw* encoding
	1	*copy rectangle* encoding
	2	*RRE* encoding
	4	*CoRRE* encoding
	5	*hextile* encoding

5.5.2.4 FramebufferUpdateRequest

เป็นการเปลี่ยนแปลงค่าต่างๆภายในเครื่องเซิร์ฟเวอร์นั้นเครื่องไคลเอนต์จะสนใจเฉพาะในพื้นที่ของเฟรมบัฟเฟอร์ซึ่งจะถูกกำหนดโดย ตำแหน่งของ เอ็กซ์,วาย ความกว้างและความสูง โดยปกติแล้วเซิร์ฟเวอร์จะตอบสนองกับ FramebufferUpdateRequest โดยการส่ง FramebufferUpdate อย่างไรก็ตามการส่ง FramebufferUpdate ครั้งเดียวอาจมีการส่ง FramebufferUpdateRequests เพื่อตอบสนองหลายครั้งก็ได้

เซิร์ฟเวอร์จะกำหนดให้ไคลเอนต์ทำการเก็บสำเนาของตัวที่มีการเปลี่ยนแปลงทั้งหมดในเฟรมบัฟเฟอร์หมายความว่าเซิร์ฟเวอร์จำเป็นจะต้องส่งข้อมูลที่มีการปรับปรุงข้อมูลไปยังเซิร์ฟเวอร์บ่อยๆ

อย่างไรก็ตามถ้ามีการสูญหายของข้อมูลที่จำเป็นเกิดขึ้นระบบจะให้ไคลเอนต์ส่ง FramebufferUpdateRequest ใหม่และทำการเซต incremental เป็น 0 (เป็นเท็จ) จากการร้องขอข้อมูลครั้งนี้เซิร์ฟเวอร์จะส่งข้อมูลทั้งหมดของพื้นที่ที่ถูกกำหนดไว้ให้เร็วที่สุดเท่าที่จะเป็นไปได้ พื้นที่เหล่านี้จะไม่ทำการปรับปรุงข้อมูลและส่งข้อมูลโดยการใช้การเข้ารหัสแบบ copy rectangle แน่นอน

ถ้าไคลเอนต์ไม่มีการสูญหายของข้อมูลของพื้นที่ที่มีการเปลี่ยนแปลงเกิดขึ้นให้มันทำการส่ง FramebufferUpdateRequests ใหม่และทำการเซตให้ incremental ไม่เป็น 0 (เป็นจริง) ถ้าเมื่อไหร่ก็ตามที่มีการเปลี่ยนแปลงของพื้นที่ที่ถูกกำหนดไว้ของเฟรมบัฟเฟอร์ เซิร์ฟเวอร์จะส่ง FramebufferUpdate ดังนั้นอาจจะไม่มีการกำหนดระยะเวลาการทำการปรับปรุงข้อมูลระหว่าง FramebufferUpdateRequests และ FramebufferUpdate

ในกรณีของไคลเอนต์แบบทำงานเร็วมันอาจจะเลือกการส่ง FramebufferUpdateRequests ในอัตราความเร็วที่ปกติเพื่อหลีกเลี่ยงการใช้เน็ตเวิร์กเกินส่วนที่จะใช้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No. of bytes	Type	[Value]	Description
1	CARD8	3	*message-type*
1	CARD8		*incremental*
2	CARD16		*x-position*
2	CARD16		*y-position*
2	CARD16		*width*
2	CARD16		*height*

5.5.2.5 KeyEvent

เป็นการกดหรือปล่อยปุ่ม โดย Down-flag จะไม่เป็น 0 (เป็นจริง) ถ้าปุ่มกำลังถูกกด และเป็น 0 (ไม่เป็นจริง) เมื่อปุ่มถูกปล่อย โดยค่าของปุ่มจะถูกกำหนดค่าโดยเรียกว่า "keysym" ค่าของปุ่มโดยปกติแล้วจะถูกกำหนดในค่ารหัส ASCII โดยค่าปุ่มทั่วๆไปมีค่าดังนี้

Key name	Keysym value
BackSpace	0xff08
Tab	0xff09
Return or Enter	0xff0d
Escape	0xff1b
Insert	0xff63
Delete	0xffff
Home	0xff50
End	0xff57
Page Up	0xff55
Page Down	0xff56
Left	0xff51
Up	0xff52
Right	0xff53
Down	0xff54

Key name	Keysym value
F1	0xffbe
F2	0xffbf
F3	0xffe0
F4	0xffe1
...	...
F12	0xffe9
Shift (left)	0xffe1
Shift (right)	0xffe2
Control (left)	0xffe3
Control (right)	0xffe4
Meta (left)	0xffe7
Meta (right)	0xffe8
Alt (left)	0xffe9
Alt (right)	0xffea

5.5.2.6 PointerEvent

เป็นตัวแสดงการเคลื่อนที่ของตัวชี้หรือการกดหรือปล่อยปุ่มของตัวชี้ โดยตัวชี้จะมีค่าตำแหน่งเอ็กซ์ ตำแหน่งวาย และสถานะปัจจุบันของปุ่มมีค่าตั้งแต่ 1 ถึง 8 โดยจะใช้บิต 0 ถึง 7 ของ button-mask ตามลำดับ โดย 0 หมายถึงขึ้น 1 หมายถึงลง (ถูกกด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No. of bytes	Type	[Value]	Description
1	CARD8	5	*message-type*
1	CARD8		*button-mask*
2	CARD16		*x-position*
2	CARD16		*y-position*

5.5.2.7 ClientCuttext

ตัวไคลเอนต์เองจะมีตัวอักษรอยู่ในบัฟเฟอร์ตัด (cut-buffer) ของมันเอง ตัวจบบรรทัดจะแทนด้วย แอสกีค่า10 และไม่จำเป็นต้องมี carriage-return

No. of bytes	Type	[Value]	Description
1	CARD8	6	*message-type*
3			*padding*
4	CARD32		*length*
length	CARD8 array		*text*

5.5.3 Server to client Messages

5.5.3.1 FramebufferUpdate

เป็นการปรับปรุงเฟรมบัฟเฟอร์ ซึ่งจะประกอบจากลำดับการส่งกรอบข้อมูลพิกเซล ฝั่งไคลเอนต์จะทำการเก็บข้อมูลของพิกเซลเหล่านั้นไว้ในเฟรมบัฟเฟอร์ของมันเองโดยมันจะทำการตอบสนองกับเมสเซจFramebufferUpdateRequests จากไคลเอนต์ซึ่งอาจจะไม่มีการกำหนดช่วงเวลาการตอบสนองระหว่าง FramebufferUpdateRequests และ FramebufferUpdate

No. of bytes	Type	[Value]	Description
1	CARD8	0	*message-type*
1			*padding*
2	CARD16		*number-of-rectangles*

number-of-rectanglesจะเป็นตัวบอกจำนวนกรอบของข้อมูลพิกเซลโดยแต่ละกรอบจะประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No. of bytes	Type	[Value]	Description
2	CARD16		*x-position*
2	CARD16		*y-position*
2	CARD16		*width*
2	CARD16		*height*
4	CARD32		*encoding-type*
		0	*raw* encoding
		1	*copy rectangle* encoding
		2	*RRE* encoding
		4	*CoRRE* encoding
		5	*hextile* encoding

ตารางดังข้างล่างนี้จะเป็นตัวกำหนดชนิดของการเข้ารหัส

- ในการเข้ารหัสแบบ raw ข้อมูลจะประกอบจาก ความกว้าง X ความสูง ของค่าพิกเซล ซึ่งค่าเหล่านี้จะแทนแต่ละพิกเซลในการแสกนเส้นจากซ้ายไปขวาตามลำดับ
- การเข้ารหัสแบบ copy rectangles ข้อมูลจะประกอบด้วย

No. of bytes	Type	[Value]	Description
2	CARD16		*src-x-position*
2	CARD16		*src-y-position*

- การเข้ารหัสแบบ RRE ข้อมูลจะมีส่วนหัวดังนี้

No. of bytes	Type	[Value]	Description
4	CARD32		*number-of-subrectangles*
n	CARD<8n>		*background-pixel-value*

ที่ 8n คือ จำนวนบิตต่อหนึ่งพิกเซลที่มีการตกลงระหว่างไคลเอนต์และเซิร์ฟเวอร์ ทั้งที่อยู่ในเมสเซจ ServerInitialisation และ เมสเซจ SetPixelFormat โดยตัวอย่างข้างล่างจะเป็นโครงสร้างของ number-of-subrectangles

No. of bytes	Type	[Value]	Description
n	CARD<8n>		*subrect-pixel-value*
2	CARD16		*x-position*
2	CARD16		*y-position*
2	CARD16		*width*
2	CARD16		*height*

