ระบบเฝ้าสังเกตจอภาพของผู้เรียนผ่านเครือข่ายในห้องเรียนคอมพิวเตอร์

นายกิติพงษ์ ตันโสภณธนาศักดิ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2542

ISBN 974-334-260-5

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

I 19189073

STUDENT'S SCREEN MONITORING SYSTEM VIA NETWORK IN A COMPUTER CLASSROOM

MR. KITIPONG TANSOPONTANASAK

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 1999

ISBN 974-334-260-5

จุฬาลงกรณ์มหาวิทยาลัย

หัวข้อวิทยานิพนธ์	ระบบเฝ้าสังเกตจอภาพของผู้เรียนผ่านเครือข่ายในห้องเรียนคอมพิวเตอร์
โดย	นายกิติพงษ์ ตันโสภณธนาศักดิ์
ภาควิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.สาธิต วงศ์ประทีป

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สมศักดิ์ ปัญญาแก้ว)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ชูชีพ ฉิมวงษ์)

..... อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ ดร.สาธิต วงศ์ประทีป)

..... กรรมการ
(อาจารย์ ชัยศิริ ปัณฑิตานนท์)

..... กรรมการ
(อาจารย์ ดร. อาทิตย์ ทองทักษ์)

นายกิติพงษ์ ตันโสภณธนาศักดิ์ : ระบบเฝ้าสังเกตจอภาพของผู้เรียนผ่านเครือข่ายในห้องเรียนคอมพิวเตอร์. (STUDENT'S SCREEN MONITORING SYSTEM VIA NETWORK IN A COMPUTER CLASSROOM) อ. ที่ปรึกษา : ผศ.ดร.ดร.สาธิต วงศ์ประทีป, 68 หน้า. ISBN 974-334-260-5.

ระบบเฝ้าสังเกตจอภาพของผู้เรียนผ่านเครือข่ายในห้องเรียนคอมพิวเตอร์ มีวัตถุประสงค์เพื่อออกแบบและพัฒนาระบบเฝ้าสังเกตจอภาพของผู้เรียนผ่านเครือข่ายในห้องเรียนคอมพิวเตอร์ โดยใช้ห้องเรียนคอมพิวเตอร์ของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัยเป็นกรณีศึกษา เพื่อเป็นโปรแกรมช่วยการสอนในห้องเรียนคอมพิวเตอร์ โดยไม่มีการต่ออุปกรณ์พิเศษเพิ่มเติมจากระบบเครือข่ายปกติ

ในการพัฒนาโปรแกรมแบ่งส่วนการทำงานออกเป็น 2 ส่วนหลักคือส่วนโปรแกรมที่ทำหน้าที่เป็นไคล์เอ็นต์ หรือ ส่วนโปรแกรมที่ทำหน้าที่เฝ้าสังเกตจอภาพของผู้เรียน ทำหน้าที่นำข้อมูลจอภาพมาแสดงผล และส่วนโปรแกรมที่ทำหน้าที่เป็นเซิร์ฟเวอร์หรือส่วนโปรแกรมที่ทำหน้าที่ส่งจอภาพของตนเองให้อีกฝ่ายที่เฝ้าสังเกต โดยการส่งผ่านข้อมูลใช้แอพพลิเคชันโปรโตคอลชื่ออาร์เอฟบี ที่ออกแบบขึ้นมาบนมาตรฐานของโปรโตคอลสื่อสาร ทีซีพี/ไอพี ผลการทดสอบโปรแกรมสามารถใช้งานได้กับห้องเรียนคอมพิวเตอร์ที่มีจำนวนเครื่องคอมพิวเตอร์อยู่ประมาณ 27 เครื่อง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่ออาจารย์ที่ปรึกษา .....
ปีการศึกษา 2542 ลายมือชื่ออาจารย์ที่ปรึกษาร่วม .....

4070214621 : MAJOR COMPUTER SCIENCE

KEY WORD: SCREEN MONITORING / NETWORK / AGENT

KITIPONG TANSOPONTANASAK : STUDENT'S SCREEN MONITORING SYSTEM VIA NETWORK IN A COMPUTER CLASSROOM. THESIS ADVISOR : ASST. PROF. SARTID VONGPRADHIP,Ph.D., 68 pp. ISBN 974-334-260-5.

The objective of student's screen monitoring system via network in a computer classroom is to design and develop student's screen monitoring system via network in a computer classroom by using computer classroom of Computer Engineer Department, Faculty of Engineer, Chulalongkorn University as a case study.It is designed to help teachers in computer classrooms by a special hardware is not required to plug in the network.

Development of the program can be divided by function into 2 sections.First section is a part of functionality as client that monitor student's computer screen and display screens.The later section is the part of functionality as server that send their screen to monitoring computers.A standard network protocol, TCP/IP, is used to send data with an application protocol, RFB.From the result, this program can be used in a computer's classroom with 27 microcomputers.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่ออาจารย์ที่ปรึกษา .....

ปีการศึกษา 2542 ลายมือชื่ออาจารย์ที่ปรึกษาร่วม .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลืออย่างดียิ่งของ ผศ.ดร.สาธิต วงศ์-ประทีป อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่างๆ ในการวิจัยด้วยดีตลอดมาและตรวจแก้วิทยานิพนธ์ฉบับนี้

ขอขอบคุณห้องปฏิบัติการ Digital System Engineering Laboratory ที่เอื้อเฟื้ออุปกรณ์ในการทำงานวิจัย รวมทั้งพี่ ๆ และเพื่อน ๆ ที่ได้ให้คำปรึกษา และ ความช่วยเหลือในด้านต่าง ๆ ซึ่งทำให้การทำงานวิจัยเป็นไปอย่างราบรื่น

ท้ายนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณบิดา มารดา ซึ่งสนับสนุนในทุกสิ่งทุกอย่างและให้กำลังใจแก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญภาพ ..... ฌ
สารบัญตาราง ..... ฎ

บทที่

1. บทนำ 1
   1.1. ความเป็นมาและความสำคัญของปัญหา 1
   1.2. วัตถุประสงค์ 2
   1.3. ขอบเขตการวิจัย 2
   1.4. ขั้นตอนและวิธีการ 3
   1.5. ประโยชน์ที่คาดว่าจะได้รับ 3
2. แนวความคิดและทฤษฎีที่เกี่ยวข้อง 5
   2.1. ไคล์เอ็นต์/เซิร์ฟเวอร์ (Client/Server) 5
   2.2. ชุดโปรโตคอล ทีซีพี/ไอพี (TCP/IP) 6
   2.3. การบีบอัดรูปภาพที่เป็นแบบเรียลทาม (Real time) 8
   2.4. เทคนิคการถ่ายโอนข้อมูล (Data Transfer Techniques) 11
3. การออกแบบโปรแกรม 15
   3.1. ลักษณะข้อมูลของจอภาพและโปรโตคอลที่ใช้ในการรับส่ง 15
   3.2. การออกแบบโปรแกรมส่วนที่ทำหน้าที่เป็นเซิร์ฟเวอร์ 16
   3.3. การออกแบบโปรแกรมส่วนที่ทำหน้าที่เป็นไคล์เอ็นต์ 18
   3.4. ขั้นตอนวิธีในการรับส่งข้อมูลหน้าจอระหว่างเซิร์ฟเวอร์กับไคล์เอ็นต์ 20
4. การพัฒนาโปรแกรม 22
   4.1. คลาสพื้นฐานของไมโครซอฟท์ที่นำมาใช้ 22
   4.2. คลาสที่ถ่ายทอดคุณสมบัติมาจากคลาสพื้นฐานของไมโครซอฟท์ และคลาสที่สร้างขึ้นเอง 23
5. ผลการทดสอบโปรแกรม 28

สารบัญ (ต่อ)

หน้า

5.1. การทดสอบโปรแกรมส่วนที่เฝ้าสังเกตจอภาพของผู้เรียน 30
5.2. การทดสอบโปรแกรมส่วนที่แสดงจอภาพของผู้สอนให้ผู้เรียน 31
6. สรุปและข้อเสนอแนะ 34
6.1. สรุป 34
6.2. ข้อเสนอแนะ 35
รายการอ้างอิง 36
ภาคผนวก 37
ภาคผนวก ก โปรโตคอล อาร์ เอฟ บี (RFB Protocol) 38
ภาคผนวก ข คู่มือการใช้โปรแกรม 60
ประวัติผู้เขียน 68

จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

หน้า

รูปที่ 2.1 รูปแบบการเข้าถึงข้อมูลระยะไกล (Remote Data Access Model) 5
รูปที่ 2.2 รูปแบบการบริการฐานข้อมูล (Database Server Model)..... 5
รูปที่ 2.3 รูปแบบการบริการแอพพลิเคชั่นชัน (Application Server Model) 6
รูปที่ 2.4 แสดงสถาปัตยกรรมระดับชั้นของโปรโตคอลทีซีพี/ไอพี..... 6
รูปที่ 2.5 แสดง Three-way handshake รูปแบบง่ายที่สุด 8
รูปที่ 2.6 แสดงการจำแนกเทคนิคการบีบอัด (a) lossless (b) lossy..... 10
รูปที่ 2.7 แสดงตัวอย่าง Conditional transfer..... 12
รูปที่ 3.1 แสดงผังงานของการออกแบบโปรแกรมส่วนที่ทำหน้าที่เป็นเซิร์ฟเวอร์ 16
รูปที่ 3.2 แสดงผังงานของการออกแบบโปรแกรมส่วนที่ทำหน้าที่เป็นไคล์เอ็นต์..... 18
รูปที่ 4.1 แสดงผังของคลาสที่ฝั่งเซิร์ฟเวอร์ 26
รูปที่ 4.2 แสดงผังของคลาสที่ฝั่งไคล์เอ็นต์ 27
รูปที่ 5.1 แสดงการทดสอบโปรแกรมส่วนที่เฝ้าสังเกตจอภาพผู้เรียน 32
รูปที่ 5.2 แสดงกราฟการทดสอบโปรแกรมส่วนที่แสดงจอภาพของเครื่องผู้สอน
ให้เครื่องผู้เรียน..... 33
รูปที่ 5.3 แสดงการทำงานของโปรแกรมขณะเฝ้าสังเกตจอภาพผู้เรียน..... 32
รูปที่ ก.1 แสดงการทำงานของโปรโตคอลโดยภาพกว้าง..... 38
รูปที่ ข.1 แสดงวินโดว์หลักของโปรแกรม..... 61
รูปที่ ข.2 แสดงการเพิ่มจอภาพที่ต้องการสังเกตโดยทราบชื่อเครื่องคอมพิวเตอร์ที่อยู่
ในกลุ่มงาน..... 62
รูปที่ ข.3 แสดงการเพิ่มจอภาพที่ต้องการสังเกตโดยไม่ทราบชื่อเครื่องคอมพิวเตอร์ที่อยู่
ในกลุ่มงาน..... 62
รูปที่ ข.4 แสดงจอภาพที่ต้องการสังเกตหลังจากใส่ชื่อเครื่องคอมพิวเตอร์ที่อยู่ในกลุ่มงานแล้ว.. 63
รูปที่ ข.5 แสดงจอภาพที่ต้องการสังเกตหลังจากใส่ชื่อเครื่องคอมพิวเตอร์ที่อยู่ใน
กลุ่มงานแล้วหลายเครื่องพร้อมกัน..... 63
รูปที่ ข.6 แสดงวิธีอนุญาตการแสดงจอภาพของผู้สอนให้ผู้เรียนสังเกต..... 64
รูปที่ ข.7 แสดงสัญรูปแสดงจอภาพของผู้สอนให้ผู้เรียนสังเกต(สีเขียว)บริเวณทาสก์บาร์ 64
รูปที่ ข.8 แสดงการแสดงจอภาพของผู้สอนให้ผู้เรียนสังเกต..... 64
รูปที่ ข.9 แสดงคำโต้ตอบก่อนแสดงจอภาพของผู้สอนให้ผู้เรียนสังเกต..... 65
รูปที่ ข.10 แสดงการเวียนจอภาพที่ต้องการสังเกตและยกเลิกการเวียน..... 65

สารบัญภาพ(ต่อ)

หน้า

รูปที่ ข.11 แสดงรูปก่อนเข้าตัวเลือกของวินโดว์จอภาพเฟ้าสังเกต 66
รูปที่ ข.12 แสดงตัวเลือกของแต่ละวินโดว์จอภาพเฟ้าสังเกต..... 66
รูปที่ ข.13 แสดงตัวเลือกของโปรแกรมที่ฝ่ายถูกเฟ้าสังเกต 67

จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

หน้า

ตารางที่ 5.1 แสดงผลการทดสอบโปรแกรม ส่วนที่เฝ้าสังเกตจอภาพของผู้เรียน
โดย round trip time 30
ตารางที่ 5.2 แสดงผลการทดสอบโปรแกรมส่วนที่แสดงจอภาพของผู้สอนให้ผู้เรียน 31
ตารางที่ ก.1 แสดงรูปแบบของ ProtocolVersion 43
ตารางที่ ก.2 แสดงรูปแบบ authentication scheme 44
ตารางที่ ก.3 แสดงรูปแบบ connection failed 44
ตารางที่ ก.4 แสดงรูปแบบของ VNC authentication challenge 44
ตารางที่ ก.5 แสดงรูปแบบของ VNC authentication response 45
ตารางที่ ก.6 แสดงรูปแบบของ VNC authentication ส่วนที่เซิร์ฟเวอร์แจ้งการพิสูจน์ตัวจริง 45
ตารางที่ ก.7 แสดงรูปแบบข้อความ Initialisation 45
ตารางที่ ก.8 แสดงรูปแบบข้อความ ServerInitialisation 46
ตารางที่ ก.9 แสดงรูปแบบ PIXEL_FORMAT 46
ตารางที่ ก.10 แสดงรูปแบบข้อความ SetPixelFormat 48
ตารางที่ ก.11 แสดงรูปแบบข้อความ FixColourMapEntries 48
ตารางที่ ก.12 แสดงรูปแบบเขตข้อมูล number-of-colour 49
ตารางที่ ก.13 แสดงรูปแบบข้อความ SetEncodings 49
ตารางที่ ก.14 แสดงรูปแบบเขตข้อมูล number-of-encodings 49
ตารางที่ ก.15 แสดงรูปแบบข้อความ FramebufferUpdateRequest 50
ตารางที่ ก.16 แสดงรูปแบบของ KeyEvent 51
ตารางที่ ก.17 แสดงค่าของ “keysym 51
ตารางที่ ก.18 แสดงรูปแบบ PointerEvent 52
ตารางที่ ก.19 แสดงรูปแบบ ClientCutText 53
ตารางที่ ก.20 แสดงรูปแบบข้อความ FramebufferUpdate 53
ตารางที่ ก.21 แสดงรูปแบบ number-of-rectangles 53
ตารางที่ ก.22 แสดงรูปแบบ Copy Rectangle 54
ตารางที่ ก.23 แสดงรูปแบบ RRE ส่วนหัว 54
ตารางที่ ก.24 แสดงรูปแบบสี่เหลี่ยมย่อยของ RRE 55
ตารางที่ ก.25 แสดงรูปแบบ CoRRE ส่วนหัว 55
ตารางที่ ก.26 แสดงรูปแบบสี่เหลี่ยมย่อยของ CoRRE 55

สารบัญตาราง (ต่อ)

หน้า

ตารางที่ ก.27 แสดงรูปแบบ subencoding type byte 56
ตารางที่ ก.28 แสดงรูปแบบ background-pixel-value 56
ตารางที่ ก.29 แสดงรูปแบบ foreground-pixel-value 57
ตารางที่ ก.30 แสดงรูปแบบ number-of-subrectangles 57
ตารางที่ ก.31 แสดงรูปแบบการกำหนดค่าสีของสี่เหลี่ยมย่อย 57
ตารางที่ ก.32 แสดงรูปแบบการบอกตำแหน่งสี่เหลี่ยมย่อยโดยใช้ค่าสีเหมือนกับสี่เหลี่ยมย่อยที่อยู่ก่อนหน้า 58
ตารางที่ ก.33 แสดงรูปแบบข้อความ SetColourMapEntries 58
ตารางที่ ก.34 แสดงรูปแบบเขตข้อมูล number-of-colours 58
ตารางที่ ก.35 แสดงรูปแบบข้อความ Bell 59
ตารางที่ ก.36 แสดงรูปแบบข้อความ ServerCutText 59

จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1
บทนำ

ระบบเฝ้าสังเกตจอภาพของผู้เรียนผ่านเครือข่ายในห้องเรียนคอมพิวเตอร์

1.1. ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันได้มีการนำคอมพิวเตอร์มาช่วยทำงานกันมากขึ้น เพื่อให้ได้รับความสะดวกรวดเร็วถูกต้อง ในการทำงานซึ่งเดิมใช้แรงงานคนดำเนินการ แต่การควบคุมคอมพิวเตอร์ให้มีการทำงานตามที่ต้องการ ทำให้ต้องมีส่วนที่ทำหน้าที่ติดต่อกันระหว่างคอมพิวเตอร์กับผู้ใช้ หรือในที่นี้คือจอภาพแสดงการทำงานของผู้ใช้บนเครื่องคอมพิวเตอร์นั้น การตรวจสอบการทำงานที่สั่งให้คอมพิวเตอร์ทำงานก็สามารถทำได้โดยการดูสถานะการทำงานบนจอภาพซึ่งก็ตรงไปตรงมา แต่ถ้าผู้ใช้คอมพิวเตอร์อยู่คนละแห่งกันต้องการตรวจสอบสถานะการทำงานก็อาจทำให้ยุ่งยาก ต้องอาศัยเวลาในการตรวจสอบมาก

ในระบบเครือข่ายที่มีการควบคุมจากส่วนกลางไปยังส่วนของเครื่องคอมพิวเตอร์ผู้ใช้งาน โดยมีการติดต่อกันด้วยอุปกรณ์เครือข่ายที่มีอยู่แล้ว จึงเป็นเรื่องที่สอดคล้องกับปัญหาที่ต้องการแก้ไขอยู่ คือ อาศัยการตรวจสอบสถานะการทำงานเครื่องคอมพิวเตอร์ผู้ใช้ผ่านระบบเครือข่าย มายังส่วนควบคุมส่วนกลาง โดยไม่ต้องมีอุปกรณ์ฮาร์ดแวร์ใดๆเพิ่มเติมอีก

ในตัวอย่างกรณีศึกษาที่ห้องเรียนคอมพิวเตอร์ เป็นหน่วยงานที่ให้บริการสอนโปรแกรมต่างๆ ต้องมีการติดตามการทำงานของผู้เข้าอบรมจากจอภาพที่แสดงการทำงาน ซึ่งปกติจะมีอาจารย์ผู้สอนอยู่หน้าห้องเรียน และมีเครื่องคอมพิวเตอร์สำหรับผู้เรียนประมาณ 20-30 เครื่องในห้องเรียนทำงานตามที่ได้รับมอบหมายไปพร้อม ๆ กับการเรียนการสอน ซึ่งถ้าอาจารย์ผู้สอนต้องการทราบผลงานของผู้เรียนขณะที่สอนไปด้วยก็ต้องไปดูที่หน้าจอตามจุดที่ผู้เรียนอยู่ หรือถ้าผู้เรียนมีปัญหาต้องการสอบถามอาจารย์ผู้สอนในเรื่องการเรียนการสอนที่ต้องใช้คอมพิวเตอร์ทำงาน ทำให้อาจารย์ผู้สอนต้องคอยติดตามที่หน้าจอ ตามจุดที่ผู้เรียนอยู่เช่นกัน ดังนั้น ถ้าห้องเรียนคอมพิวเตอร์ในอนาคตมีจำนวนเครื่องที่มีผู้เรียนในห้องเป็นจำนวนมากขึ้นอีก อาจทำให้ต้องใช้เวลานานและต้องจำนวนอาจารย์ผู้สอนคอยดูแลมากพอ

ดังนั้นจากปัญหาที่เกิดขึ้น ถ้าได้มีการวิเคราะห์และออกแบบการทำงานของระบบให้สามารถเฝ้าสังเกต (monitor) การทำงานบนจอภาพผู้เรียนผ่านเครือข่ายในห้องเรียนคอมพิวเตอร์ ด้วยอุปกรณ์เครือข่ายเท่าที่มีอยู่ คือ เครื่องคอมพิวเตอร์ อุปกรณ์เดินสายสัญญาณและอุปกรณ์เอ็นไอซี (Network

Interface Card) โดยไม่ต้องเพิ่มเติมฮาร์ดแวร์พิเศษอีก เช่น อุปกรณ์เดินสายสัญญาณเพิ่มเติม กล้องวิดีโอ หรือกล่องควบคุม (Control box) ต่าง ๆ ทั้งนี้เพื่อให้มีประสิทธิภาพมากพอที่จะติดตามการทำงานของผู้เรียน ก็เป็นแนวทางออกได้อีกทางหนึ่ง ซึ่งจากการศึกษาพบว่ามีโปรแกรมเฝ้าสังเกตการทำงานบนจอภาพผ่านระบบเครือข่ายบนท้องตลาดแล้ว เช่น IBM Netview, RemotelyAnywhere, Unicenter TNG, HP Openview เป็นต้น แต่จะมีลักษณะการทำงานที่จำกัดเฉพาะความต้องการและปิดกันไม่ให้มีการนำมาพัฒนาต่อได้ ดังนั้นการพัฒนาโปรแกรมขึ้นเพื่อเฝ้าสังเกตจอภาพผ่านระบบเครือข่ายในห้องเรียน จึงเป็นเรื่องจำเป็นถ้าต้องการให้ได้โปรแกรมที่ตรงตามความต้องการของห้องเรียนคอมพิวเตอร์ เช่น เฝ้าสังเกตการทำงานบนจอภาพผู้เรียนผ่านระบบเครือข่ายในห้องเรียนคอมพิวเตอร์ โดยระบุเครื่องหรือไม่ระบุเครื่องแต่ให้เวียนเฝ้าสังเกตไป, ยกตัวอย่างหน้าจอแสดงให้ผู้เรียนทุกคนในห้องดูผ่านระบบเครือข่าย เป็นต้น

1.2.วัตถุประสงค์

เพื่อวิเคราะห์ออกแบบและพัฒนาระบบเฝ้าสังเกตจอภาพของผู้เรียนผ่านเครือข่ายห้องเรียนคอมพิวเตอร์

1.3.ขอบเขตการวิจัย

1.3.1. ใช้ข้อมูลและสภาพแวดล้อมของ ห้องเรียนคอมพิวเตอร์ของภาควิชาวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ซึ่งมีเครื่องพีซีประมาณ 20-30 เครื่องเป็นกรณีศึกษา

1.3.2. เฝ้าสังเกตการทำงานบนจอภาพผู้เรียนผ่านเครือข่ายในห้องเรียนคอมพิวเตอร์ โดยระบุเครื่องหรือไม่ระบุเครื่องแต่ให้เวียนเฝ้าสังเกตไป, ยกตัวอย่างหน้าจอแสดงให้ผู้เรียนทุกคนในห้องดูผ่านระบบเครือข่าย โดยการติดต่อกันด้วยโปรโตคอลทีซีพี/ไอพี และระบุเครื่องในห้องด้วย IP Address

1.3.3. เครื่องคอมพิวเตอร์ที่ใช้งาน

1.3.3.1. เครื่องเซิร์ฟเวอร์ (Server)

1.3.3.1.1. ฮาร์ดแวร์

1.3.3.1.1.1. หน่วยประมวลผลกลาง (CPU) Pentium เป็นอย่างน้อย

1.3.3.1.1.2. หน่วยความจำหลัก (RAM) ความจุ 64 เมกกะไบท์ เป็นอย่างน้อย

1.3.3.1.1.3. ฮาร์ดดิสก์ ความจุ 500 เมกกะไบท์ เป็นอย่างน้อย

1.3.3.1.2. ซอฟต์แวร์

1.3.3.1.2.1. ระบบปฏิบัติการวินโดวส์ เอ็นที รุ่น 4.0 เป็นอย่างน้อย

1.3.3.1.2.2. สนับสนุนโพรโตคอล ทีซีพี/ไอพี (TCP/IP)

1.3.3.1.2.3. โปรแกรมเชิงวัตถุ (Visual Basic 5.0 และ/หรือ Delphi 4.0 และ/หรือ Visual C++ 5.0)

1.3.3.2. เครื่องไคล์เอ็นต์ (Client)

1.3.3.2.1. ฮาร์ดแวร์

1.3.3.2.1.1. หน่วยประมวลผลกลาง (CPU) Pentium เป็นอย่างน้อย

1.3.3.2.1.2. หน่วยความจำหลัก (RAM) ความจุ 16 เมกกะไบต์ เป็นอย่างน้อย

1.3.3.2.1.3. ฮาร์ดดิสก์ ความจุ 100 เมกกะไบต์ เป็นอย่างน้อย

1.3.3.2.2. ซอฟต์แวร์

1.3.3.2.2.1. ระบบปฏิบัติการวินโดวส์ 95 เป็นอย่างน้อย

1.3.3.2.2.2. สนับสนุนโพรโตคอล ทีซีพี/ไอพี (TCP/IP)

1.4. ขั้นตอนและวิธีการ

1.4.1. ศึกษาการทำงานของโปรแกรมระบบเฝ้าสังเกตจอภาพของผู้เรียนผ่านระบบเครือข่ายในปัจจุบัน

1.4.2. กำหนดความต้องการและรวบรวมปัญหาต่างๆ

1.4.3. ออกแบบและกำหนดโครงร่างการติดตั้งระบบ

1.4.4. พัฒนาระบบเฝ้าสังเกตจอภาพของผู้เรียนผ่านเครือข่ายห้องเรียนคอมพิวเตอร์

1.4.5. ทดสอบระบบผ่านเครือข่าย

1.4.6. วิเคราะห์ผลที่ได้จากการทดสอบ

1.4.7. สรุป เรียบเรียง และจัดทำวิทยานิพนธ์

1.5. ประโยชน์ที่คาดว่าจะได้รับ

1.5.1. ระบบเฝ้าสังเกตจอภาพของผู้เรียนของห้องเรียนคอมพิวเตอร์ผ่านเครือข่าย

1.5.2. ช่วยลดเวลาและแรงงานของบุคคลากรที่ใช้ในการเฝ้าสังเกตจอภาพของผู้เรียนผ่านเครือข่ายห้องเรียนคอมพิวเตอร์เป็นแนวทางในการพัฒนาช่วยลดค่าใช้จ่ายที่ต้องสูญเสียไปในการเฝ้าสังเกตจอภาพของผู้เรียนผ่านเครือข่ายห้องเรียนคอมพิวเตอร์ หรือระบบที่คล้ายคลึงกันในลำดับต่อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

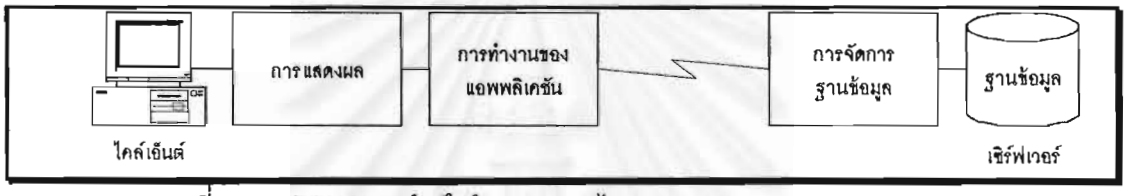
บทที่ 2
แนวความคิดและทฤษฎีที่เกี่ยวข้อง

2.1 ไคล์เอ็นต์/เซิร์ฟเวอร์ (Client/Server)

เราแบ่งรูปแบบตามส่วนประกอบทางตรรกระหว่างไคล์เอ็นต์และเซิร์ฟเวอร์ได้ 3 รูปแบบคือ

2.1.1. รูปแบบการเข้าถึงข้อมูลระยะไกล (Remote Data Access Model)

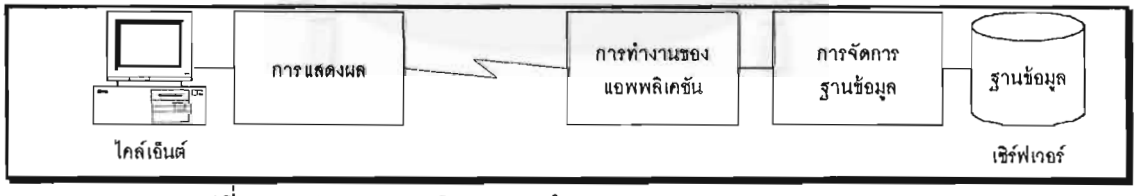
ส่วนแสดงผลและส่วนประยุกต์ข้อมูลการทำบนฮาร์ดแวร์ของไคล์เอ็นต์ ส่วนการเข้าถึงข้อมูลนั้น ไคล์เอ็นต์จะสั่งให้เซิร์ฟเวอร์ดำเนินการให้ ข้อมูลยังคงมีการย้ายโอนไปมาระหว่างไคล์เอ็นต์และเซิร์ฟเวอร์มาก



รูปที่ 2.1 รูปแบบการเข้าถึงข้อมูลระยะไกล (Remote Data Access Model)

2.1.2. รูปแบบการบริการฐานข้อมูล (Database Server Model)

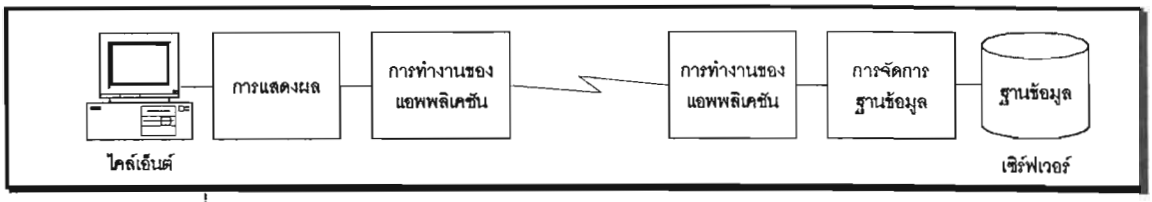
ส่วนที่เป็นไคล์เอ็นต์ทำหน้าที่เกี่ยวกับการแสดงผล และรับข้อมูลทางแป้นพิมพ์หรืออุปกรณ์อื่นๆ ส่วนที่เหลือเซิร์ฟเวอร์จะดำเนินการให้หมดในลักษณะของการเก็บคำสั่ง



รูปที่ 2.2 รูปแบบการบริการฐานข้อมูล (Database Server Model)

2.1.3. รูปแบบการบริการแอพพลิเคชัน (Application Server Model)