- การเข้ารหัสแบบ CoRRE ข้อมูลจะมีส่วนหัวดังนี้

No. of bytes	Type	[Value]	Description
4	CARD32		*number-of-subrectangles*
n	CARD<8n>		*background-pixel-value*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ 8n เป็น จำนวนบิตต่อหนึ่งพิกเซลซึ่งถูกตกลงโดยไคลเอนต์และเซิร์ฟเวอร์ไว้แล้วกำหนดได้ทั้งเมสเซจ ServerInitialisation หรือ เมสเซจ SetPixelFormat

โดย number-of-subrectangles มีตัวอย่างโครงสร้างดังข้างล่าง

No. of bytes	Type [Value]	Description
n	CARD<8n>	*subrect-pixel-value*
1	CARD8	*x-position*
1	CARD8	*y-position*
1	CARD8	*width*
1	CARD8	*height*

- การเข้ารหัสแบบ hextile กรอบจะถูกแบ่งออกเป็นชิ้นๆละ 16X16 พิกเซล โดยจะเริ่มต้นจากบนซ้ายโดยจะวิ่งจากซ้ายไปขวาจากบนลงล่างตามลำดับ ถ้าความกว้างของกรอบไม่หารด้วย 16 ลงตัว ให้ความกว้างของชิ้นสุดท้ายในแต่ละแถวมีขนาดเล็กลงในทำนองเดียวกันหากความสูงหารด้วย 16 ไม่ลงตัวชิ้นสุดท้ายในแต่ละแถวจะต้องมีขนาดเล็กลงเช่นเดียวกัน แต่ละชิ้นของกรอบจ้อมูลย่อยจะต้องเริ่มต้นด้วยไบต์ subencoding ซึ่งภายในจะมีบิตเหล่านี้

No. of bytes	Type [Value]	Description
1	CARD8	*subencoding-mask:*
	1	*Raw*
	2	*BackgroundSpecified*
	4	*ForegroundSpecified*
	8	*AnySubrects*
	16	*SubrectsColoured*

ถ้าบิต Raw มีการเซตบิตอื่นๆจะไม่ได้รับความสนใจอีกเลย

BackgroundSpecified ถ้ามีการเซตบิตนี้ ค่าพิกเซลของข้อมูลชิ้นนั้นจะถูกกำหนดสีพื้นหลังไว้ด้วย

No. of bytes	Type [Value]	Description
n	CARD<8n>	*background-pixel-value*

ที่ 8n คือจำนวนบิตต่อหนึ่งพิกเซลที่ตกลงกันไว้ระหว่างไคลเอนต์กับเซิร์ฟเวอร์โดยใช้เมสเซจ ServerInitialisation หรือเมสเซจ SetPixelFormat ชิ้นแรกของข้อมูลที่เข้ารหัสที่ไม่ใช่การเข้ารหัสแบบ raw จะต้องมีการเซตบิตนี้ก่อน ถ้าไม่มีการเซตบิตนี้ก่อนพื้นหลังจะเหมือนกับชิ้นสุดท้ายก่อนหน้านี้

ForegroundSpecified ถ้ามีการเซตบิตนี้ค่าพิกเซลที่ตามมาจะเป็นตัวกำหนดสีพื้นหน้าเพื่อใช้สำหรับทุกกรอบย่อยในชิ้นนั้น

No. of bytes	Type [Value]	Description
n	CARD<8n>	*foreground-pixel-value*

ถ้าหากมีการเซตบิตนี้ SubrecsColoured จะต้องเป็น 0

AnySubrects ถ้าถูกเซต ค่าหนึ่งไบต์ที่ตามมาจะเป็นตัวบอกจำนวนของกรอบย่อยมีรูปแบบตามข้างล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No. of bytes	Type [Value]	Description
1	CARD8	*number-of-subrectangles*

SubrectsColoured ถ้าถูกเซตแต่ละกรอบย่อยจะถูกนำด้วยค่าของพิกเซลที่จะเป็นตัวกำหนดสีของกรอบย่อยนั้นๆ ดังนั้นกรอบย่อยจึงเป็น

No. of bytes	Type [Value]	Description
n	CARD<8n>	*subrect-pixel-value*
1	CARD8	*x-and-y-position*
1	CARD8	*width-and-height*

ถ้าไม่ถูกเซตกรอบย่อยทั้งหมดจะมีสีเดียวกันสำหรับสีพื้นหลังถ้าบิต ForegroundSpecified ไม่ถูกเซตให้ค่าของพื้นหน้าเหมือนกับชิ้นสุดท้าย กรอบย่อยจะมีค่า

No. of bytes	Type [Value]	Description
1	CARD8	*x-and-y-position*
1	CARD8	*width-and-height*

ตำแหน่งและขนาดของแต่ละกรอบย่อยจะถูกกำหนดในสองไบต์ทั้ง x-and-y-position และ width-and-height สี่บิตที่มีความสำคัญสูงสุดของ x-and-y-position กำหนดตำแหน่งของ X ตัวที่มีความสำคัญน้อยที่สุดกำหนดตำแหน่ง Y สี่บิตที่มีความสำคัญสูงสุดของ width-and-height กำหนดความกว้างบวกด้วยหนึ่ง และบิตต่ำลงมากำหนดความกว้างบวกด้วยหนึ่ง

5.5.3.2 SetColourMapEntries

ปัจจุบันนี้มีคอมพิวเตอร์จำนวนน้อยหรือเกือบไม่มีเลยที่สนับสนุนการทำงานของแผนที่สี หรืออาจจะทำงานแต่ไม่สมบูรณ์ จึงมีการเพิ่มบางอย่างลงไปดังนี้

เมื่อรูปแบบของพิกเซลเรียกใช้ แผนที่สี เมสเซจนี้จะบอกให้เครื่องไคลเอนต์กำหนดค่าของสีที่ควรจะทำการแมพไปเป็นความเข้มของสี RGB โดยเซิร์ฟเวอร์อาจจะกำหนดค่าของพิกเซลซึ่งไคลเอ็ท์อาจจะไม่พร้อมสำหรับการเซตความสว่างของ RGB โดยใช้ FixColourMapEntries

ปัจจุบันนี้ระบบที่สนับสนุนการทำงานของแผนที่สีทั้งหมดมีเพียงเซิร์ฟเวอร์ของ X-system เท่านั้น

No. of bytes	Type [Value]	Description
1	CARD8 1	*message-type*
1		*padding*
2	CARD16	*first-colour*
2	CARD16	*number-of-colours*

ข้างล่างจะเป็นรายละเอียดของ number-of-colours

No. of bytes	Type [Value]	Description
2	CARD16	*red*
2	CARD16	*green*
2	CARD16	*blue*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.3.3 Bell

คือการกำหนดเสียงกริ่งของเครื่องไคลเอนต์

No. of bytes	Type	[Value]	Description
1	CARD8	2	*message-type*

5.5.3.4 ServerCutText

ตัวไคลเอนต์เองจะมีตัวข้ออักษรอยู่ในบัฟเฟอร์ตัด (cut-buffer) ของมันเอง ตัวจบบรรทัดจะแทนด้วย แอสกีค่า10 และไม่จำเป็นต้องมี carriage-return

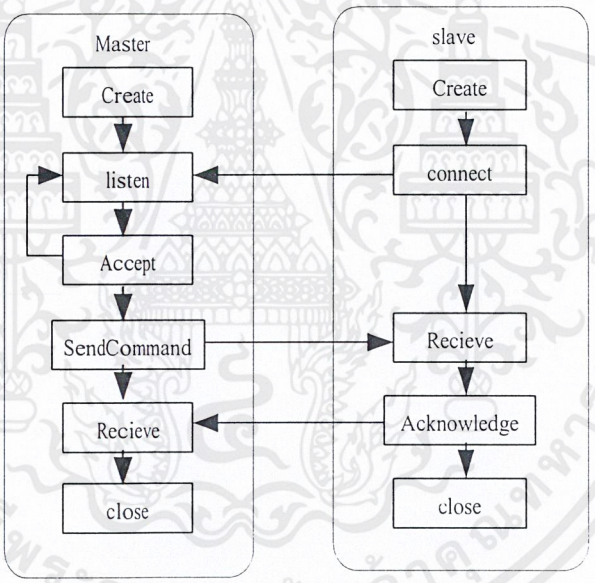
No. of bytes	Type	[Value]	Description
1	CARD8	3	*message-type*
3			*padding*
4	CARD32		*length*
length	CARD8 array		*text*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การออกแบบและการพัฒนาซอฟต์แวร์