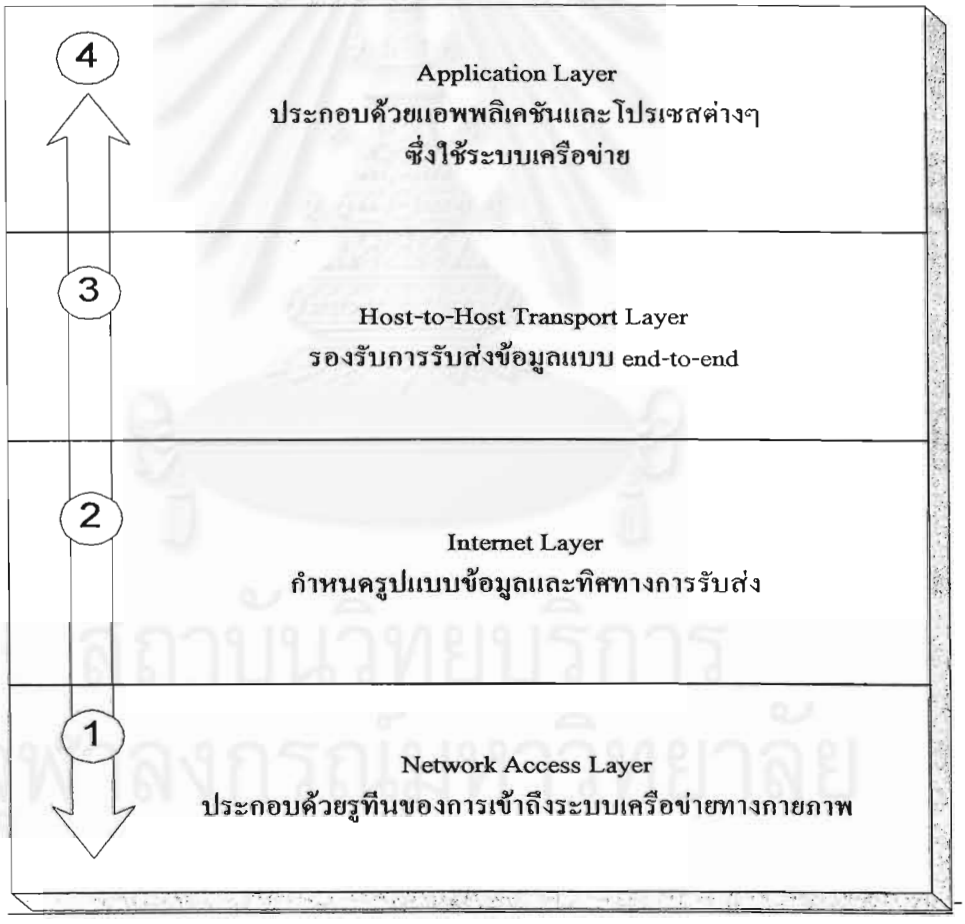
ในรูปแบบนี้ไคล์เอ็นต์จะจัดการเกี่ยวกับการแสดงผลและแป้นพิมพ์ แต่ตรรกะในทาการจัดการแอพพลิเคชันถูกกำหนดไว้ที่เซิร์ฟเวอร์ และพร้อมที่จะให้โปรแกรมส่วนของไคล์เอ็นต์เรียกใช้ได้ในลักษณะโปรแกรมสู่โปรแกรม



รูปที่ 2.3 รูปแบบการบริการแอพพลิเคชัน (Application Server Model)

2.2 ชุดโปรโตคอล ทีซีพี/ไอพี (TCP/IP)[8]

ทีซีพี/ไอพี เป็นชุดโปรโตคอลในการสื่อสารข้อมูลที่เป็นมาตรฐาน โดยมีแนวความคิดมาจากกระทรวงกลาโหมของสหรัฐอเมริกาที่ต้องการเชื่อมต่อเครื่องคอมพิวเตอร์ที่มีอยู่มากมายหลายยี่ห้อเข้าด้วยกันเป็นเครือข่ายเดียว โดยมอบหมายให้ DARPA (Defense Advanced Research Project Agency) เป็นผู้รับผิดชอบ



รูปที่ 2.4 แสดงสถาปัตยกรรมระดับชั้นของโปรโตคอลทีซีพี/ไอพี

2.2.1 สถาปัตยกรรมของโปรโตคอลทีซีพี/ไอพี

สามารถแบ่งได้เป็น 4 ระดับชั้นตามรูปมีลักษณะคล้ายกับโอเอสไอโมเดล คือ ข้อมูลจะถูกส่งลงเป็นระดับชั้นและข้อมูลอยู่ในลักษณะของระดับชั้นที่สูงขึ้นเมื่อมีการรับข้อมูลจากระบบเครือข่าย โดยแต่ละระดับชั้นจะเพิ่มข้อมูลควบคุม(Control information) เพื่อรับประกันการรับส่งที่ดีขึ้นเรียกว่า encapsulation เข้าไปในชุดข้อมูลที่จะส่ง

2.2.2 โปรโตคอลไอพี(Internet Protocol)

มีหน้าที่ต่าง ๆ ดังนี้

2.2.2.1 กำหนดรูปแบบข้อมูล(datagram)ที่เป็นหน่วยพื้นฐานของการรับส่งในอินเตอร์เน็ต

2.2.2.2 กำหนดหลักการ Internet Addressing

2.2.2.3 รับข้อมูลมาจาก Network Access Layer ไปยัง Host-to-Host Transport Layer

2.2.2.4 หาทิศทางการส่งข้อมูลไปยังโฮสต์ ที่อยู่ไกลออกไป(remote hosts)

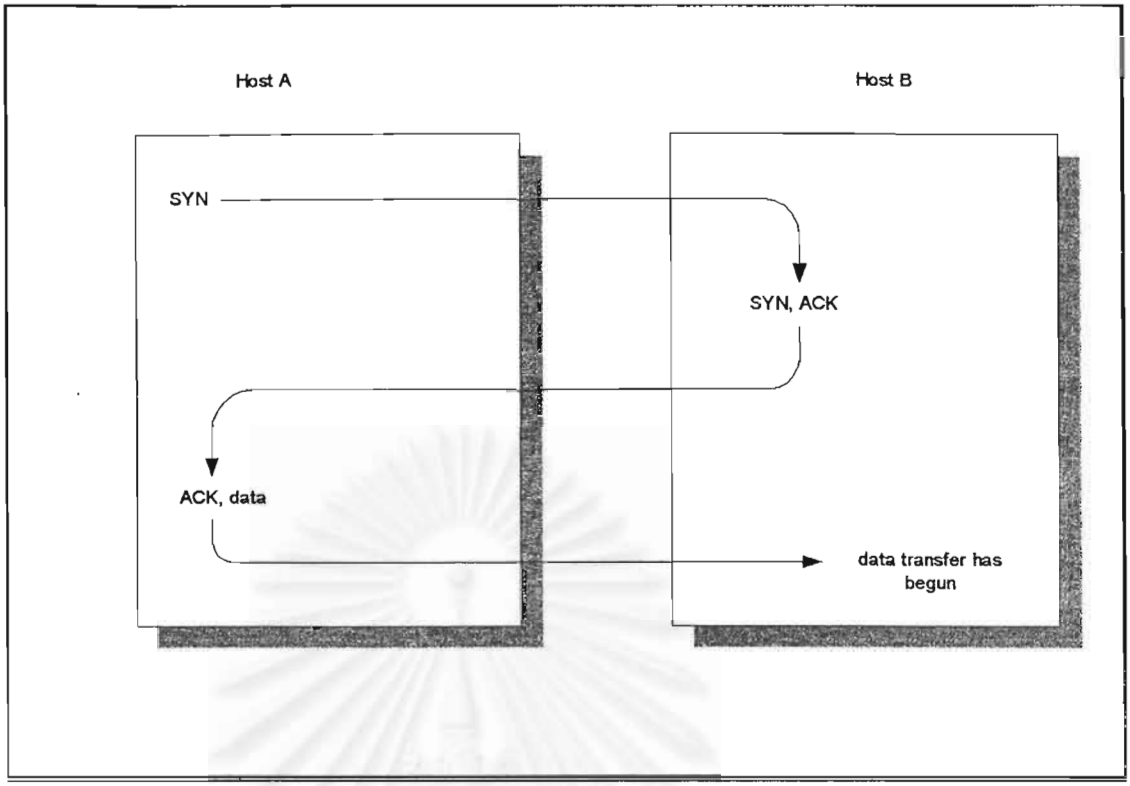
2.2.2.5 ทำการแบ่งและรวมข้อมูลระหว่างการรับส่งข้อมูล

2.2.3 โปรโตคอลทีซีพี(Transmission Control Protocol)

เป็นโปรโตคอลที่ให้ความน่าเชื่อถือในการรับส่ง(eliable, connection-oriented, byte-stream protocol) โดยจะต้องมีการสร้างเส้นทางที่จะมีการรับส่งก่อน(handshake) ดังรูปที่ 2.5

ถ้า Host A ต้องการส่งข้อมูลให้ B A จะเริ่มการติดต่อโดยส่ง "Synchronize sequence number" (SYN) bit set ให้ B เพื่อบอกว่า A จะเริ่มการส่งข้อมูลด้วยเลข SYN Host B ก็จะตอบ ACK มากับ SYN จากนั้น Host A จึงเริ่มการส่งข้อมูลได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 2.5 แสดง Three-way handshake รูปแบบง่ายที่สุด

2.3 การบีบอัดรูปภาพที่เป็นแบบเรียลทาม (Real time)

2.3.1 มาตรฐานของการบีบอัดข้อมูล[12]

การตั้งมาตรฐานขึ้นเป็นสิ่งจำเป็นสำหรับเทคโนโลยีใดๆก็ตาม เพื่อจะได้สนับสนุนสำหรับผู้ผลิตจำนวนมากในอนาคต เพราะจะทำให้ง่ายต่ออุปกรณ์ต่างๆของผู้ผลิตที่จะเข้ามามีส่วนร่วม ให้สามารถผลิตในสิ่งที่ตรงตามความต้องการ ทั้งฮาร์ดแวร์และซอฟท์แวร์ ทั้งยังขจัดสิ่งที่ไม่จำเป็นทิ้งไปด้วย การบีบอัดข้อมูลมีมาตรฐานสำหรับสำหรับรูปภาพที่กำหนดขึ้นโดย CCITT (International Consultative Committee for Telephone and Telegraph) ซึ่งมีทั้งการบีบอัดข้อมูลสื่อประสมทั้งรูปแบบ non-lossy และ lossy compression

2.3.2 ความจำเป็นที่ต้องใช้การบีบอัดข้อมูล[12]

เมื่อข้อมูลสื่อประสม (multimedia data) อย่างเช่น รูปภาพขาวดำ รูปภาพสีเทา รูปภาพสี หรือรูปจำพวกวีดีทัศน์ มีการนำมาเป็นเอกสารใช้งาน จึงทำให้เกิดข้อมูลดิจิทัลขนาดใหญ่ขึ้นมาขนาดของข้อมูลจะขึ้นอยู่กับความคมชัดจากการกราดตรวจ (resolution of scanning) ยกตัวอย่างเช่น ภาพที่เพิ่มความคมชัดจาก 200 dpi เป็น 400 และ 600 dpi ตามลำดับ ขนาดข้อมูลที่ใช้เก็บจะเพิ่มขึ้นเป็นลำดับเรขาคณิต รูปสี่เหลี่ยมจัตุรัสขนาด 1

นิ้ว ของรูป 400 dpi จะประกอบด้วย 160,000 จุดภาพ แต่ละจุดภาพอาจจะถูกแทนด้วย 1 บิต ถ้าจุดภาพเป็นสีเทา หรือเป็นภาพสีก็ต้องมีข้อมูลที่ต้องเก็บต่อจุดภาพเพิ่มขึ้นอีกมากกว่า 1 บิต รูปขนาด 8.5x11 นิ้ว มีพื้นที่ 93.5 ตารางนิ้ว ดังนั้นถ้ารูปที่ไม่มีการบีบอัดก็ต้องเก็บข้อมูลทั้งหมดขนาดหลายเมกะบิต เมื่อจะต้องมีการเก็บ ค้นคืน ส่งผ่าน และแสดงข้อมูล จึงเกิดปัญหา 2 อย่างคือ ที่เก็บข้อมูลมีไม่เพียงพอ และการส่งผ่านข้อมูลที่ขนาดใหญ่ทำให้ต้องใช้เวลานาน

แม้ว่าความเร็วในการสื่อสารในระบบเครือข่ายจะมีแนวโน้วที่เพิ่มขึ้น อย่างเช่น เทคโนโลยีเอทีเอ็มที่สื่อสารได้ในความเร็วมากกว่า 100 เมกะบิตต่อวินาที แต่ข้อมูลขนาดใหญ่ก็ยังใช้เวลาในการส่งผ่านเป็นเวลาหลายวินาที

ในการบีบอัดข้อมูลสื่อประสมขนาดใหญ่ ข้อมูลเหล่านี้จะถูกบีบอัดเพื่อลดขนาดของแฟ้มที่จะเก็บ ขั้นตอนวิธีบีบอัดหลายวิธีพยายามที่จะ ขจัดความซ้ำซ้อนของรูปแบบข้อมูลลง ตัวอย่างเช่น ถ้าจุดภาพสีดำมีติดต่อกันถึง 20 จุดภาพ ก็ไม่จำเป็นต้องเก็บข้อมูลที่เหมือนกันทั้ง 20 จุดภาพ เมื่อกลไกของรหัสนี้ถูกใช้ ความซ้ำซ้อนของรูปแบบข้อมูลก็จะถูกขจัดไป ขนาดของข้อมูลก็ลดลง ทำให้เวลาที่ต้องส่งผ่านข้อมูลบนระบบเครือข่ายลดลง เป็นต้น

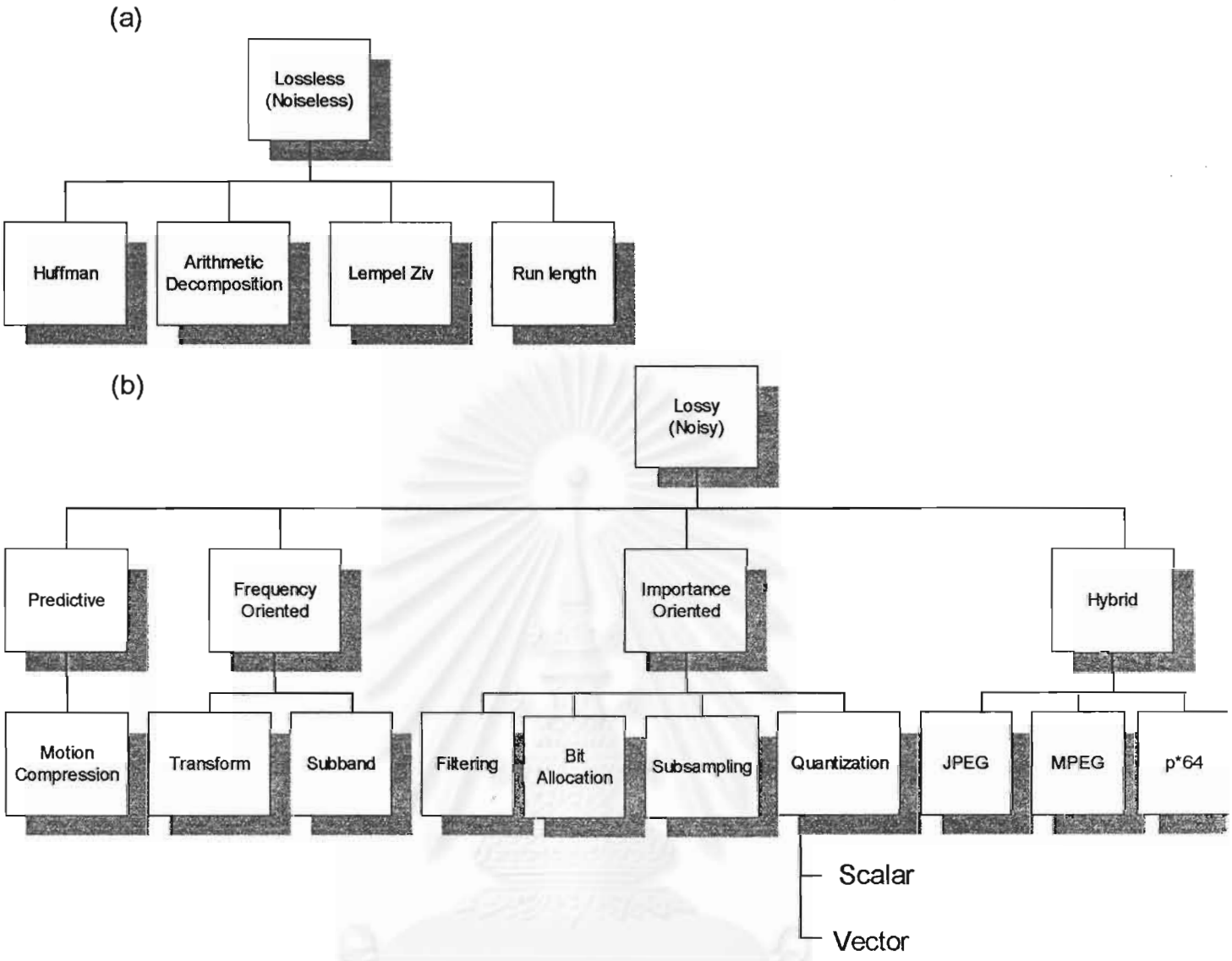
2.3.3 ประเภทของการบีบอัด(Classification of Compression Techniques)[11]
เราสามารถแบ่งเป็น 2 ประเภทใหญ่ ๆ ในการบีบอัดคือ