6.1 การออกแบบซอฟต์แวร์

ซอฟต์แวร์ตัวนี้ประกอบด้วย 2 ส่วนสำหรับใช้ติดตั้ง คือ ส่วนที่เป็นตัวขอบริการในที่นี้จะเรียกว่าเครื่องมาสเตอร์ (Master) หรือเครื่องแม่ข่ายเป็นเครื่องที่ทำหน้าที่ควบคุมหรือใช้เฝ้าดูการทำงานของเครื่องเป้าหมายซึ่งในเครื่องเป้าหมายจะต้องมีโปรแกรมที่ทำหน้าที่ส่งข้อมูลที่มีการเปลี่ยน หรือส่งเฟรมมาให้กับตัวมาสเตอร์นี้ และฝั่งให้บริการจะเรียกว่าเครื่องลูกข่าย (Slave) ที่จะทำการส่งข้อมูลที่มาสเตอร์ต้องการ ตัวโปรแกรมทั้งสองนี้จะทำงานร่วมกันโดยทำงานเป็นแบบไคลเอ็นต์ เซิร์ฟเวอร์



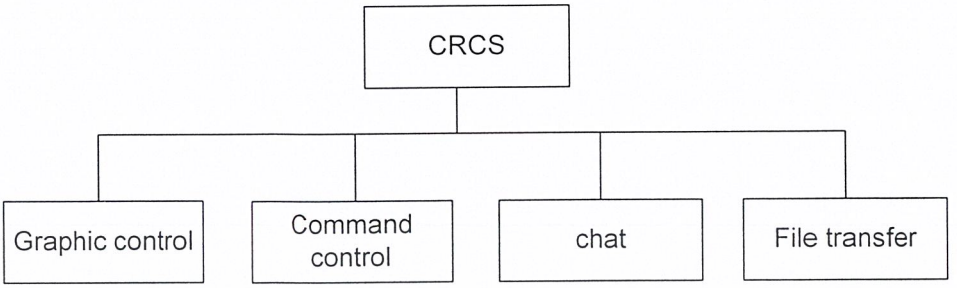
รูปที่ 6-1 สถานะการทำงานของโปรแกรม

ชุดโปรแกรมนี้จะประกอบด้วยส่วนประกอบย่อยที่ประกอบเข้าด้วยกันเพื่อให้ชุดโปรแกรมนี้มีความสามารถที่จะตอบสนองความจำเป็นพื้นฐานในการจัดการกับเครื่องปลายทางที่ต้องการ แต่ส่วนหลักของชุดโปรแกรมนี้จะอยู่ที่ส่วนของตัวที่ทำการเข้าควบคุมการทำงานของเครื่องลูกข่ายโดยจะมีลักษณะโดยรวมดังนี้

- โปรแกรมควบคุมเครื่องลูกข่ายผ่านกราฟิกโหมด
- โปรแกรมควบคุมเครื่องลูกข่ายผ่านเท็กซ์โหมด
- โปรแกรมสนทนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

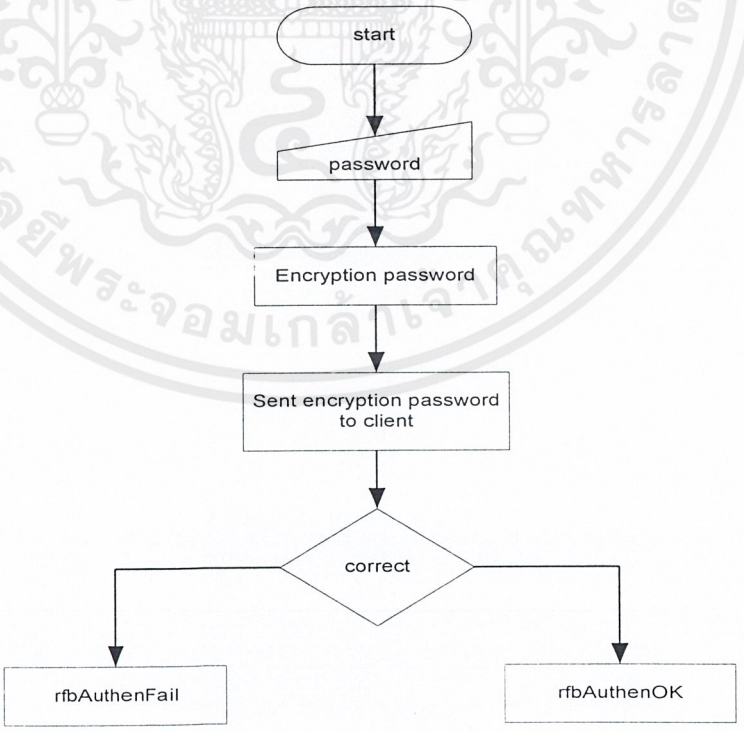
- โปรแกรมโอนย้ายข้อมูล



รูปที่ 6-2 ส่วนประกอบหลักของโปรแกรม

6.2การพิสูจน์ตน

ในส่วนของการอนุญาตให้เครื่องมาสเตอร์เข้าควบคุมเครื่องลูกข่ายนั้นจะต้องมีการพิสูจน์ตนว่าผู้ที่เข้ามาทำการติดต่อเพื่อเข้าควบคุมนั้นมีสิทธิในการเข้าควบคุมจริง เพื่อป้องกันผู้อื่นที่ใช้โปรแกรมฝั่งของมาสเตอร์สวมรอยมาเข้าควบคุมโดยการทำการขออนุญาตก่อนโดยการตรวจสอบรหัสผ่านหากรหัสผ่านที่ได้ตั้งไว้ถูกต้องก็จะยอมอนุญาตให้เครื่องมาสเตอร์เครื่องนั้นสามารถเข้ามาทำการควบคุมได้ แต่ถ้าหากว่ารหัสผ่านที่ถูกส่งเข้ามาเปรียบเทียบแล้วไม่ตรงกับที่กำหนดไว้ก็จะทำการปฏิเสธการเชื่อมนั้นเสีย โดยมีการทำงานดังรูป



รูปที่ 6-3 แสดงขั้นตอนการตรวจสอบรหัสผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขั้นตอนเริ่มแรกเมื่อผู้ดูแลระบบต้องการทำการติดต่อ จะต้องทำการใส่รหัสผ่านเสียก่อนโดยรหัสผ่านนั้นจะถูกเข้ารหัสก่อนที่จะถูกส่งออกไปยังเครื่องปลายทางโดยใช้การเข้ารหัสแบบ เดส(DES) ถ้าหากมีผู้ประสงค์ร้ายต้องการจะดักข้อมูลที่เป็นรหัสผ่านในระหว่างที่รหัสกำลังถูกส่งไปปลายทางก็จะทำให้ไม่สามารถรู้ว่ารหัสผ่านคืออะไร เมื่อรหัสผ่านถึงปลายทาง เครื่องปลายทางซึ่งมีตัวถอดรหัสอยู่ก็จะทำการถอดรหัสแล้วนำรหัสผ่านนั้นมาเปรียบเทียบกับค่าที่ได้ตั้งไว้ หากค่าที่ได้รับถูกต้องก็จะอนุญาตให้ทำการเชื่อมต่อได้หากไม่ก็จะทำการปฏิเสธการเชื่อมต่อนั้นเสีย

6.3การตรวจสอบสถานะของเครื่องลูกข่าย

เนื่องจากว่าเมื่อมีการใช้งานโปรแกรมนี้เราต้องทำการตรวจสอบสถานะของเครื่องลูกข่ายในขณะนั้นว่าเครื่องลูกข่ายมีสถานะใดเนื่องจากว่าเครื่องลูกข่ายสามารถที่จะออนไลน์หรือออฟไลน์เมื่อไหร่ก็ได้ดังนั้นจึงต้องทำการตรวจสอบสถานะของเครื่องลูกข่ายทุกเครื่องโดยการส่งสัญญาณออกไปทดสอบสถานะของเครื่องลูกข่ายตามเวลาที่เรากำหนดไว้โดยตั้งไว้ที่ 1 นาที เมื่อเครื่องลูกข่ายออนไลน์ก็จะทำการตอบกลับมาทำให้สามารถรู้ว่าเครื่องลูกข่ายเครื่องไดมีสถานะอย่างไร

6.4โปรแกรมควบคุมเครื่องลูกข่ายผ่านกราฟิกโหมด

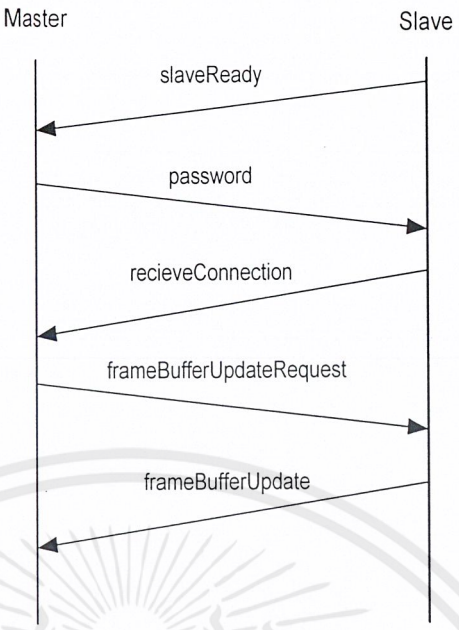
6.4.1หลักการ

ตัวโปรแกรมส่วนนี้ถือได้ว่ามีความสำคัญมาก เนื่องจากว่าส่วนนี้จะอนุญาตให้เราสามารถทำการควบคุมเครื่องลูกข่ายได้อย่างง่ายดายเสมือนทำงานอยู่หน้าจอของเครื่องของเราเอง ทำให้เราสามารถทำการติดตามพฤติกรรมการใช้งานของผู้ใช้งานได้เพื่อจะหามาตรการที่จะควบคุมการใช้งานของผู้ใช้งานในขั้นต่อไป หรือผู้ดูแลอาจเข้าทำการควบคุมเครื่องลูกข่ายได้ในกรณีที่เกิดปัญหาเพื่อทำการแก้ไขปัญหานั้นจากระยะไกล

6.4.2การทำงาน

ทางเครื่องฝั่งควบคุมจะต้องทำการเปิดการเชื่อมต่อรออยู่ก่อนเพื่อที่จะให้ฝั่งควบคุมสามารถที่จะทำการเชื่อมต่อได้ ถ้าเครื่องลูกข่ายเครื่องนั้นมีรายชื่ออยู่ในรายการของเครื่องฝั่งควบคุมแล้วเครื่องลูกข่ายเครื่องนั้นมีการใช้งาน(เปิดเครื่องใช้งาน)ขึ้นมา มันก็จะส่งสัญญาณมาบอกเครื่องฝั่งควบคุมเพื่อเปลี่ยนสถานะจากสถานะออฟไลน์มาเป็นสถานะออนไลน์ หากเครื่องลูกข่ายไม่มีรายชื่ออยู่ในรายการแต่มีการร้องขอการเชื่อมต่อก็จะแสดงข้อความแจ้งมายังผู้ดูแลที่ทำการควบคุมเครื่องแม่ข่ายอยู่ หากผู้ดูแลต้องการให้เครื่องอยู่ในรายการของเครื่องลูกข่ายก็จะสามารถทำการเพิ่มรายชื่อลงไปยังรายการได้ โปรแกรมจะมีลำดับการทำงานเริ่มต้นด้วยการร้องขอให้เครื่องลูกข่ายส่งเฟรมของลำดับภาพที่เกิดขึ้นมาให้เมื่อลำดับภาพถูกส่งมาให้เครื่องแม่ข่ายแล้วทางเครื่องแม่ข่ายนั้นจะต้องทำการแมพภาพที่ได้มาจากเครื่องลูกข่ายมาคำนวณเพื่อหาอัตราส่วนที่เหมาะสมสำหรับการแสดงผลที่หน้าจอของเครื่องแม่ข่ายเองเนื่องจากอาจจะมีการกำหนดค่าของรายละเอียดการแสดงผลไม่เท่ากันจึงต้องทำการแมพเพื่อแสดงผลก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-4 แสดงการทำงานของการควบคุมเครื่องลูกข่ายกราฟิกโหมด

ทางฝั่งเครื่องลูกข่ายหลังจากมีการยอมรับการเชื่อมต่อแล้ว มันก็จะทำการส่งลำดับภาพของเฟรมบัฟเฟอร์ไปให้เครื่องแม่ข่ายเพื่อแสดงผลตามลำดับเวลาที่ได้กำหนดไว้

ในส่วนของการเข้าควบคุมหน้าจอของเครื่องลูกข่ายนั้นนั้นเราจะต้องทำการควบคุมเมสเซจของวินโดวส์ให้มันทำการรับเมสเซจอีเวนต์จากเครื่องแม่ข่ายหากมีการส่งอีเวนต์ใดๆมาจากเครื่องแม่ข่ายก็ให้เครื่องลูกข่ายทำงานตามอีเวนต์ที่เครื่องแม่ข่ายส่งมาก่อน โดยอีเวนต์ที่ส่งออกไปจากเครื่องแม่ข่ายออกไปนั้นจะแบ่งได้เป็นสองแหล่งคือ จากคีย์บอร์ดหรือเม้าส์

6.5โปรแกรมควบคุมเครื่องลูกข่ายผ่านเท็กซ์โหมด

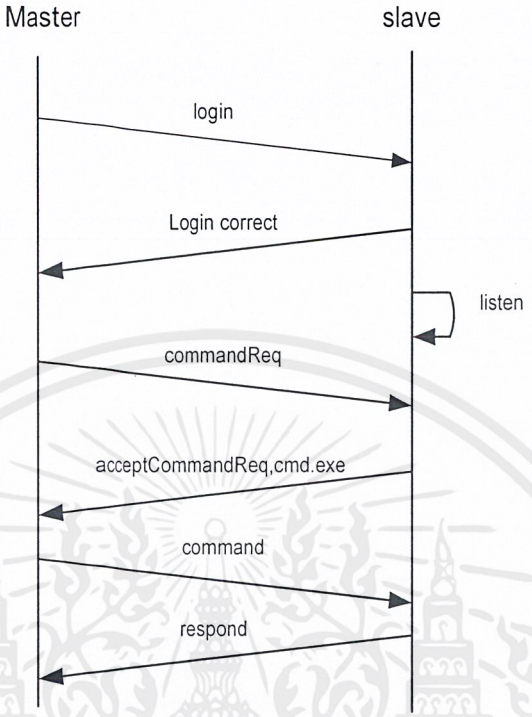
6.5.1หลักการ

การควบคุมทางเท็กซ์โหมดจะมีข้อดีกว่าการควบคุมผ่านทางกราฟิกโหมดคือ จะใช้ทรัพยากรในการทำงานของโปรแกรมน้อยกว่ามากเนื่องจากส่วนใหญ่ในกราฟิกโหมดจะอยู่ในลักษณะของไฟล์ภาพซึ่งมีขนาดใหญ่กินทรัพยากรมาก แต่ในเท็กซ์โหมดจะมีแต่เท็กซ์และข้อมูลที่จำเป็นเท่านั้นที่วิ่งอยู่จึงทำมันกินทรัพยากรน้อยกว่าและกินแบนด์วิดธ์ไม่มาก มันจะช่วยให้ผู้ใช้งานมีความสะดวกในการควบคุมเครื่องลูกข่ายถ้าหากในเครือข่ายนั้นมีทราฟฟิกวิ่งอยู่หนาแน่น

6.5.2การทำงาน

โปรแกรมในส่วนนี้จะทำการส่งคอมมานด์พรอมต์ของเครื่องเป้าหมายที่เราต้องการมาให้เมื่อเราต้องการเข้าควบคุมเครื่องลูกข่ายเครื่องนั้นๆ โดยตัวโปรแกรมที่ถูกติดตั้งไว้ในเครื่องลูกข่ายปลายทางจะถูกกำหนดไว้ให้ทำการรอคอยการเชื่อมต่อไว้ที่พอร์ตหนึ่งเมื่อมีการเชื่อมเข้ามาตัวโปรแกรมจะทำการรันคอมมานด์เชลล์ขึ้นมาโดยเชลล์นี้สามารถส่งกลับคำสั่งไปยังเครื่องลูกข่ายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-5 แสดงการทำงานของการควบคุมเครื่องลูกข่ายผ่านเท็กซ์โหมด