2.3.3.1 lossless approach

มาตรฐานสำหรับการบีบอัดข้อมูลรูปภาพแบบ non-lossy หรือ lossless ถูกออกแบบมาสำหรับข้อมูลสื่อประสมซึ่งสามารถใช้กับข้อมูล ทั้งรูปภาพ เสียง และวีดีทัศน์ กลุ่ม CCITT 2,3 และ 4 จะเป็นในรูปแบบ non-lossy

CCITT กลุ่ม 2 ได้ออกแบบหลักการสำหรับภาพที่มีความคมชัด 100 dpi แต่ไม่กล่าวถึงการบีบอัดในระดับสูง CCITT กลุ่ม 3 1D จะรู้จักกับในนาม การเข้ารหัสแบบ run-length หลักการอยู่ในพื้นฐานของสมมติฐานของการกราดตรวจตามแถว ในจุดที่มีสีซ้ำกัน แต่เนื่องจากหลักการนี้ออกแบบไว้สำหรับภาพสีขาวดำเท่านั้น แอพพลิเคชันที่ใช้จึงเป็นพวกเอกสารธรรมดา แต่จะใช้กับพวกเอกสารรูปภาพได้ไม่ดี

CCITT กลุ่ม 3 2D มีหลักการที่แก้ไขหลักการเดิม เพื่อให้แอพพลิเคชันสามารถใช้การบีบอัดกับพวกรูปภาพดีขึ้น ซึ่งก็เป็นแนวโน้มเดียวกับ CCITT กลุ่ม 4 ซึ่งมีอัตราการบีบอัดที่ดีขึ้น



รูปที่ 2.6 แสดงการจำแนกเทคนิคการบีบอัด (a) lossless (b) lossy

CCITT กลุ่ม 5 ได้ออกแบบมาตรฐานที่เข้ารหัสกับภาพสีได้

2.3.3.2 lossy approach

ขณะที่ภาพถ่ายมีการใช้ความคมที่สูง ประมาณ 1,000 จุดภาพต่อนิ้ว การบีบอัดข้อมูลที่มีขนาดใหญ่ก็ยังให้ผลลัพธ์ขนาดใหญ่ จึงเกิดแนวคิดการบีบอัดที่ยอมเสียรายละเอียดไปบ้างแต่ไม่ทำให้สังเกตได้ชัดเจนในรายละเอียด สำหรับมาตรฐานในการบีบอัดแบบนี้จะไม่ขอกล่าวในรายละเอียดเนื่องจากไม่ได้นำมาใช้อ้างอิง (ใช้กับพวกภาพที่ต้องการคมชัดสูง เช่น ภาพวีดีทัศน์ ภาพเคลื่อนไหว)

ปกตินิยมใช้เทคนิคแบบ lossy เนื่องจากให้อัตราการบีบอัดข้อมูลที่มากกว่า เราสามารถจำแนกเทคนิคแบบ lossy ได้ดังนี้

2.3.3.2.1 prediction-base techniques จะมีลักษณะที่ทำนายค่าจากการสังเกตค่าเก่า เช่น adaptive differential pulse-code modulation (ADPCM)

2.3.3.2.2 frequency-oriented techniques ใช้กับการประยุกต์ของ discrete cosine transform (DCT) ซึ่งสัมพันธ์กับ fast Fourier transform

2.3.3.2.3 Importance-oriented techniques ใช้คุณลักษณะอื่นๆ ของภาพเพื่อการบีบอัดเช่น digital video interface (DVI ™) technique จะใช้ตารางค้นหาสี (color lookup tables) และการกรองข้อมูล

2.3.3.2.4 hybrid compression techniques เช่น JPEG, MPEG และ px64 เป็นการรวมเทคนิคการบีบอัดหลาย ๆ เทคนิค เช่น DCT และ vector quantization หรือ differrential pulse-code modulation

2.4 เทคนิคการถ่ายโอนข้อมูล (Data Transfer Techniques)[13]

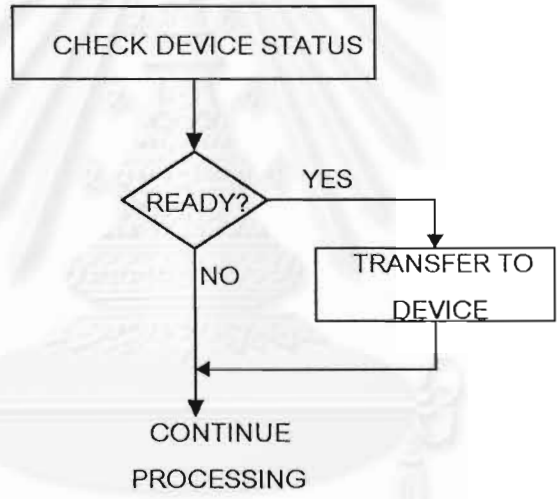
ถึงแม้ว่าความหมายในการถ่ายโอนข้อมูลจะแตกต่างกันในต่างกระบวนการ (process) กันหรือ จะมีการใช้อุปกรณ์รอบข้าง (peripheral) ที่ต่างกัน แต่ยังมีคุณลักษณะหลายประการที่มีความสัมพันธ์กันในเรื่องของการถ่ายโอนข้อมูลจากส่วนต่อประสาน (interface) ไปยังเครื่องคอมพิวเตอร์ คุณลักษณะของอุปกรณ์ต่อประสาน (interface device) ส่วนมากจะทำงานแบบประสานเวลากับการทำงานของเครื่องคอมพิวเตอร์ที่ความเร็วต่ำ การควบคุมโดยตรงจากอุปกรณ์ต่อประสานโดยเครื่องคอมพิวเตอร์จะรู้จักกันในนาม การถ่ายโอนที่ได้จากการโปรแกรมไว้ล่วงหน้า (programmed transfer) ด้วยการทำงานของหน่วยประมวลผลกลาง การถ่ายโอนที่ได้จากการโปรแกรมไว้ล่วงหน้าจะมีความยืดหยุ่นในการทำงาน แต่เนื่องจากประสิทธิภาพของหน่วยประมวลผลกลาง และอุปกรณ์ต่อประสานที่ยังไม่ดีทำให้ประสิทธิภาพการถ่ายโอนข้อมูลไม่ดีพอ ทางเลือกในการแก้ปัญหานี้คือการใช้การเข้าถึงหน่วยความจำโดยตรง (DMA : Direct Memory Access) คือการถ่ายโอนจะเกิดขึ้นเมื่อมีการควบคุมโปรแกรม ขณะที่ข้อมูลจะถูกถ่ายโอนโดยไม่รบกวนการทำงานของหน่วยประมวลผลกลาง (ยกเว้นการใช้งานเมื่อทดสอบบัส : bus clycles)

ปัญหาหลักของการถ่ายโอนข้อมูลคือเวลา ถ้าคิดในการทำงานภายใต้ การถ่ายโอนที่ได้จากการโปรแกรมไว้ล่วงหน้า คอมพิวเตอร์สามารถจะอ่านจากอุปกรณ์หรือเขียนไปยังอุปกรณ์ ณ เวลาใดก็ได้ เรียกว่าเป็นการถ่ายโอนแบบไม่มีเงื่อนไข (unconditional transfer) ตัวอย่างเช่น บางกระบวนการที่ทำงานกับอุปกรณ์ผลลัพธ์ (output device) เช่น สวิทช์ และตัวตรวจจับแสง ซึ่งต่ออยู่กับส่วนต่อประสานผลลัพธ์แบบดิจิทัล (digital output interface) ลักษณะเช่นนี้จะสามารถเกิดการถ่ายโอนแบบไม่มีเงื่อนไขได้ เพราะอุปกรณ์เหล่านี้เตรียมพร้อมที่จะรับข้อมูลตลอดเวลา สำหรับ

อุปกรณ์ผลลัพธ์อย่างอื่นเช่น เครื่องพิมพ์ และช่องทางการติดต่อสื่อสาร (communication channals) ที่ไม่มีความเร็วพอที่จะตอบสนองเครื่องคอมพิวเตอร์ จะต้องรับข้อมูลเป็นลำดับของชิ้นส่วนข้อมูล โดยต้องไม่มีการขาดหาย ลักษณะเช่นนี้ จะไม่เกิดการถ่ายโอนแบบไม่มีเงื่อนไข คอมพิวเตอร์ต้องแน่ใจก่อนว่าอุปกรณ์พร้อมที่จะรับข้อมูลในชิ้นถัดไป นั่นคือมีการใช้ทั้งการทำซ้ำการทดเวลา(timing loop) เพื่อประสานเวลากับการทำงานของเครื่องคอมพิวเตอร์กับอุปกรณ์ภายนอก และมีการใช้การถ่ายโอนแบบมีเงื่อนไข (conditional transfer) ด้วย สำหรับการถ่ายโอนแบบมีเงื่อนไขมักจะถูกใช้กับข้อมูลเข้า (input) แบบดิจิทัล มักจะไม่ใช้กับข้อมูลเข้าแบบพัลส์ (pulse) หรือข้อมูลแอนะล็อก (analog)

2.4.1 โพลลิง (polling)

ลักษณะที่ใช้ถ่ายโอนข้อมูลคือ เครื่องคอมพิวเตอร์ต้องตรวจสอบอุปกรณ์ที่จะถ่ายโอนข้อมูลต้องอยู่ในสภาพเตรียมพร้อมก่อนเสมอ ปัญหาจะเกิดขึ้นเมื่อมีการรอการถ่ายโอนแบบมีเงื่อนไข แต่จะถูกแก้ไขได้ถ้าสามารถใช้สัญญาณขัดจังหวะ (interrupt signal)



รูปที่ 2.7 แสดงตัวอย่าง Conditional transfer

2.4.2 การขัดจังหวะ (Interrupts)

เป็นกลไกที่ทำให้การไหลของการทำงานในโปรแกรมสามารถหยุดชั่วขณะ เพื่อไปทำงานพิเศษก่อนซึ่งเรียกว่า interrupt service routine หรือ ตัวจัดเรื่องขัดจังหวะ (interrupt handler) เมื่อรูทีนทำงานเสร็จแล้วโปรแกรมที่หยุดอยู่จึงทำงานต่อ

การขัดจังหวะเป็นสิ่งจำเป็นสำหรับงานที่ต้องการความถูกต้องสูง ส่วนมากเป็นพวกระบบคอมพิวเตอร์ที่เป็นเรียลทาม (real-time computer system) เพื่อใช้ในการแก้ปัญหาการรอแบบมีเงื่อนไข สำหรับงานที่ใช้การขัดจังหวะมีดังนี้

2.4.2.1 Real-time clock อุปกรณ์ฮาร์ดแวร์ภายนอกจะให้สัญญาณเป็นช่วงเวลาปกติ โดยมีตัวจัดเรื่องขัดจังหวะเป็นตัวนับสัญญาณแล้วเก็นเป็นสัญญาณนาฬิกา

2.4.2.2 Alarm inputs ตัวรับรู้(sensors) ต่างกันสามารถนำมาใช้ให้เปลี่ยนค่าในระดับตรรกะ ตามเหตุการณ์ของการเตือนได้ เนื่องจากการส่งสัญญาณเตือนไม่ถูกทำบ่อย แต่ต้องการการตอบสนองที่รวดเร็ว

2.4.2.3 Manual override ใช้การขัดจังหวะเพื่ออนุญาตให้มีการควบคุมระบบจากภายนอกเพื่อการบำรุงรักษา และซ่อมแซมระบบ

2.4.2.4 Hardware Failure indication ความผิดพลาดจากการทำงานของอุปกรณ์ฮาร์ดแวร์ภายนอกหรือหน่วยต่อประสาน สามารถส่งสัญญาณไปหน่วยประมวลผลโดยใช้การขัดจังหวะ

2.4.3 การเข้าถึงหน่วยความจำโดยตรง (Direct Memory Access)

ปกติมี 3 วิธีที่ใช้ด้วยกันคือ burst mode, distributed mode และ cycle stealing ใน burst mode ตัวควบคุมดีเอ็มเอ (DMA Controller) จะควบคุมเส้นทางข้อมูลของคอมพิวเตอร์และจองเวลาช่วงหนึ่งของหน่วยประมวลผลกลาง เพื่อการโอนถ่ายข้อมูล คือใช้ 256 ไบต์ถ่ายโอนระหว่าง fast memory และ backing memory การใช้งานใน burst mode จะมีผลมากกับระบบที่ต้องการการตอบสนองรวดเร็วกับเหตุการณ์ภายนอก

ใน distributed mode ตัวควบคุมดีเอ็มเอ จะได้โอกาสใช้เวลาหลายรอบจากการควบคุมของหน่วยประมวลผลกลางของเครื่อง ในแต่ละรอบจะใช้การถ่ายโอนข้อมูลทีละไบต์ ระหว่าง fast memory และ backing memory ในระบบที่ไม่เป็นเรียลทาม (non-real-time system) การขาดหายไปหลาย ๆ รอบของหน่วยประมวลผลกลาง ไม่สามารถจะสังเกตได้ อย่างไรก็ดีในระบบเรียลทามซึ่งใช้ซอฟท์แวร์ในการวนซ้ำเวลา เมื่อมีการเสียหลายๆรอบของหน่วยประมวลผลกลาง จะไปมีผลกับเวลาได้ ในโปรแกรมที่ไม่คำนึงถึงจำนวนรอบที่ใช้โดยตัวควบคุมดีเอ็มเอ ก็จะยังทำงานด้วยจำนวนชุดคำสั่งเท่าเดิม อย่างไรก็ดีช่วงเวลาที่ใช้อาจจะใช้จำนวนรอบถึง 200 รอบแทนที่จะเป็น 100 รอบ ตามที่คาดคะเนไว้

วิธีของ cycle stealing เป็นเพียงวิธีเดียวที่ใช้ช่วงเวลาที่หน่วยประมวลผลกลางไม่ใช้บัสทำการถ่ายโอน ดังนั้นโปรแกรมก็จะทำงานในความเร็วปกติ และไม่ได้รับผลจากการถ่ายโอนข้อมูลจากการเข้าถึงหน่วยความจำโดยตรง อย่างไรก็ดีก็เป็นวิธีที่การถ่ายโอนข้อมูลระหว่าง fast memory กับ backing store ช้าที่สุด

2.4.4 ข้อเปรียบเทียบของเทคนิคการถ่ายโอนข้อมูลแบบต่างๆ

โพลลิงซึ่งมีทั้งการรอโดยไม่เกิดงาน (busy wait) และต้องมีการตรวจสอบสถานะของอุปกรณ์ เป็นวิธีที่ง่ายที่สุด สำหรับการถ่ายโอนในแง่ของการโปรแกรมและการทดสอบ

โปรแกรม ขณะที่การขัดจังหวะด้วยซอฟท์แวร์ซึ่งจะมีลักษณะไม่เป็นรูปแบบสำหรับโปรแกรม แต่ควบคุมการทำงานได้ตรงๆ นั่นคือ มีศากยภาพในการควบคุมการถ่ายโอนทุกๆจุดในโปรแกรม

ระบบตัวขับขัดจังหวะ (Interrupt-driven systems) จะยากมากขึ้นที่จะทดสอบระบบเพราะข้อผิดพลาดหลายอย่างอาจจะเกิดขึ้นจากช่วงเวลาด้วย ด้วยหลักการง่ายๆ ที่จะทดสอบส่วนที่เป็นการขัดจังหวะของโปรแกรมคือเมื่อเกิดข้อผิดพลาดขึ้นแล้ว การเตรียมรูทีนในการทดสอบระบบที่เป็นการขัดจังหวะจึงทำได้ยาก การทดสอบที่เหมาะสมจำเป็นต้องเตรียมการทดสอบระบบที่เป็นรูปแบบสุ่มกับการขัดจังหวะที่จะเกิด แล้วนำมาวิเคราะห์ผลลัพธ์