6.6 โปรแกรมสนทนา

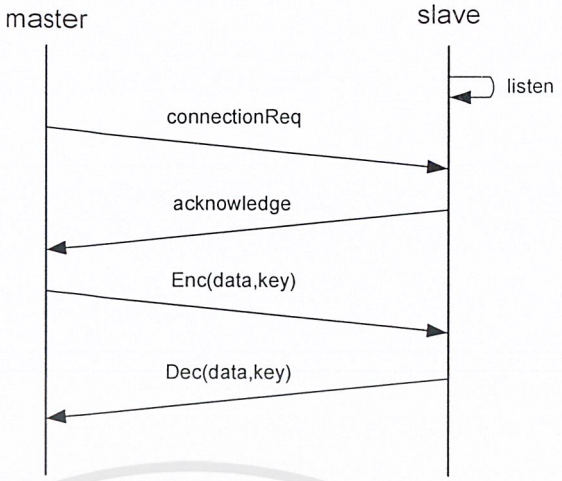
6.6.1หลักการ

โปรแกรมสนทนาที่ทำขึ้นใช้งานในโปรแกรมนี้เป็นโปรแกรมสนทนาแบบบุคคลต่อบุคคลคือ ระหว่างผู้ใช้เครื่องแม่ข่ายกับเครื่องของลูกข่ายเท่านั้น เพื่อที่จะใช้ในการแก้ไขปัญหาเร่งด่วนหรือสอบถามกันระหว่างเครื่อง เนื่องจากบางปัญหาสามารถทำการแก้ไขได้โดยง่ายเพียงให้คำแนะนำเท่านั้น จึงเป็นการสะดวกแก่ผู้ดูแลในการแก้ไขปัญหาที่เราสร้างโปรแกรมสนทนาขึ้นมาเองเพราะว่าในบางกรณีที่ระบบเน็ตเวิร์กไม่สามารถออกสู่อินเตอร์เน็ตได้ โปรแกรมสนทนาทั่วไปก็ไม่สามารถทำงานได้ แต่ถ้าหากระบบเน็ตเวิร์กภายในยังใช้งานได้โปรแกรมตัวนี้ยังคงสามารถใช้งานได้เหมือนเดิม

6.6.2การทำงาน

เมื่อเครื่องลูกข่ายเปิดใช้งานเครื่อง(บูตเครื่องเพื่อใช้งาน) ตัวโปรแกรมสนทนาทางฝั่งของเครื่องลูกข่ายก็จะเริ่มทำงานโดยการรอคอยการติดต่ออยู่แล้ว เมื่อเครื่องแม่ข่ายทำการเชื่อมต่อเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-6 แสดงการทำงานของโปรแกรมสนทนา

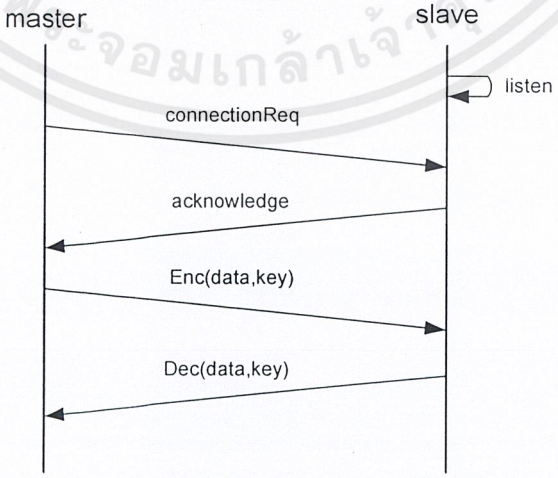
6.7 โปรแกรมโอนย้ายข้อมูล

6.7.1หลักการ

โปรแกรมส่วนนี้จะใช้ในการโอนย้ายข้อมูลระหว่างเครื่องภายใต้การดูแลของผู้ใช้งานเครื่องแม่ข่ายเพื่อความสะดวกของผู้ใช้งานในการย้ายข้อมูลไปยังเครื่องต่างๆ เพื่อความรวดเร็วและสะดวกในการดูแลระบบทำให้การจัดการระบบง่ายยิ่งขึ้น

6.7.2การทำงาน

การทำงานของส่วนนี้ก็คล้ายๆกับการทำงานของโปรแกรมสนทนาข้างต้นคือ จะทำการรอคอยการเชื่อมต่อเข้ามาอยู่แล้วเมื่อมีการยอมรับการเชื่อมต่อเรียบร้อยแล้วก็จะสามารถรับส่งไฟล์ได้เช่นเดียวกับการส่งข้อความหรือข้อมูลประเภทต่างๆ

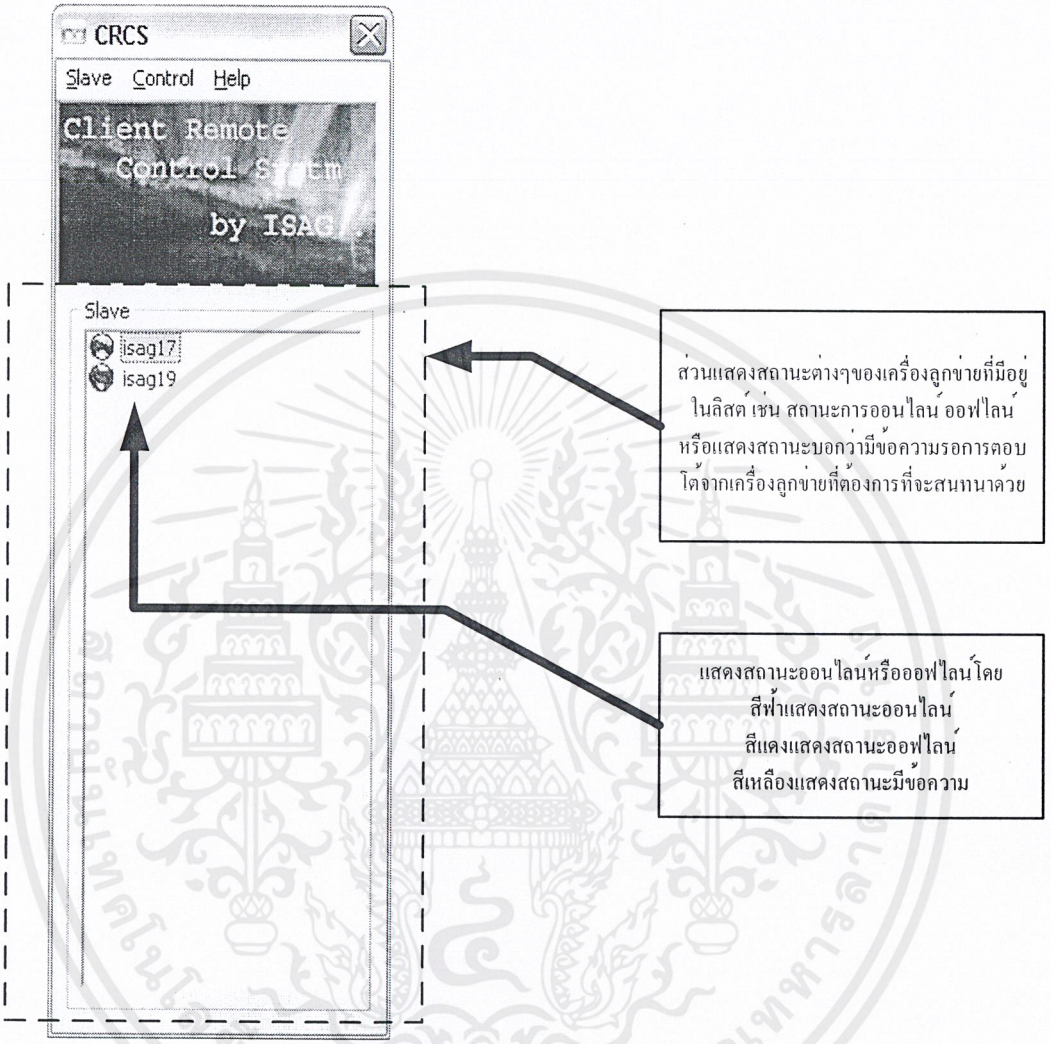


รูปที่ 6-7 แสดงการทำงานของการถ่ายโอนข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.8 การออกแบบส่วนติดต่อผู้ใช้งาน