ขณะที่อัตราการถ่ายโอนข้อมูลที่สูงจากการขัดจังหวะ จะไม่มีประสิทธิภาพเลย เนื่องจากมีส่วนสิ้นเปลืองไปจากการใช้ตัวจัดเรื่องขัดจังหวะ คือต้องมีการสำรองและกู้คืนสภาพแวดล้อมด้วยเสมอ ดังนั้นโพลลิงจึงเป็นวิธีที่ใช้กันมาก แต่ทางเลือกหนึ่งที่อัตราการถ่ายโอนข้อมูลจากฮาร์ดแวร์ย้ายไปทำแทนโดยซอฟท์แวร์คือใช้การเข้าถึงหน่วยความจำโดยตรง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3
การออกแบบโปรแกรม

โปรแกรมที่พัฒนาขึ้นนี้มุ่งให้ผู้ใช้งานได้เฝ้าสังเกตจอภาพของผู้เรียนผ่านระบบเครือข่ายในห้องเรียนคอมพิวเตอร์ ซึ่งจะต้องแสดงจอภาพเครื่องคอมพิวเตอร์ของผู้เรียนที่ต้องการสังเกตได้ ดังนั้นการเชื่อมต่อระหว่างชื่อเครื่องคอมพิวเตอร์ที่ต้องการสังเกต แอพพลิเคชันโปรโตคอลในการรับส่งข้อมูล ขั้นตอนวิธีในการรับส่งข้อมูลจอภาพ เพื่อการแสดงผลสามารถลดการรับส่งจำนวนข้อมูลมากในระบบเครือข่าย และไม่ทำให้การแสดงผลเกิดขึ้นช้าจึงเป็นปัจจัยสำคัญในการออกแบบโปรแกรม

3.1 ลักษณะข้อมูลของจอภาพและโปรโตคอลที่ใช้ในการรับส่ง

สำหรับการรับส่งข้อมูลของจอภาพผ่านระบบเครือข่าย จะใช้การรับส่งในลักษณะที่มีการกำหนดขึ้นเป็นโปรโตคอลในระดับแอพพลิเคชัน โดยในขั้นต้นต้องมีการกำหนดที่พักข้อมูลในเครือข่าย(network buffer) เป็นรูปแบบที่เหมาะสมกับงานนี้ก่อน[1] เพื่อให้การรับส่งข้อมูลในระบบเครือข่ายมีประสิทธิภาพเพียงพอกับคุณภาพของการแสดงผล และความรวดเร็วที่สัมพันธ์กับภาพที่จะแสดงให้เห็น

โปรแกรมที่พัฒนาขึ้นนี้ใช้การออกแบบที่พักข้อมูลและแอพพลิเคชันโปรโตคอลของ อาร์ เอฟ บี [2] มาช่วยพัฒนา ซึ่งจะใช้ในการอธิบายลักษณะรูปแบบข้อมูลที่จะใช้ในการรับส่งข้อมูลในระดับแอพพลิเคชัน โดยโปรโตคอลจะถูกใช้ตั้งแต่เริ่มต้นก่อนที่จะมีการเชื่อมโยงติดต่อกันในระบบเครือข่าย ในระหว่างที่มีการติดต่อกันก็จะมีการรับส่งข้อมูลจอภาพกัน จนจบสิ้นการเชื่อมโยงติดต่อกันในระบบเครือข่าย

Thierry Turletti และ Christian Huitema[14] กล่าวว่า ข้อมูลที่รับส่งในระบบเครือข่ายที่มีขนาดใหญ่ที่สุด (MTU) จะมีขนาดขึ้นอยู่กับชนิดของระบบเครือข่ายเช่น Ethernet ใช้ 1536 ไบต์ ขณะที่ Internet ใช้เพียงแค่ 576 ไบต์เท่านั้น

ดังนั้นการออกแบบลักษณะข้อมูลที่จะใช้รับส่งในโปรแกรมนี้จะใช้รองรับระบบเครือข่าย Internet เพื่อสนับสนุนกับวัตถุประสงค์ที่จะใช้กับการอ้างอิงด้วย Internet Protocol Address จึงใช้ขนาดความยาวของข้อมูลที่ไม่เกิน 576 ไบต์ ซึ่งจะได้ระบุไว้ในโปรโตคอลอาร์ เอฟ บี [2]

เครื่องมือและวิธีการที่ใช้ในการออกแบบโปรแกรม

โปรแกรมที่พัฒนาจะพัฒนาด้วยภาษา C++ โดยไมโครซอฟท์วิชวล C++ รุ่น 5.0 (Microsoft Visual C++ 5.0) ส่วนโครงสร้างหลักของโปรแกรม ส่วนการติดต่อกับผู้ใช้จะใช้คลาส

พื้นฐานของไมโครซอฟท์ (Microsoft Foundation Class หรือ MFC) ดังนั้นการพัฒนาโปรแกรมจึงต้องใช้วิธีการโปรแกรมเชิงวัตถุ (Object Oriented Programming)

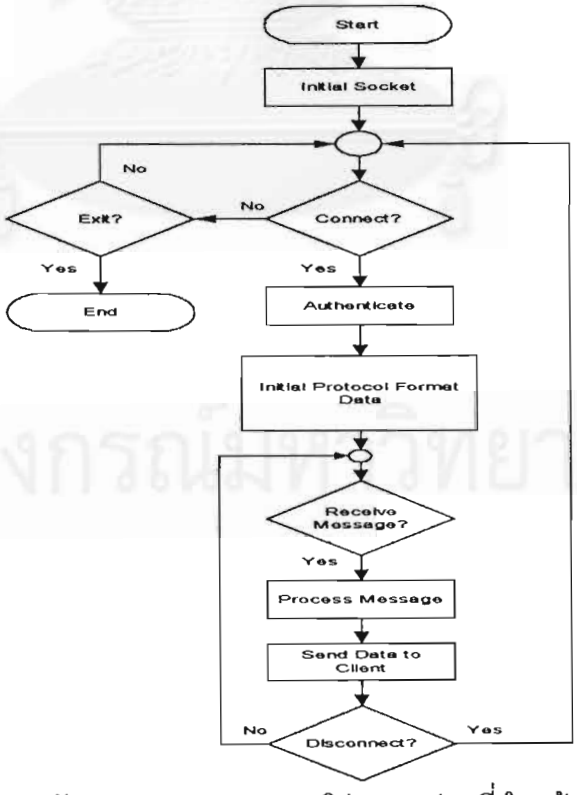
แนวทางการออกแบบโปรแกรม โปรแกรมเฝ้าสังเกตจอภาพของผู้เรียนผ่านระบบเครือข่ายในห้องเรียนคอมพิวเตอร์มีจุดประสงค์เพื่อเฝ้าสังเกตจอภาพของผู้เรียนที่ต้องการสังเกตได้ โดยระบุชื่อเครื่องที่ต้องการหรือเฝ้าสังเกตจอภาพของผู้เรียนเป็นหมู่คณะพร้อมกันหลายจอภาพ พร้อมกับสามารถแสดงจอภาพของผู้สอนให้ผู้เรียนเฝ้าสังเกตได้

ในการพัฒนาโปรแกรมจะแบ่งการทำงานออกเป็น 2 ส่วนคือ

- ส่วนโปรแกรมที่ทำหน้าที่เป็นเซิร์ฟเวอร์
- ส่วนโปรแกรมที่ทำหน้าที่เป็นไคล์เอ็นต์

3.2 การออกแบบโปรแกรมส่วนที่ทำหน้าที่เป็นเซิร์ฟเวอร์

ในส่วนโปรแกรมที่ทำหน้าที่เป็นเซิร์ฟเวอร์ หรือ ส่วนโปรแกรมที่เป็นฝ่ายส่งจอภาพของตนเองให้อีกฝ่ายหนึ่งเฝ้าสังเกต คือ ส่วนโปรแกรมที่ต้องนำไปติดตั้งที่เซิร์ฟเวอร์ ทำหน้าที่คอยส่งข้อมูลหน้าจอของตัวเองที่มีการเปลี่ยนแปลงให้กับไคล์เอ็นต์ในลักษณะที่เกิดขึ้นตามเหตุการณ์ ทำให้ไคล์เอ็นต์สามารถรับทราบจอภาพที่เห็นที่เซิร์ฟเวอร์ได้เหมือนกับอยู่หน้าเครื่องเซิร์ฟเวอร์



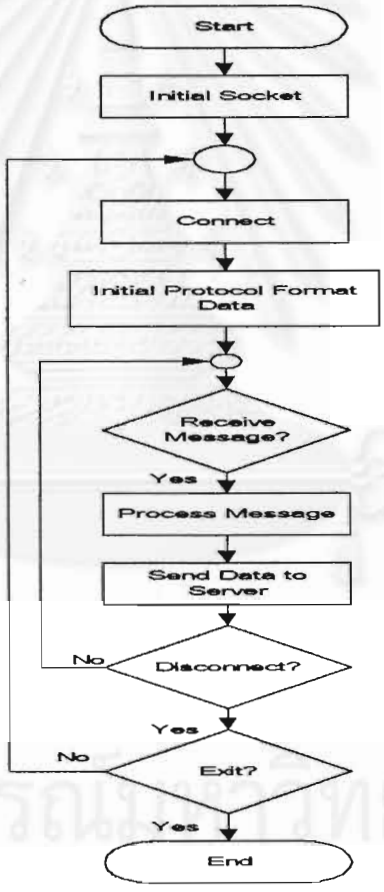
รูปที่ 3.1 แสดงผังงานของการออกแบบโปรแกรมส่วนที่ทำหน้าที่เป็นเซิร์ฟเวอร์

สำหรับแต่ละเซิร์ฟเวอร์หลังจากได้ทำการติดต่อกับไคล์เอ็นต์เพื่อเตรียมการรับส่งข้อมูลเป็นที่เรียบร้อยแล้วก็จะมีฟังก์ชันหน้าที่การทำงานเหมือนกันทุกเซิร์ฟเวอร์ ดังนี้ (ดูในรายละเอียดโปรโตคอล อาร์ เอฟ บี [1] เพิ่มเติม)

InitVersion	เป็นฟังก์ชันที่ทำหน้าที่ส่งหมายเลขรุ่นของโปรโตคอล อาร์ เอฟ บีให้กับไคล์เอ็นต์เพื่อเป็นการตกลงกันให้สามารถเข้าใจในการสื่อสารกันก่อน
InitAuthenticate	เป็นฟังก์ชันที่ทำหน้าที่ในการกำหนดและให้สิทธิในการให้เข้าใช้ หรือเฝ้าสังเกตจอภาพของเซิร์ฟเวอร์
Init Pixel Format ,screen format	เป็นฟังก์ชันที่ทำหน้าที่กำหนดรูปแบบข้อมูลจุดภาพและรูปแบบข้อมูลหน้าจอที่เหมาะสมเพื่อใช้ในการรับส่ง
Get desktop name	เป็นฟังก์ชันที่ทำหน้าที่เข้าถึงข้อมูลของระบบ ในที่นี้จะได้ข้อมูลของชื่อเครื่องคอมพิวเตอร์ของเซิร์ฟเวอร์เองพร้อมกับชื่อผู้ใช้ขณะนั้น ถ้าไม่มีการใส่ชื่อผู้ใช้ก่อนเข้าระบบเครือข่าย ก็จะแทนคำที่มีความหมายว่าไม่ได้ผ่านเข้าระบบ
Send server format to client	เป็นฟังก์ชันที่ทำหน้าที่ส่งข้อความ rfbServerInit เพื่อบอกรูปแบบข้อมูลจุดภาพและรูปแบบข้อมูลหน้าจอที่จะใช้แสดงผลไปยังไคล์เอ็นต์
Add full screen update to client's update list	เริ่มส่งข้อมูลหน้าจอของเซิร์ฟเวอร์ให้ไคล์เอ็นต์ในรูปแบบเต็มทั้งจอภาพ
Message Loop	ทำซ้ำการรอรับข้อมูลจากไคล์เอ็นต์เพื่อทำการเปลี่ยนแปลงข้อมูลในเซิร์ฟเวอร์โดยรอรับข้อความต่างๆดังนี้ • rfbSetPixelFormat เมื่อได้รับข้อความนี้ก็จะเปลี่ยนแปลงรูปแบบข้อมูลจุดภาพที่จะใช้ในการรับส่งตามที่ไคล์เอ็นต์ระบุมา • rfbSetEncodings เมื่อได้รับข้อความนี้ก็จะเปลี่ยนแปลงรูปแบบการเข้ารหัสข้อมูลจุดภาพที่จะใช้ในการรับส่งตามที่ไคล์เอ็นต์ระบุมา • rfbFramebufferUpdateRequest เมื่อได้รับข้อความนี้ก็จะส่งข้อความ rfbFramebufferUpdate กลับไปให้ไคล์เอ็นต์เพื่อบอกจุดภาพที่มีการเปลี่ยนแปลง • rfbKeyEvent เมื่อได้รับข้อความนี้ก็จะส่งข้อความให้กับ

	วินโดว์ตามที่ระบุเป็นอักขระ ASCII และทำงานเหมือนกับได้กดแป้นอักขระที่เครื่องเซิร์ฟเวอร์เอง • rfbPointerEvent เมื่อได้รับข้อความนี้ก็จะกำหนดตำแหน่งของพอยเตอร์ใหม่ให้กับวินโดว์พร้อมกับจำลองการกดพอยเตอร์ และทำงานเหมือนกับได้กดพอยเตอร์ที่เครื่องเซิร์ฟเวอร์เอง • rfbClientCutText เมื่อได้รับข้อความนี้ก็จะส่งข้อมูลบัฟเฟอร์ของสายอักขระที่ได้รับมาให้กับวินโดว์เพื่อใช้กับ Clipboard

3.3 การออกแบบโปรแกรมส่วนที่ทำหน้าที่เป็นไคล์เอ็นต์



รูปที่ 3.2 แสดงผังงานของการออกแบบโปรแกรมส่วนที่ทำหน้าที่เป็นไคล์เอ็นต์

ในส่วนโปรแกรมที่ทำหน้าที่เป็นไคล์เอ็นต์ หรือส่วนโปรแกรมที่ทำหน้าที่เฝ้าสังเกตจอภาพของผู้เรียนต้องนำไปติดตั้งในเครื่องคอมพิวเตอร์ไคล์เอ็นต์ หน้าที่หลักก็คือการนำข้อมูลจอภาพที่รับมาจากเซิร์ฟเวอร์มาแสดงผลให้ทันต่อเหตุการณ์ เหมือนกับที่เซิร์ฟเวอร์แสดงผล นอกจากนี้

เนื่องจากโปรแกรมที่ทำหน้าที่เฝ้าสังเกตจอภาพของผู้เรียนนี้เป็นส่วนที่ต้องแแสดงส่วนต่อประสานกราฟิกกับผู้ใช้ (graphic user interface) ให้ผู้ใช้เห็นและใช้งานได้สะดวก ในที่นี้จึงจะพิจารณาแยกโปรแกรมส่วนนี้ออกเป็นส่วนต่อประสานกราฟิกกับผู้ใช้ และส่วนหน้าที่หลักของโปรแกรม

ส่วนหน้าที่หลักของโปรแกรมเฝ้าสังเกตจอภาพของผู้เรียน จะมีฟังก์ชันการทำงานคล้ายกับส่วนโปรแกรมที่เซิร์ฟเวอร์ สำหรับโปรแกรมนี้ก็จะมีหน้าที่การทำงาน ดังนี้ (ดูในรายละเอียดโปรโตคอล อาร์ เอฟ บี[2] เพิ่มเติม)