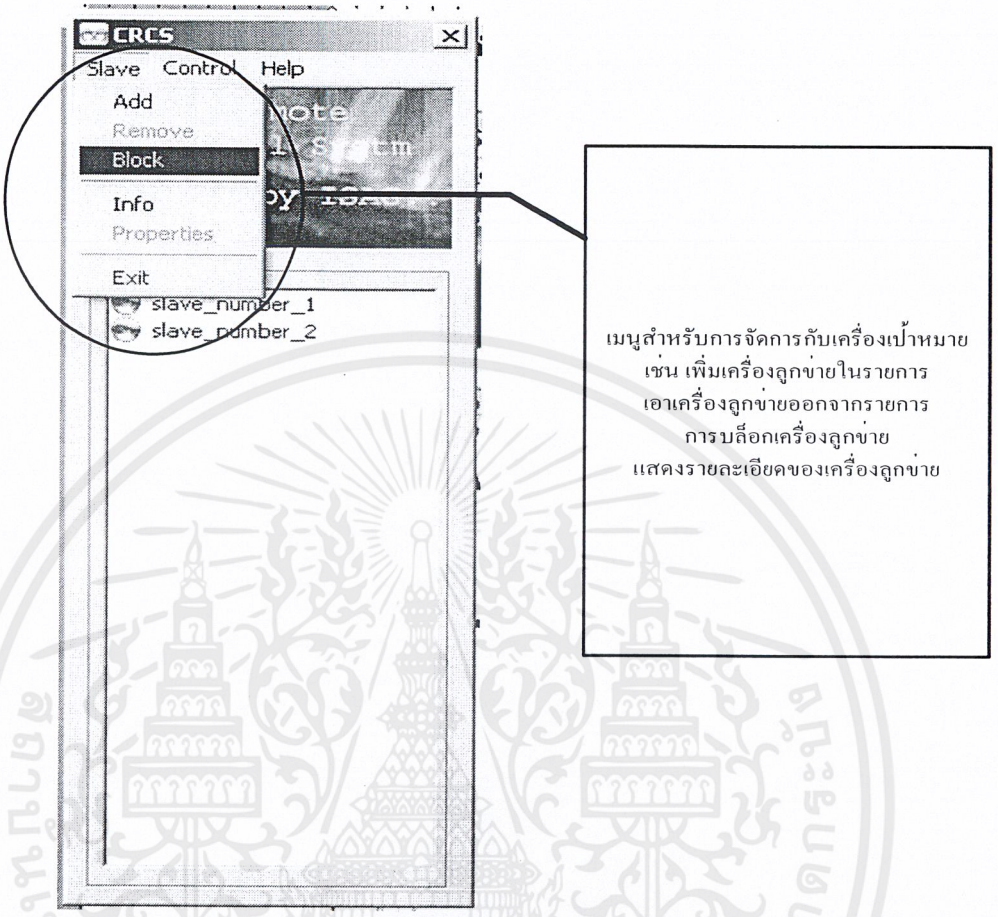
6.8.1 ส่วนติดต่อกับผู้ใช้งานฝั่งเครื่องแม่ข่าย



รูปที่ 6-8 แสดงหน้าจอของส่วนแสดงสถานะ

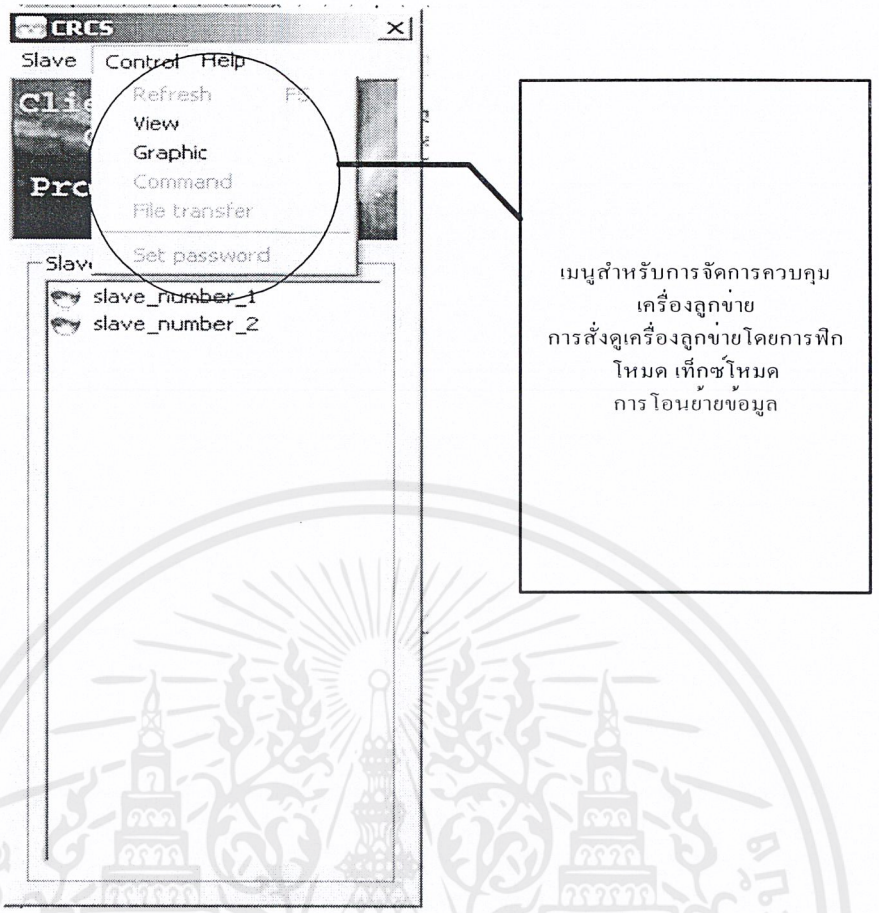
ส่วนนี้จะเป็นหน้าต่างหลักของการควบคุมเครื่องลูกข่ายทั้งหมด โดยหน้าต่างนี้จะทำการแสดงสถานะของเครื่องลูกข่ายที่มีรายการอยู่และยังสามารถเข้าทำการควบคุมเครื่องลูกข่ายที่ต้องการผ่านหน้าต่างนี้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



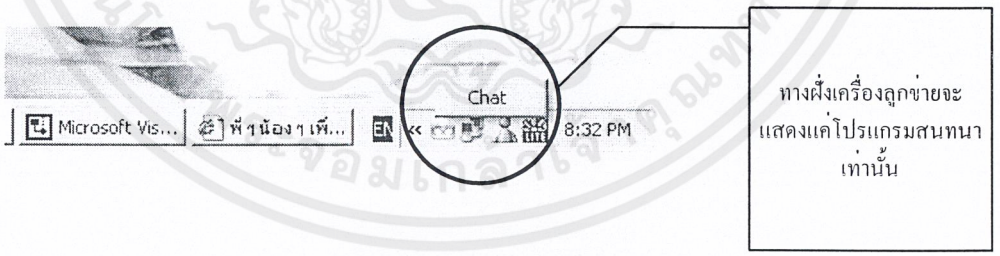
รูปที่ 6-9 แสดงเมนูการจัดการเครื่องลูกข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-10 แสดงเมนูการควบคุมเครื่องลูกข่าย

6.8.2ส่วนติดต่อกับผู้ใช้งานในฝั่งเครื่องลูกข่าย



รูปที่ 6-11 แสดงส่วนติดต่อผู้ใช้ฝั่งเครื่องลูกข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

ผลการทดสอบ

ในการพัฒนาซอฟต์แวร์นั้นมีหลายขั้นตอนซึ่งการทดสอบความถูกต้องของซอฟต์แวร์ก็เป็นหนึ่งในหลายขั้นตอนของการพัฒนาซอฟต์แวร์ ซึ่งนับได้ว่ามีความสำคัญมากอย่างหนึ่งเพราะจะทำให้เรามั่นใจได้ว่าซอฟต์แวร์ที่เราทำการพัฒนาขึ้นมานั้นสามารถนำไปใช้งานได้อย่างถูกต้องและมีปัญหาหรือไม่ อีกทั้งขั้นตอนการทดสอบความถูกต้องของซอฟต์แวร์นี้จะทำให้ เราค้นพบข้อผิดพลาดที่ยังไม่ปรากฏออกมาในซอฟแวร์ของเรา ซึ่งทำให้เราแก้ไขได้ก่อนที่จะมีการนำไปใช้งานได้จริง

7.1 การทดสอบประสิทธิภาพของระบบ

ขั้นตอนการทดสอบการทำงานของโปรแกรมควบคุมเครื่องลูกข่ายระยะไกล มีรายระเอียดของเครื่องที่นำมาทดสอบดังต่อไปนี้

เครื่องที่ติดตั้งโปรแกรมควบคุม

- หน่วยประมวลผล 650 MHz
- หน่วยความจำหลัก 256 MB
- ระบบปฏิบัติการวินโดวส์ เอ็กซ์พี เซอร์วิสแพ็ก 1
- การ์ดแลนความเร็ว 100 Mb/s
- ทดสอบผ่านระบบเครือข่ายแลน

เครื่องที่ติดตั้งโปรแกรมฝั่งลูกข่าย

- หน่วยประมวลผล 800 MHz
- หน่วยความจำหลัก 256 MB
- ระบบปฏิบัติการวินโดวส์ เอ็กซ์พี เซอร์วิสแพ็ก 1
- การ์ดแลนความเร็ว 100 Mb/s
- ทดสอบผ่านระบบเครือข่ายแลน