GetConnectionDetails	เป็นฟังก์ชันที่ทำหน้าที่รับค่าของชื่อเครื่องคอมพิวเตอร์ในระบบเครือข่ายท้องถิ่น (LAN) พร้อมกับ port เพื่อการติดต่อไปยังเซิร์ฟเวอร์ที่ต้องการ
Connect	เป็นฟังก์ชันที่ทำหน้าที่ติดต่อกับโปรแกรมที่ส่วนของเซิร์ฟเวอร์โดยใช้ชื่อเครื่องคอมพิวเตอร์ที่ได้รับมาจากส่วนต่อประสานกราฟิกกับผู้ใช้ แปลงเป็น IP Address พร้อมกับ port ที่ระบุ แล้วติดต่อไปด้วย socket[9] เป็นทางเชื่อมต่อในการรับส่งข้อมูลกัน
NegotiateProtocolVersion	เป็นฟังก์ชันที่ทำหน้าที่รับเอาหมายเลขรุ่นโปรโตคอลอาร์ เอฟ บี เพื่อพิจารณาตกลงกันก่อนที่จะยอมให้มีการส่งข้อมูลจุดภาพกันได้
Authenticate	เป็นฟังก์ชันที่ทำหน้าที่รับรหัสผ่านจากผู้ใช้เพื่อจะได้สิทธิในการเฝ้าสังเกตจอภาพของเซิร์ฟเวอร์เมื่อเปรียบเทียบกับรหัสที่มีอยู่จากเซิร์ฟเวอร์แล้ว
CreateDisplay	เป็นฟังก์ชันที่ทำหน้าที่สร้างวินโดว์เปล่าขึ้นมารอการส่งข้อมูลจอภาพก่อน
SendClientInit	เป็นฟังก์ชันที่ทำหน้าที่ส่งข้อความไปบอกเซิร์ฟเวอร์ให้เริ่มการส่งข้อมูลจอภาพกันได้
ReadServerInit	เป็นฟังก์ชันที่ทำหน้าที่รับข้อมูลจอภาพเริ่มต้นในรูปแบบที่ตกลงกันไว้จากเซิร์ฟเวอร์ เช่น ความกว้าง ความยาว ของจอภาพที่เซิร์ฟเวอร์ จำนวนบิตต่อจุดภาพที่เซิร์ฟเวอร์ใช้ เพื่อนำมาคำนวณและวาดหน้าจอที่ไคล์เอ็นต์ได้ถูกต้อง
CreateLocalFramebuffer	เป็นฟังก์ชันที่ทำหน้าที่สร้างบัฟเฟอร์รับข้อมูลตามรูปแบบที่ระบุไว้ในครั้งแรกที่มีการส่งข้อมูลจอภาพ พร้อมกับวาดจอภาพรอสำหรับรอการรับส่งจริง
SetupPixelFormat	เป็นฟังก์ชันที่ทำหน้าที่กำหนดค่าของรูปแบบจุดภาพที่ต้องการจะใช้ในการรับส่งข้อมูลโดยกำหนดได้จากหน้าจอ Connection

	Options ถ้าไม่มีการกำหนดก็จะใช้รูปแบบจุดภาพที่เซิร์ฟเวอร์กำหนดให้
SetFormatAndEncodings	เป็นฟังก์ชันที่ทำหน้าที่ส่งรูปแบบของข้อมูลจุดภาพที่จะใช้ในการรับส่งกัน พร้อมทั้งส่งลักษณะการเข้ารหัสของข้อมูลที่จะใช้ในการรับส่งไปยังเซิร์ฟเวอร์ด้วย
Start_undetached	เป็นฟังก์ชันที่ทำหน้าที่ส่งข้อความไปบอกให้วินโดว์เริ่มการทำงานได้เอง ตามแนวคิดของการโปรแกรมเชิงวัตถุโดยรับและแปลข้อความที่ได้รับมาตามเหตุการณ์ของวินโดว์นั้น เช่น ถ้ามีข้อความ WM_PAINT จากคำสั่ง InvalidateRect วินโดว์ก็จะทำการวาดจอภาพของตนเองใหม่ให้ทันต่อเหตุการณ์ ฉะนั้นการแสดงผลเพื่อเฝ้าสังเกตจอภาพของผู้เรียนผ่านระบบเครือข่ายก็จะเริ่มขึ้นจากฟังก์ชันนี้ได้

ส่วนต่อประสานกราฟิกกับผู้ใช้จะมีลักษณะเป็นรายการเลือกให้ผู้ใช้งานสามารถเลือกหน้าจอของเครื่องคอมพิวเตอร์ที่อยู่ในระบบเครือข่ายเพื่อเฝ้าสังเกตได้ โดยที่หน้าจอหลักของโปรแกรมจะแบ่งเป็นส่วนแสดงชื่อเครื่องคอมพิวเตอร์ที่อยู่ในระบบเครือข่ายใน workgroup เดียวกันทั้งหมดในรูปลักษณะต้นไม้ (tree-view)ให้สามารถเลือกได้ และส่วนพื้นที่แสดงหน้าจอคอมพิวเตอร์ที่จะเฝ้าสังเกต ในส่วนพื้นที่ที่จะแสดงหน้าจอจะเตรียมไว้เพื่อแสดงหน้าจอขนาดย่อจากของจริงลงมา 5 เท่า เพื่อจะสังเกตจอภาพได้คราวละหลายเครื่องคอมพิวเตอร์ นอกจากนี้ก็จะมีรายการเลือกคำสั่งให้ผู้ใช้ได้เลือกที่ส่วนหัวของรายการเลือก และที่ต้นไม้ของส่วนแสดงชื่อเครื่องคอมพิวเตอร์ที่จะเฝ้าสังเกต ยกเลิกการติดต่อ เวียนเฝ้าสังเกต แสดงหน้าจอที่เฝ้าสังเกตให้ผู้เรียนที่เครื่องคอมพิวเตอร์เครื่องอื่นในระบบเครือข่ายห้องเรียนได้เฝ้าสังเกตได้ เป็นต้น

3.4 ขั้นตอนวิธีในการรับส่งข้อมูลหน้าจอระหว่างเซิร์ฟเวอร์กับไคล์เอ็นต์

ในการรับส่งข้อมูลจอภาพของเครื่องเซิร์ฟเวอร์เพื่อแสดงผลที่ไคล์เอ็นต์ ต้องมีความต่อเนื่องของการแสดงภาพให้เห็นเหมือนกับที่เครื่องเซิร์ฟเวอร์ ดังนั้นจึงต้องมีขั้นตอนวิธีเพื่อจัดการการรับส่งข้อมูลหน้าจอโดยจะได้นำเสนอขั้นตอนวิธีของกระบวนการเปรียบเทียบและส่งภาพหน้าจอที่มีการเปลี่ยนแปลงไปให้กับไคล์เอ็นต์ ซึ่งขั้นตอนวิธีนี้จะอยู่ในส่วนของโปรแกรมที่เซิร์ฟเวอร์ส่วนในฝ่ายไคล์เอ็นต์จะรับข้อมูลภาพมาแปลความหมายและวาดลงในหน้าจอของตนเองในภาย

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

หลัง สำหรับขั้นตอนวิธีที่ฝ่ายเซิร์ฟเวอร์ใช้จะเปรียบเทียบกับข้อมูลที่มีอยู่ในบัฟเฟอร์ ก่อนที่จะส่งเฉพาะส่วนที่เปลี่ยนแปลงไปให้ไคล์เอ็นต์ มีดังนี้

1. กำหนดขนาดกรอบข้อมูลที่จะใช้สำหรับอ่านข้อมูลจากบัฟเฟอร์ขึ้นมาเปรียบเทียบพร้อมกับกำหนดค่าเริ่มต้นของตำแหน่งข้อมูลในสี่เหลี่ยมผืนผ้าของจอภาพที่จะเริ่มอ่าน และตำแหน่งเริ่มต้นของบัฟเฟอร์ที่เก็บข้อมูลเก่าที่เคยอ่าน
2. เปรียบเทียบตำแหน่งข้อมูลในบัฟเฟอร์ที่อ่านไปตามแนวตั้งของจอภาพ ถ้ายังไม่เกินขอบล่างของจอภาพให้ทำต่อไป มิเช่นนั้นจบขั้นตอนวิธี
3. คำนวณตำแหน่งข้อมูลในบัฟเฟอร์ที่มุมบนซ้ายของกรอบข้อมูลตำแหน่งเดียวกันทั้งในบัฟเฟอร์เก่าและปัจจุบัน แล้วหาตำแหน่งขอบล่างของกรอบข้อมูลไว้ด้วย
4. เปรียบเทียบตำแหน่งข้อมูลในบัฟเฟอร์ที่อ่านไปตามแนวนอนของจอภาพ ถ้ายังไม่เกินขอบขวาของจอภาพให้ทำต่อไป มิเช่นนั้นไปทำ 13.
5. คำนวณตำแหน่งแถวเริ่มต้นของข้อมูลในกรอบข้อมูลที่กำลังพิจารณา ทั้งในบัฟเฟอร์เก่าและปัจจุบันและหาตำแหน่งขอบขวาของกรอบข้อมูลไว้ก่อน
6. เปรียบเทียบแถวของข้อมูลในกรอบข้อมูลที่กำลังพิจารณา ถ้ายังไม่เกินขอบล่างของกรอบข้อมูลที่กำลังพิจารณาให้ทำต่อไป มิเช่นนั้นไปทำ 12.
7. เปรียบเทียบข้อมูลทั้งแถวในกรอบข้อมูลที่กำลังพิจารณา ระหว่างในบัฟเฟอร์เก่าและใหม่ ถ้าไม่เหมือนกันให้ทำต่อไป มิเช่นนั้นเลื่อนแถวของข้อมูลในกรอบข้อมูลที่กำลังพิจารณาลงมาอีก 1 แถว แล้วไปทำ 6.
8. เพิ่มแถวข้อมูลที่มีการเปลี่ยนแปลงลงในสี่เหลี่ยมผืนผ้าที่เก็บเฉพาะส่วนที่มีการเปลี่ยนแปลง เพื่อเตรียมส่งให้กับไคล์เอ็นต์ต่อไป
9. คำนวณตำแหน่งแถวของข้อมูลในกรอบข้อมูลที่กำลังพิจารณา ทั้งในบัฟเฟอร์เก่าและปัจจุบัน
10. สำรองข้อมูลจากบัฟเฟอร์ปัจจุบันลงในบัฟเฟอร์เก่า เฉพาะในกรอบข้อมูลที่กำลังพิจารณา
11. ไปทำ 6.
12. เลื่อนกรอบข้อมูลไปทางขวาด้วยขนาดของกรอบข้อมูลที่กำหนดตอนเริ่มต้น แล้วไปทำ 4.
13. เลื่อนกรอบข้อมูลไปข้างล่างด้วยขนาดของกรอบข้อมูลที่กำหนดตอนเริ่มต้น แล้วไปทำ 2.

บทที่ 4
การพัฒนาโปรแกรม

ในการพัฒนาโปรแกรมนี้ได้เลือกใช้ไมโครซอฟท์วิชวล C++ ซึ่งมีเครื่องมือในการช่วยสร้างโปรแกรม และมีคลาสพื้นฐานของไมโครซอฟท์ (Microsoft Foundation Class หรือ MFC) ในการทำงานด้านต่างๆ เช่น ส่วนติดต่อกับผู้ใช้ ดังนั้นการจัดทำโปรแกรมจึงต้องใช้วิธีการโปรแกรมเชิงวัตถุ (Object Oriented Programming) มีคลาสพื้นฐานของไมโครซอฟท์ ได้แก่

4.1 คลาสพื้นฐานของไมโครซอฟท์ที่นำมาใช้

CObject	เป็นคลาสพื้นฐานของ MFC เป็นต้นแบบของคลาสส่วนใหญ่ และสามารถเป็นคลาสพื้นฐานของคลาสที่จะสร้างขึ้นใหม่เองได้
CCmdTarget	เป็นคลาสพื้นฐานของ MFC ที่ทำหน้าที่ด้านการจัดการแมสเสจของวินโดว์ เช่น การรับแมสเสจของปุ่มคำสั่ง
CWinApp	เป็นคลาสพื้นฐานสำหรับแอพพลิเคชันทำหน้าที่เป็นโครงของโปรแกรมประยุค ฉะนั้นแต่ละโปรแกรมจะมีวัตถุจากคลาสนี้เพียงวัตถุเดียว
CDocTemplate	เป็นคลาสพื้นฐานของที่ทำหน้าที่ในการจัดความสัมพันธ์ระหว่างคลาส CDocument CView และ CFrameWnd หรือคลาสที่ถ่ายทอดลักษณะมาจากคลาสเหล่านี้
CMultiDocTemplate	เป็นคลาสที่ถ่ายทอดลักษณะมาจากคลาส CDocTemplate สำหรับโปรแกรมแบบเอกสารหลายรูปแบบ (Multiple Document Interface หรือ MDI)
CDocument	เป็นคลาสที่จัดการเกี่ยวกับเอกสารของแอพพลิเคชัน เช่น การเปิด ปิดแฟ้มข้อมูล การอ่านแฟ้มข้อมูล
CWnd	เป็นคลาสพื้นฐานในการจัดการวินโดว์ทั่วๆไปเช่น การสร้างวินโดว์ ลักษณะของวินโดว์แต่ละวินโดว์
CFrameWnd	เป็นคลาสที่ถ่ายทอดลักษณะมาจากคลาส CWnd สำหรับเป็นวินโดว์หลักของโปรแกรมแบบเอกสารรูปเดียว (Single Document Interface หรือ SDI)
CMDIChildWnd	เป็นคลาสของวินโดว์สำหรับแสดงเอกสารสำหรับโปรแกรมแบบเอกสารหลายรูปแบบ วินโดว์ที่ถ่ายทอดลักษณะมาจากคลาสนี้จะสามารถอยู่ใน

	วินโดว์หลักของโปรแกรมเท่านั้น
CMDIFrameWnd	เป็นคลาสของวินโดว์หลักสำหรับโปรแกรมแบบเอกสารหลายรูปแบบ
CDialog	เป็นคลาสพื้นฐานที่แสดง dialog box
CView	เป็นคลาสพื้นฐานสำหรับคลาสที่ใช้ในการแสดงเอกสาร คลาสที่ถ่ายทอดลักษณะมาจากคลาสนี้จะอยู่ภายในวัตถุที่ถ่ายทอดลักษณะมาจากคลาส CFrame

4.2 คลาสที่ถ่ายทอดคุณสมบัติมาจากคลาสพื้นฐานของไมโครซอฟท์ และคลาสที่สร้างขึ้นเอง

4.2.1 คลาสที่อยู่ที่ฝั่งเซิร์ฟเวอร์ที่สำคัญ

vncBuffer	เป็นวัตถุที่ทำให้เกิดการสำรองข้อมูลจอภาพของไคล์เอ็นต์ และยังบอกได้ว่าบิตใดที่มีการเปลี่ยนแปลงไปถ้ากำหนด region มาให้ โดยการอ่านข้อมูลจอภาพจะเรียกใช้คลาส vncDesktop ทำให้
vncClient	มี 1 วัตถุต่อ 1 ไคล์เอ็นต์ และดูแลทุกอย่างที่เกิดขึ้นในไคล์เอ็นต์นั้น เช่น รับข้อมูลเข้าจาก socket และเก็บข้อมูลลงบัฟเฟอร์ โดยหน้าที่หลักจะมี -รับการร้องขอจากไคล์เอ็นต์ที่เชื่อมต่ออยู่ และตอบกลับ -ทำการเปลี่ยนแปลงจอภาพโดยใช้คลาส vncBuffer ในการดูแลหน้าจอที่เปลี่ยนแปลง แล้วส่งข้อมูลต่อให้คลาส vncDesktop และ vncServer ในการติดต่อกัน
vncDesktop	เป็นคลาสที่ดูแลการดึงค่าข้อมูลจากบัฟเฟอร์ของจอภาพให้ได้ข้อมูลของการเคลื่อนไหวของเม้าส์และการเปลี่ยนแปลงของจอภาพ
vncEncoder	เป็นคลาสที่เข้ารหัสบัฟเฟอร์ของจอภาพที่เก็บเป็น region เพื่อการส่งให้ไคล์เอ็นต์
vncEncodeCoRRE	เป็นคลาสที่ถ่ายทอดลักษณะมาจากคลาส vncEncoder ทำการเข้ารหัสย่อข้อมูลในการส่งข้อมูลสี่เหลี่ยมผืนผ้าในจอภาพให้ไคล์เอ็นต์ โดยการเข้ารหัสจะใช้แบบ Compact RRE ซึ่งจะทำให้จำนวนไบต์ที่แทนข้อมูลแต่ละสี่เหลี่ยมผืนผ้าน้อยลง เนื่องจากกำหนดให้สี่เหลี่ยมย่อยหนึ่งมีขนาดใหญ่ที่สุดได้แค่ 256 จุดภาพ ทั้งทางกว้างและยาว
vncEncodeHext	เป็นคลาสที่ถ่ายทอดลักษณะมาจากคลาส vncEncoder ทำการเข้ารหัสย่อข้อมูลในการส่งข้อมูลสี่เหลี่ยมผืนผ้าในจอภาพให้ไคล์เอ็นต์ โดยการเข้ารหัสจะใช้แบบ Hextile