7.2 เริ่มต้นการทดสอบโปรแกรม

- การเริ่มต้นการทำงานของเครื่องฝั่งควบคุม

  ครั้งแรกที่โปรแกรมเริ่มต้นการทำงานครั้งแรก จะมีลักษณะดังนี้

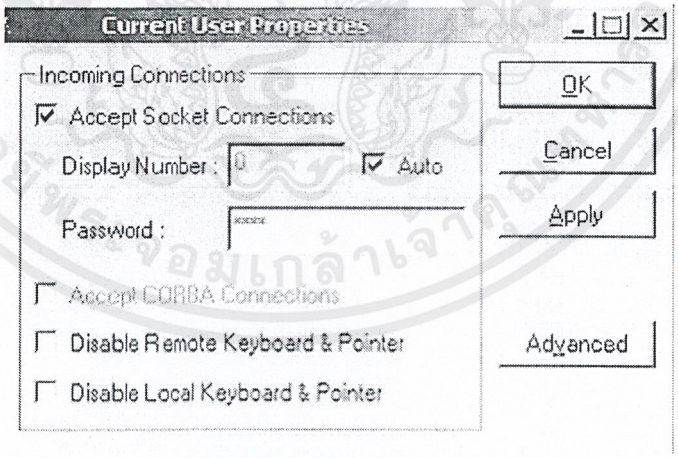
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-1 แสดงสถานะเริ่มต้นและแสดงการเพิ่มเครื่องลูกข่าย

- การเริ่มต้นการทำงานของโปรแกรมฝั่งลูกข่าย

เครื่องทางฝั่งลูกข่ายเมื่อเริ่มต้นการทำงาน มันจะมีลักษณะที่สามารถทำงานในลักษณะที่ผู้ใช้งานจะไม่สังเกตุเห็นอะไรโผล่ขึ้นมาเลย จะมีเพียงไอคอนที่ซิสเต็มเทรย์ที่ใช้สำหรับการสนทนาเท่านั้น แล้วให้ทำการเซ็ตค่าต่างๆตามต้องการพร้อมทั้งกำหนดรหัสผ่านด้วย

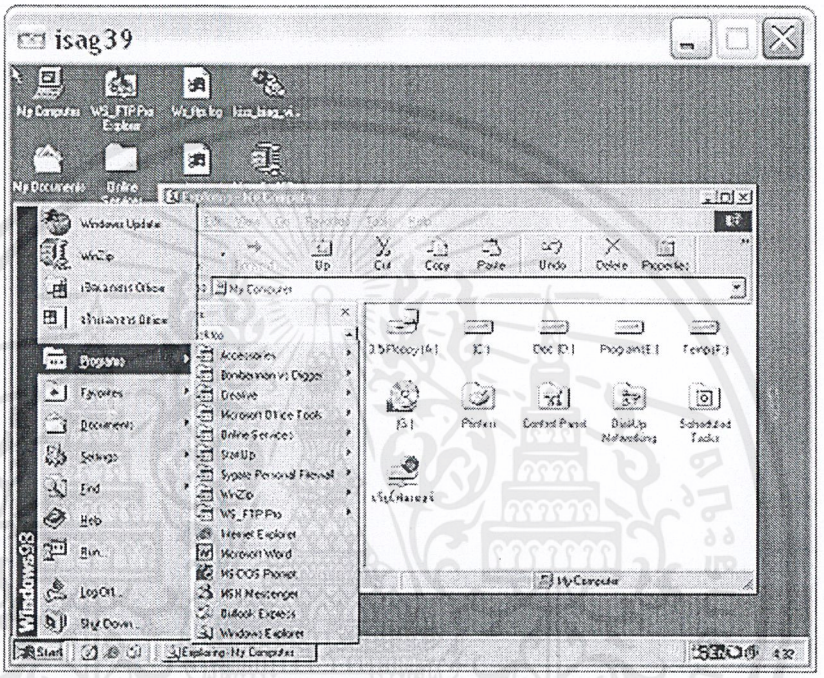


รูปที่ 7-2 การกำหนดรหัสผ่านและเซ็ตค่าต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2.1 ทดสอบการทำงานของโปแกรมควบคุมผ่านกราฟิก

โดยการกำหนดให้แสดงหน้าจอของเครื่องลูกข่ายทีละหลายๆเครื่อง พบว่าหากเราทำการส่งภาพพื้นหลังออกไปด้วยมันจะทำให้ข้อมูลที่เราทำการส่งออกไปนั้นมีขนาดใหญ่มาก ทำให้การแสดงผลของฝั่งมาสเตอร์ทำงานได้ช้า เป็นปัญหาอย่างยิ่งในการส่งข้อมูลครั้งละหลายๆเครื่องพร้อมๆกัน ดังนั้นเพื่อการที่จะสามารถแสดงผลให้ได้อย่างรวดเร็วเราจึงทำการตัดภาพพื้นหลังออกจากข้อมูลที่ส่งออกไป ทำให้การแสดงผลมีความรวดเร็วมากขึ้น

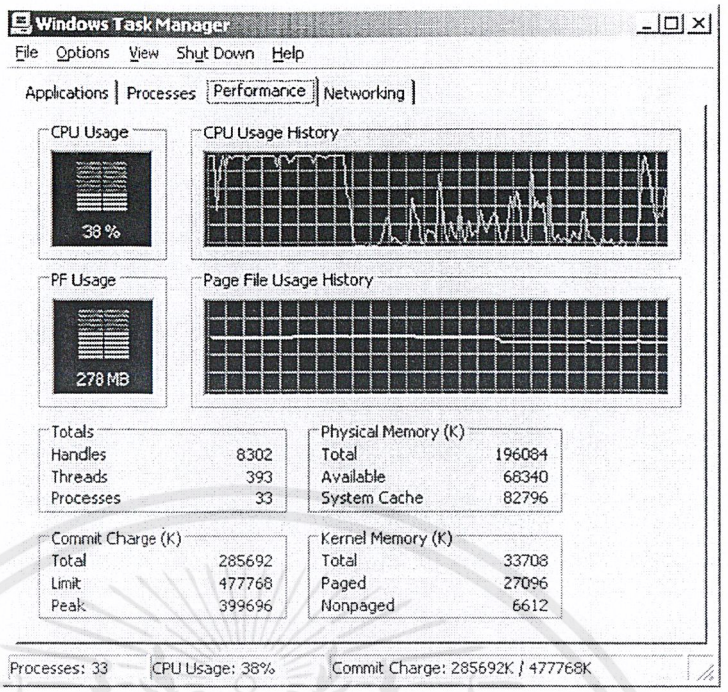


รูปที่ 7-3 แสดงการทำการควบคุม

ในการทดสอบการทำงานของส่วนนี้จะทำการป้อนไอพีแอดเดรสและรหัสผ่าน แล้วจึงทำการจับขนาดของข้อมูลที่ส่งไปมาโดยไอพีและรหัสผ่านที่ถูกส่งออกไปแล้วนั้นได้ทำการเข้ารหัสไว้แล้ว ฉะนั้นหากมีการดักจับข้อมูลส่วนนี้ไปได้แต่ก็ไม่สามารถนำไปใช้งานได้เนื่องจากข้อมูลถูกเข้ารหัสด้วยวิธีการ ดีอีเอส ที่มีความน่าเชื่อถือสูงว่าจึงปลอดภัยต่อการที่จะนำมาใช้ในการเข้ารหัส

หลังจากนั้นก็ทำการทดลองควบคุมทีละหลายๆเครื่องทำให้พบว่าการทำงานเริ่มกระตุกจนสังเกตได้ที่มีเครื่องลูกข่ายประมาณ 20 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-4 แสดงการใช้ทรัพยากรของโปรแกรมเมื่อมีการเรียกเครื่องลูกข่าย 20 เครื่อง

7.2.2 ทดสอบการทำงานของโปรแกรมควบคุมผ่านเท็กซ์โหมด

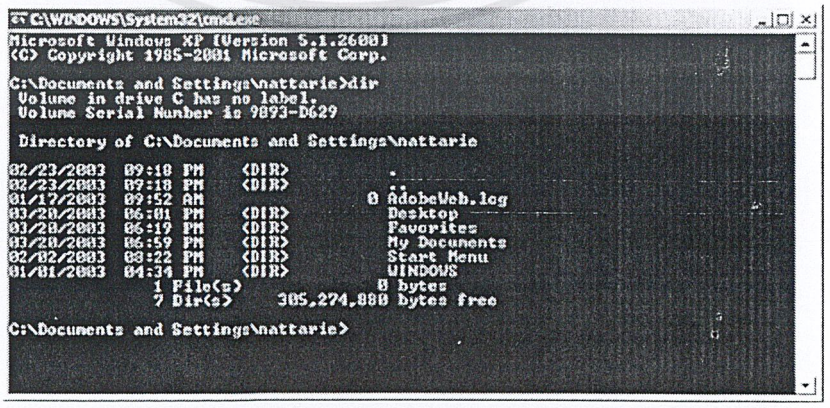
การทดสอบเริ่มต้นโดยการสั่งให้เครื่องเป้าหมายทำการรอคอยการเชื่อมต่อไว้ก่อน โดยกำหนดพอร์ตและโปรแกรมที่จะให้มันทำการส่งกลับมายังเครื่องควบคุมดังนี้

C:\nc –l –p port [option] [host name] [port]

เมื่อจะติดต่อเครื่องเป้าหมายก็ให้เรียกใช้ดังนี้

C:\nc [-option] hostname port [port]

เครื่องปลายทางก็จะได้เชลล์คอมมานด์ไลน์ของเครื่องเป้าหมายมาดังนี้

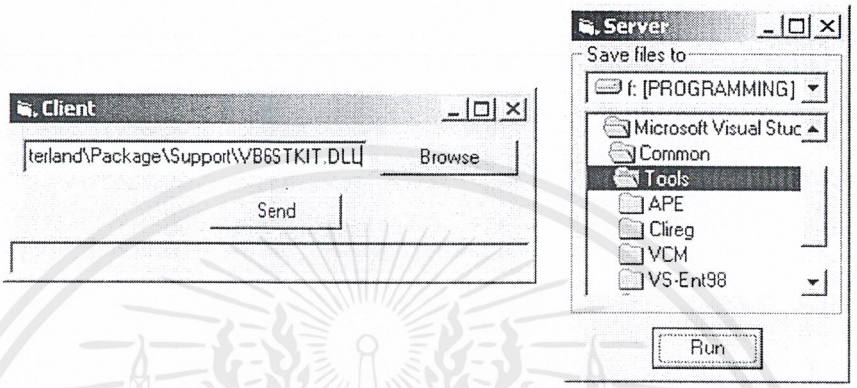


รูปที่ 7-5 แสดงคอมมานด์เชลล์ที่หลังจากเชื่อมต่อแล้วลิสต์ที่แสดงเป็นรายการของเครื่องปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2.3 ทดสอบการทำงานของโปแกรมโอนย้ายข้อมูล

การทำงานของโปแกรมโมดูลนี้ จะทำงานเหมือนกับโปรแกรมโอนย้ายข้อมูลทั่วๆไป แต่ข้อมูลที่ไหลผ่านทั้งสองฝั่งได้ถูกเข้ารหัสไว้แล้วทำให้ปลอดภัยต่อการดักจับข้อมูล ในการโอนย้ายข้อมูลสามารถทำได้ครั้งละหนึ่งเครื่องเท่านั้น



รูปที่ 7-6 แสดงหน้าต่างสำหรับการการโอนย้ายไฟล์

7.2.4 ทดสอบการทำงานของโปรแกรมสนทนา

ในส่วนของโปรแกรมส่วนนี้จะทำได้ถูกออกแบบมาเพื่อการใช้สำหรับการสนทนาคุยกันระหว่างผู้ใช้งานกับผู้ดูแลระบบเท่านั้นไม่อนุญาตให้มีการสนทนากันระหว่างผู้ใช้กับผู้ใช้ด้วยกันเอง โปรแกรมทำงานโดยข้อมูลที่ส่งผ่านไปมาจะเข้ารหัสด้วย เมื่อมีการดักจับข้อมูล ข้อมูลได้ถูกทำการเข้ารหัสไว้ทำให้มีความปลอดภัยพอสมควรในการใช้โปรแกรมส่วนนี้

7.3 ปัญหาที่เกิดขึ้นขณะทดสอบ

ปัญหาเกิดขึ้นเมื่อมีการรับการเชื่อมต่อกับเครื่องลูกข่ายทีละหลายเครื่อง เมื่อมีการแสดงผลภาพภาพที่นำมาแสดงจะกระตุก จึงแก้ปัญหาโดยการตัดเอาภาพพื้นหลังของหน้าจอออกทำให้การแสดงผลได้ดีมากกว่าเดิม แต่ปัญหาเดิมที่เกิดขึ้นอีกเมื่อมีการรับการเชื่อมต่อของเครื่องลูกข่ายเพิ่มขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
บทวิจารณ์และสรุป

8.1 วิจารณ์โครงงาน

เนื่องจากโครงงานชิ้นนี้ต้องการความเร็วในการส่งข้อมูลระหว่างเครื่องและต้องการความเร็วในการรีเฟรชข้อมูลของการปรับปรุงภาพหน้าจอของฝั่งไคลเอนต์สูง หากความเร็วในการส่งข้อมูลไม่เร็วพอจะทำให้ภาพที่เกิดขึ้นเกิดอาการกระตุกได้ และตัวโปรแกรมได้ทำการเข้ารหัสผ่านไว้เท่านั้น ไม่ได้เข้ารหัสข้อมูลภาพที่วิ่งไปมาอยู่บนเน็ตเวิร์กทำให้ผู้ไม่ประสงค์ดีสามารถทำการดักจับข้อมูลเหล่านั้นได้ อาจเป็นอันตรายต่อเครื่องที่ถูกควบคุมได้ ถ้าผู้ไม่ประสงค์ดีนั้นทำการปลอมแปลงเป็นผู้ดูแลระบบ หากแต่เครื่องมือชิ้นนี้เป็นเครื่องมือที่ใช้ในระบบปิด คือออกแบบให้ผู้ดูแลระบบนำมาใช้กับกลุ่มคอมพิวเตอร์ที่เป็นเครื่องสาธารณะอยู่แล้ว เช่น เครื่องคอมพิวเตอร์ในห้องปฏิบัติการ หรือห้องคอมพิวเตอร์ที่เปิดให้พนักงานเข้ามาหาข้อมูล ซึ่งมีการควบคุมการใช้งานเครื่องคอมพิวเตอร์ที่ดีอยู่แล้ว หากจะมีการเข้าควบคุมเครื่องเครือข่ายจากภายนอกนั้นก็ต้องผ่านระบบรักษาความปลอดภัยขององค์กรนั้นๆเสียก่อน

ปัญหาทั่วไปที่เป็นอุปสรรคต่อการพัฒนาโปรแกรม

- ในการหาข้อมูลในแต่ละเรื่องที่จะเอามาใช้นั้นบางเรื่องต้องเสียเวลาในการค้นหาข้อมูลเป็นเวลานานทำให้เสียเวลาเป็นจำนวนมาก
- ปัญหาจากความสามารถของตัวภาษาที่นำมาใช้ จึงทำให้ต้องค้นหาเทคนิคต่างๆมาแก้ปัญหาเหล่านั้น

เครื่องมือชุดนี้ทำงานอยู่บนแพลตฟอร์มวินโดวส์ไม่สามารถทำงานบนแพลตฟอร์มอื่นๆที่ไม่ใช่ระบบปฏิบัติการของไมโครซอฟท์ ในระบบคอมพิวเตอร์ที่มีเครื่องคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการอื่นนั้นไม่สามารถเข้ามาดูแลผ่านโปรแกรมชุดนี้ได้

8.2 สรุปผลโครงงาน

โครงงานชิ้นนี้ประกอบด้วยฟังก์ชันการทำงานที่หลากหลายทำให้เสียเวลาในการค้นหาข้อมูลและทำความเข้าใจเป็นเวลานาน และผลงานที่ออกมาก็สามารถทำงานได้ตามที่กำหนดไว้จึงสามารถนำไปใช้งานได้จริง และมีประสิทธิภาพอยู่ในเกณฑ์ดี จากโครงงานชิ้นนี้ทำให้ผู้จัดทำได้ความรู้ในการเขียนโปรแกรมผ่านเน็ตเวิร์กมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Jesse Liberty: "*Teach Yourself More C++ in 21 day"s* ,SAM Publishing ,1996

[2] Charles Ptzold *: " Programming Windows"* , Microsoft Press, 1990

[3] Douglas E. Comer , David L. Stevens, "Internetworking with TCP/IP", Second Edition, Prentice Hall , 1996

[4] Andrew S. Tanenbaum : "Computer Network" ,Original U.S. Edition, Prentice Hall ,1999

[5] สุวัฒน์ ปุณณชัยยะ, ต้น ตันฑ์สุทธิพงศ์, สุพจน์ ปุณณชัยยะ *: " เปิดโลก TCP/IP และโพรโตคอลของอินเตอร์เน็ต"* , Second Edition , Provision Co.Ltd., 2545

[7] ธาริน สิทธิธรรม : *"Visual Basic version 6.0 ฉบับเพื่อการใช้งานจริง"*, พิมพ์ครั้งที่ 3 , Success Media Co.Ltd.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้