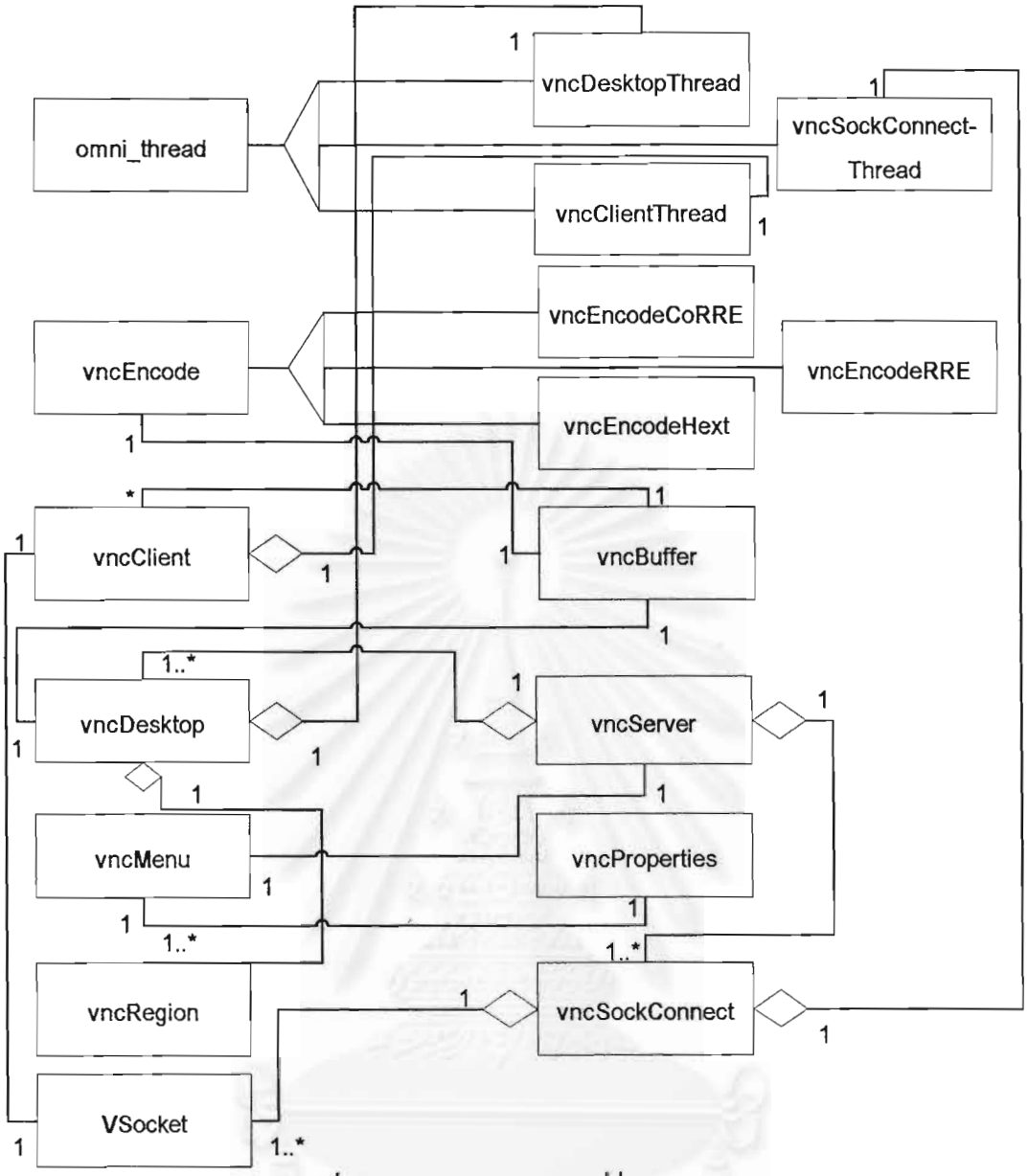
vncEncodeRRE	เป็นคลาสที่ถ่ายทอดลักษณะมาจากคลาส vncEncoder ทำการเข้ารหัสย่อข้อมูลในการส่งข้อมูลสี่เหลี่ยมผืนผ้าในจอภาพให้ไคล์เอ็นต์ โดยการเข้ารหัสจะใช้แบบ RRE
KeyMap	เป็นคลาสที่แทนค่าอักขระตัวอักษรเป็นรหัสเพื่อการเรียกใช้
vncMenu	เป็นคลาสที่สร้าง icon ที่ system tray และทำรายการเลือกขึ้นมา
vncPasswd	เป็นคลาสที่ทำการเข้ารหัสลับในการขอเข้าใช้ระบบต่อ 1 หน้าจอ
vncProperties	เป็นคลาสที่จัดการสร้างตัวเลือกคุณสมบัติต่างๆ ของระบบพร้อมกับตรวจสอบตัวเลือก
vncRegion	เป็นคลาสที่ทำหน้าที่สร้างลิสของสี่เหลี่ยมผืนผ้าบนจอภาพที่ต่างกัน โดยไม่ให้มีส่วนเหลื่อมล้ำกันอยู่
vncServer	เป็นคลาสที่มีหน้าที่ดูแลสิ่งเหล่านี้คือ -เตรียมให้ไคล์เอ็นต์สามารถเฝ้าสังเกตหรือยกเลิกการเฝ้าสังเกตเครื่องตัวเองได้ -ส่งข้อมูลการเปลี่ยนแปลงหน้าจอตัวเองให้ไคล์เอ็นต์ -รับข้อมูลการเปลี่ยนแปลงของเม้าส์และแป้นอักขระจากไคล์เอ็นต์มา
vncSockConnect	เป็นคลาสที่ทำหน้าที่สร้าง socket ขึ้นมารอรับการเชื่อมโยงเพื่อติดต่อด้วย address และ port ที่ระบุมา โดยเมื่อวัตถุถูกทำลายจะทำให้ทรัพยากรหายไป รวมทั้ง thread ที่รอรับการเชื่อมโยงนี้ด้วย
VSocket	เป็นคลาสที่ทำหน้าที่สร้าง socket โดยไม่ขึ้นอยู่กับ platform ใด

4.2.2 คลาสที่อยู่ที่ฝั่งไคล์เอ็นต์ที่สำคัญ

CMainFrame	เป็นคลาสที่ถ่ายทอดลักษณะมาจากคลาส CMDIFrameWnd เป็นวินโดว์หลักของโปรแกรม โดยจะแสดงแถบเครื่องมือ และเส้นกั้นวินโดว์เป็น 2 ส่วน
CScreenWatchApp	เป็นคลาสที่ถ่ายทอดลักษณะมาจากคลาส CWinApp เป็นโครงของโปรแกรมประยุค
CScreenWatchView	เป็นคลาสที่ถ่ายทอดลักษณะมาจากคลาส CScrollView เป็นส่วนที่แสดงจอภาพที่ต้องการสังเกต โดยมีแถบสคอลเลื่อนได้
CNetworkTree	เป็นคลาสที่ถ่ายทอดลักษณะมาจากคลาส CTreeView โดยจะแสดงชื่อเครื่องคอมพิวเตอร์ที่อยู่ในระบบเครือข่ายท้องถิ่นเดียวกันเป็นลักษณะต้นไม้

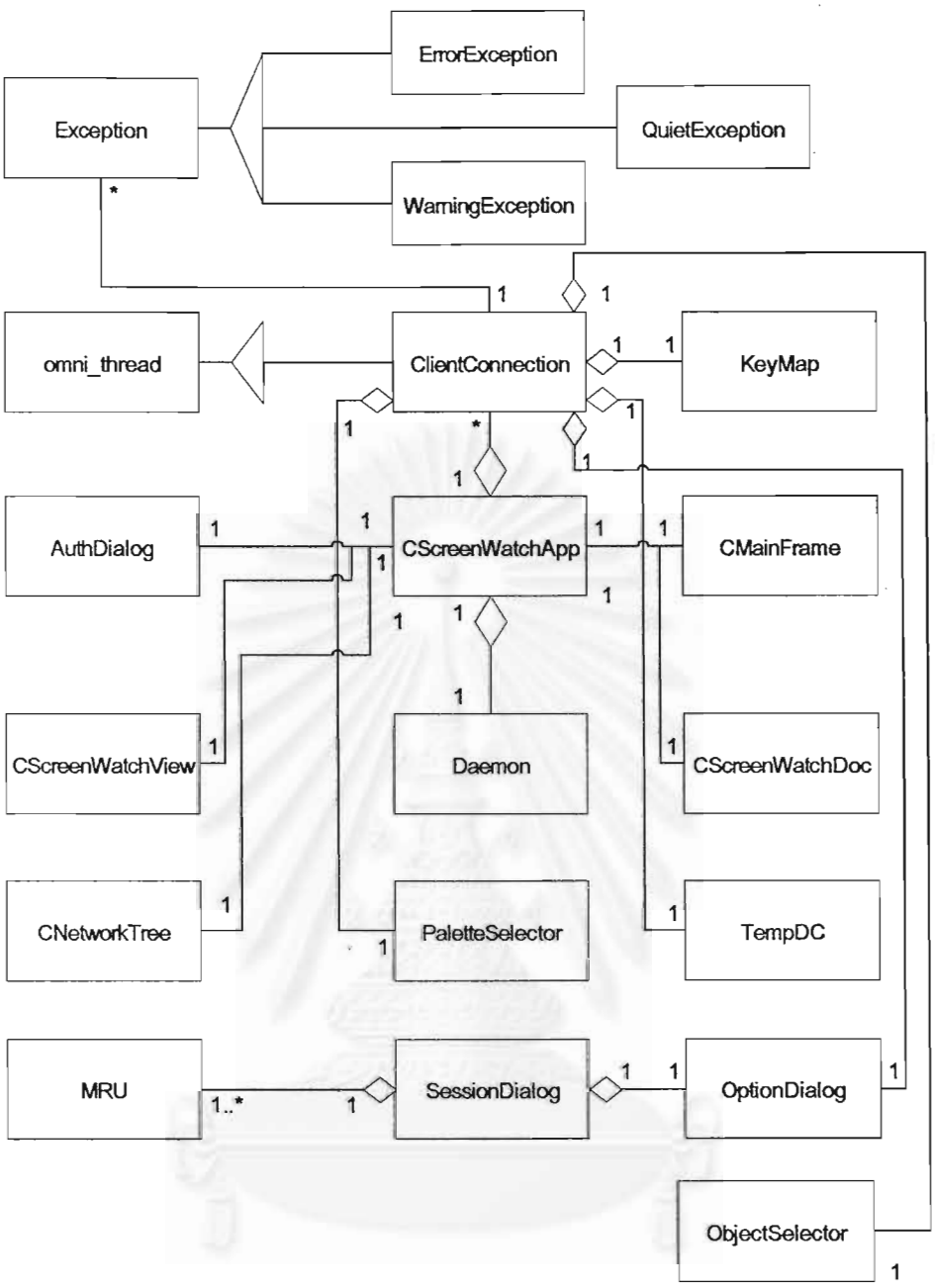
ClientConnection	เป็นคลาสที่ควบคุมและดูแลทุกอย่างที่ต้องติดต่อกับเซิร์ฟเวอร์ ประกอบด้วย 2 threads โดยตัวหลักจะทำการติดตั้งการเชื่อมเส้นทาง และจัดการกับแมสเสจของวินโดว์ ส่วนอีกตัวหนึ่งจะทำการรับและถอดรหัสข้อมูลที่ได้ แล้ววาดรูปเป็นผลลัพธ์
Exception	เป็นคลาสที่ดูแลสิ่งผิดพลาดที่เกิดขึ้นในโปรแกรม
ErrorException	เป็นคลาสที่ถ่ายทอดลักษณะมาจากคลาส Exception ตรวจสอบถ้าเกิดสิ่งผิดพลาดที่ร้ายแรงจะมีข้อความบอก และปิดการติดต่อ
QuietException	เป็นคลาสที่ถ่ายทอดลักษณะมาจากคลาส Exception ตรวจสอบถ้าเกิดสิ่งผิดพลาดที่ต้องปิดการติดต่อ
WarningException	เป็นคลาสที่ถ่ายทอดลักษณะมาจากคลาส Exception ตรวจสอบถ้าเกิดสิ่งผิดพลาดที่ไม่ร้ายแรงจะมีข้อความเตือน
MRU	เป็นคลาสที่เก็บรายชื่อ “Most Recently Used” ที่อยู่ใน registry ของวินโดว์
ObjectSelector	เป็นคลาสที่ทำการเลือกวัตถุที่จะเป็นเครื่องมือสำหรับวาดภาพซึ่งวินโดว์มีให้ใช้
PaletteSelector	เป็นคลาสที่ทำการเลือกวัตถุที่จะเป็นตารางสีสำหรับวาดภาพ

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.1 แสดงผังของคลาสที่ฝั่งเซิร์ฟเวอร์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.2 แสดงผังของคลาสที่ฝั่งไคล์เอ็นต์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5
ผลการทดสอบโปรแกรม

ในการทดสอบโปรแกรมของ ระบบเฝ้าสังเกตจอภาพของผู้เรียนผ่านระบบเครือข่ายห้องเรียนคอมพิวเตอร์ ได้ทำการทดสอบในสภาพแวดล้อมของเครือข่ายที่มีรูปแบบการเชื่อมต่อ (Topologies) แบบดาว(Star) ของเครื่องคอมพิวเตอร์ในห้องเรียน การทดสอบได้ทดสอบกับเครื่องคอมพิวเตอร์ผู้เรียน 27 เครื่องที่มีการใช้งานอยู่ ใช้เครื่องคอมพิวเตอร์เฝ้าสังเกตเครื่องคอมพิวเตอร์ผู้เรียน 1 เครื่อง

ระบบควบคุมสำหรับการทดสอบคือ สภาพแวดล้อมของการใช้งานเครื่องคอมพิวเตอร์ผู้เรียนในระบบเครือข่ายห้องเรียนคอมพิวเตอร์ โดยได้ทดสอบกับคอมพิวเตอร์ผู้เรียน 27 เครื่องที่มีการใช้งานอยู่ และใช้ประสิทธิภาพของระบบเครือข่าย(Network Utilization) แบบไม่สม่ำเสมอ แต่โดยเฉลี่ยใช้ประสิทธิภาพของระบบเครือข่ายประมาณ 15.0 %

ระบบทดลองสำหรับการทดสอบนี้จะแบ่งเป็น 2 การทดสอบ คือ การทดสอบโปรแกรมส่วนที่เฝ้าสังเกตจอภาพของผู้เรียนผ่านระบบเครือข่าย และการทดสอบโปรแกรมส่วนที่แสดงจอภาพของผู้สอนให้ผู้เรียนผ่านระบบเครือข่าย ในที่นี้เครื่องคอมพิวเตอร์ผู้สอนจะเป็นเครื่องคอมพิวเตอร์ที่ทำหน้าที่เฝ้าสังเกตจอภาพของเครื่องคอมพิวเตอร์ผู้เรียน ขณะที่เครื่องคอมพิวเตอร์ผู้เรียนจะเป็นเครื่องคอมพิวเตอร์ที่ถูกเฝ้าสังเกตจอภาพ

การทดสอบโปรแกรมส่วนที่เฝ้าสังเกตจอภาพของผู้เรียนผ่านระบบเครือข่าย จะใช้เครื่องคอมพิวเตอร์ที่ทำหน้าที่ เฝ้าสังเกตจอภาพของเครื่องคอมพิวเตอร์ผู้เรียน 1 เครื่อง และเครื่องคอมพิวเตอร์ที่ถูกเฝ้าสังเกตจอภาพ จะมีจำนวนเพิ่มขึ้น 1 เครื่องต่อการทดลอง 1 ครั้ง นั่นคือ การทดลองครั้งที่ 1 จะมี 1 เครื่องของผู้ถูกเฝ้าสังเกต การทดลองครั้งที่ 2 จะมี 2 เครื่องของผู้ถูกเฝ้าสังเกต จนการทดลองครั้งที่ 27 จะมี 27 เครื่องของผู้ถูกเฝ้าสังเกต ขณะที่ทดสอบโปรแกรมพบว่าการใช้ประสิทธิภาพของระบบเครือข่ายไม่สม่ำเสมอ แต่โดยเฉลี่ยใช้ประสิทธิภาพของระบบเครือข่ายประมาณ 15.0 % เช่นเดียวกับระบบควบคุม โดยการทดสอบจะเริ่มจากการติดต่อเครื่องคอมพิวเตอร์ที่เฝ้าสังเกตกับ เครื่องคอมพิวเตอร์ที่ถูกเฝ้าสังเกตจอภาพก่อนเสมอ จากนั้น จะทำการส่งข้อมูลเพื่อใช้ในการทดสอบเวลาที่ใช้ในการรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ โดยใช้ข้อความ Bell ของโปรโตคอลอาร์เอฟบี[2] ซึ่งมีความยาว 8 ไบต์ โดยไม่นับรวมกับข้อมูลที่จะยาวเพิ่มขึ้นจากการใช้โปรโตคอลทีซีพี/ไอพี ข้อความที่ใช้ทดสอบความเร็วในการรับส่งของโปรแกรมจะถูกส่งจากเครื่องคอมพิวเตอร์ที่เฝ้าสังเกตจอภาพ หลังจากคำสั่งในการส่งข้อความสิ้นสุดลงจะเริ่มนับจับเวลา ที่ข้อความถูกส่งไปยังเครื่องคอมพิวเตอร์ที่ถูกเฝ้าสังเกตจอภาพ เมื่อเครื่องคอมพิวเตอร์ที่ถูกเฝ้าสังเกตจอภาพได้รับข้อความ Bell แล้วจะส่งข้อความตอบกลับมาที่เครื่อง

คอมพิวเตอร์ที่เฝ้าสังเกตจอภาพด้วยข้อความเดียวกัน จึงหยุดการนับจับเวลาพร้อมกับบันทึกเวลาที่ใช้รับส่งข้อมูลไว้ ทำการทดลองซ้ำโดยเพิ่มจำนวนเครื่องคอมพิวเตอร์ผู้เรียน แล้วนับจับเวลาที่ใช้ในการรับส่งข้อมูลระหว่างเครื่องผู้เรียนที่เพิ่มเข้ามาใหม่ โดยใช้วิธีทดลองเหมือนเดิมจนครบ 27 เครื่อง

การทดสอบโปรแกรมส่วนที่แสดงจอภาพของผู้สอนให้ผู้เรียนผ่านระบบเครือข่าย จะใช้เครื่องคอมพิวเตอร์ที่ทำหน้าที่ แสดงจอภาพของผู้สอนให้ผู้เรียน 1 เครื่อง และเครื่องคอมพิวเตอร์ที่รอรับการแสดงจอภาพจากผู้สอน จะมีจำนวนเพิ่มขึ้น 1 เครื่องต่อการทดลอง 1 ครั้ง ขณะที่ทดสอบโปรแกรม พบว่าการใช้ประสิทธิภาพของระบบเครือข่ายไม่สม่ำเสมอ แต่โดยเฉลี่ยใช้ประสิทธิภาพของระบบเครือข่ายประมาณ 15.0 % โดยการทดสอบจะทดลองส่งทั้งจอภาพของผู้สอนให้ผู้เรียน จะเริ่มจากการติดต่อระหว่างเครื่องคอมพิวเตอร์ที่แสดงจอภาพของผู้สอน กับเครื่องคอมพิวเตอร์ที่รับการแสดงจอภาพจากผู้สอน ทันทีที่การติดต่อระหว่างเครื่องรับส่งเกิดขึ้นจะมีการส่งสัญญาณจอภาพมาแสดงที่เครื่องคอมพิวเตอร์ที่รับการแสดงจอภาพจากผู้สอน การทดสอบเวลาที่ใช้ในการรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ จะถูกนับจับเวลาทันทีหลังจากที่มีการติดต่อระหว่างคู่สัญญาณ และหยุดเมื่อการส่งข้อความจากเครื่องคอมพิวเตอร์ที่แสดงจอภาพ ถูกส่งไปให้เครื่องคอมพิวเตอร์ที่รับการแสดงจอภาพจากผู้สอนแล้วเสร็จ พร้อมกับบันทึกเวลาที่ใช้รับส่งข้อมูลไว้ ทำการทดลองซ้ำโดยเพิ่มจำนวนเครื่องคอมพิวเตอร์ผู้เรียน แล้วนับจับเวลาแบบสะสมที่ใช้ในการรับส่งข้อมูลระหว่างเครื่องผู้เรียนที่เพิ่มเข้ามาใหม่ โดยใช้วิธีทดลองเหมือนเดิมจนครบ 27 เครื่อง

ผู้วิจัยได้ทำการทดสอบโปรแกรมที่พัฒนาขึ้นบนระบบปฏิบัติการวินโดว์ เอ็น ที สำหรับเซิร์ฟเวอร์รุ่น 4.0 (Windows NT Server 4.0) สำหรับโปรแกรมฝั่งที่เป็นเครื่องที่เฝ้าสังเกตจอภาพ ซึ่งใช้หน่วยความจำหลัก 128 เมกะไบต์ หน่วยประมวลผลกลาง รุ่นเพียนเทียม ทู ความเร็ว หน่วยประมวลผลกลาง 266 เมกะเฮิร์ตซ์ ต่ออยู่กับเครือข่ายท้องถิ่นในห้องเรียนคอมพิวเตอร์ที่เป็นกลุ่มร่วมงาน(workgroup) เดียวกัน ซึ่งใช้มาตรฐานของระบบเครือข่าย 10/100 เมกะบิตต่อวินาที (10/100 Mb/s Ethernet Network) โปรโตคอลหลักของเครือข่ายที่ใช้เป็นไอ พี (IP Protocol) และ ที ซี พี (TCP Protocol) และใช้งานกับโปรแกรมที่ฝั่งผู้สอนบนระบบปฏิบัติการวินโดว์ เอ็น ที สำหรับสถานีงาน รุ่น 4.0 (Windows NT Workstation 4.0) และระบบปฏิบัติการวินโดว์ 95 /98 (Windows 95 /98) ซึ่งทุกเครื่องใช้หน่วยความจำหลัก 64 เมกะไบต์ หน่วยประมวลผลกลางรุ่นเพียนเทียมความเร็ว หน่วยประมวลผลกลาง 200 เมกะเฮิร์ตซ์

ผู้วิจัย ได้ทำการทดสอบโปรแกรมที่พัฒนาขึ้น 27 จอภาพโดยเริ่มเพิ่มจำนวนเครื่อง ที่ต้องการเฝ้าสังเกตทีละเครื่อง ผลการทดสอบโปรแกรมสามารถเฝ้าสังเกตจอภาพได้ตาม วัตถุประสงค์ ตามรูป (รูปที่ 5.1) สำหรับเครื่องคอมพิวเตอร์ฝั่งเครื่องคอมพิวเตอร์ผู้เรียน ใช้หน่วย

ประมวลผลกลางในการทำงานเฉลี่ยประมาณ 20.0 % ใช้หน่วยความจำหลักเฉลี่ยประมาณ 4 เมกะไบต์ สำหรับเครื่องคอมพิวเตอร์ฝั่งผู้สอนใช้หน่วยประมวลผลกลางในการทำงานเฉลี่ยประมาณ 15.0 % ใช้หน่วยความจำหลักเฉลี่ยประมาณ 3 เมกะไบต์

5.1 การทดสอบโปรแกรมส่วนที่เฝ้าสังเกตจอภาพของผู้เรียน

สำหรับประสิทธิภาพของเครือข่าย(network utilization) ขณะโปรแกรมใช้งาน ผู้วิจัยได้ใช้โปรแกรมตรวจสอบระบบเครือข่ายวัดปรากฏว่า เฉพาะโปรแกรมที่พัฒนาขึ้นใช้ประสิทธิภาพของเครือข่ายเฉลี่ยประมาณ 15.0 % เวลาที่ใช้ในการส่งข้อมูล 1 กลุ่มข้อมูล (packet) โดยใช้ความยาวข้อมูล 8 ไบต์ในการทดสอบจากเครื่องที่เฝ้าสังเกตจอภาพของผู้เรียนไปยังเครื่องที่ถูกเฝ้าสังเกตจอภาพแล้วเครื่องที่ถูกเฝ้าสังเกตจอภาพตอบกลับ(round trip time) โดยใช้วิธีเขียนโปรแกรมใส่จุดทดสอบในรหัสต้นฉบับ (source code) จับเวลาเฉลี่ยใช้เวลา 17,397.57 มิลลิวินาที

ตารางที่ 5.1 แสดงผลการทดสอบโปรแกรมส่วนที่เฝ้าสังเกตจอภาพของผู้เรียนโดยround trip time

จำนวนเครื่องผู้เรียน	เวลาที่ใช้ในการรับส่งข้อมูลระหว่างเครื่องผู้สอนกับเครื่องผู้เรียน (ms)
1	27,069
2	13,449
3	18,997
4	14,210
5	27,069
6	16,975
7	20,359
8	15,131
9	18,166
10	15,673
11	15,893
12	15,893
13	16,164
14	16,223
15	16,414

จำนวนเครื่องผู้เรียน	เวลาที่ใช้ในการรับส่งข้อมูลระหว่างเครื่องผู้สอนกับเครื่องผู้เรียน (ms)
16	13,219
17	16,513
18	16,574
19	16,503
20	17,715
21	14,471
22	14,331
23	19,328
24	13,840
25	20,490
26	26,669
27	13,219

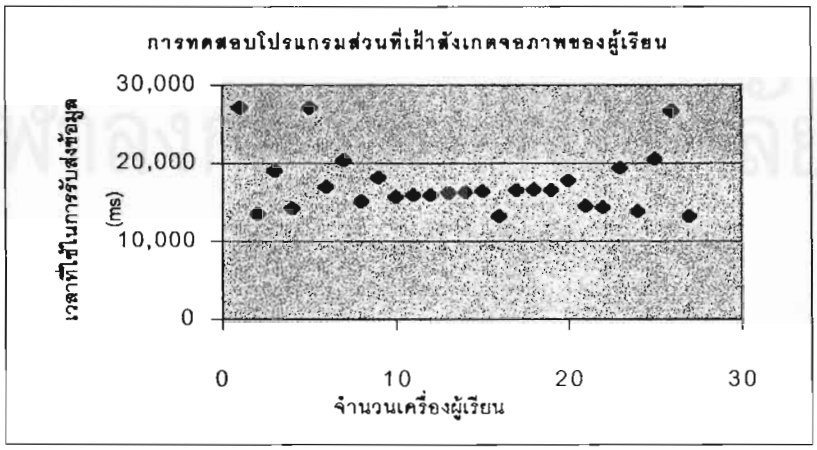
5.2 การทดสอบโปรแกรมส่วนที่แสดงจอภาพของผู้สอนให้ผู้เรียน

สำหรับประสิทธิภาพของเครือข่าย(network utilization) ขณะโปรแกรมใช้งาน ผู้วิจัยได้ใช้โปรแกรมตรวจสอบระบบเครือข่ายวัดปรากฏว่า เฉพาะโปรแกรมที่พัฒนาขึ้นใช้ประสิทธิภาพของเครือข่ายเฉลี่ยประมาณ 15.0 % เหมือนกับการทดสอบโปรแกรมส่วนที่เฝ้าสังเกตจอภาพของผู้เรียน เวลาที่ใช้ในการส่งข้อมูลเฉลี่ยจาก เครื่องผู้สอนไปยังที่เครื่องผู้เรียนโดยใช้วิธีเขียนโปรแกรมใส่จุดทดสอบในรหัสต้นฉบับ (source code) จับเวลา พบว่าใช้เวลา 693.53 มิลลิวินาที ต่อ 1 เครื่องของผู้เรียน

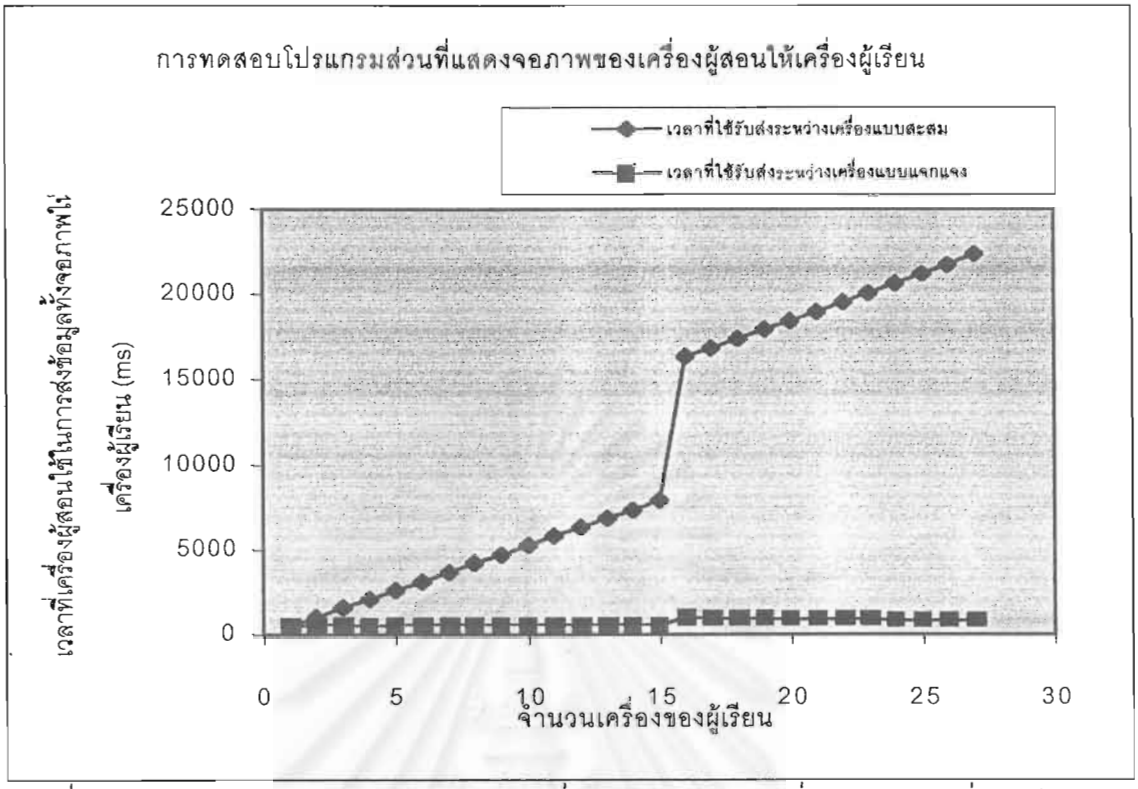
ตารางที่ 5.2 แสดงการทดสอบโปรแกรมส่วนที่แสดงจอภาพของผู้สอนให้เครื่องผู้เรียน

จำนวนเครื่องผู้เรียน	เวลาที่เครื่องผู้สอนใช้ในการส่งข้อมูลทั้งจอภาพให้เครื่องผู้เรียน (ms)
1	501
2	1,042
3	1,582
4	2,053
5	2,634
6	3,145

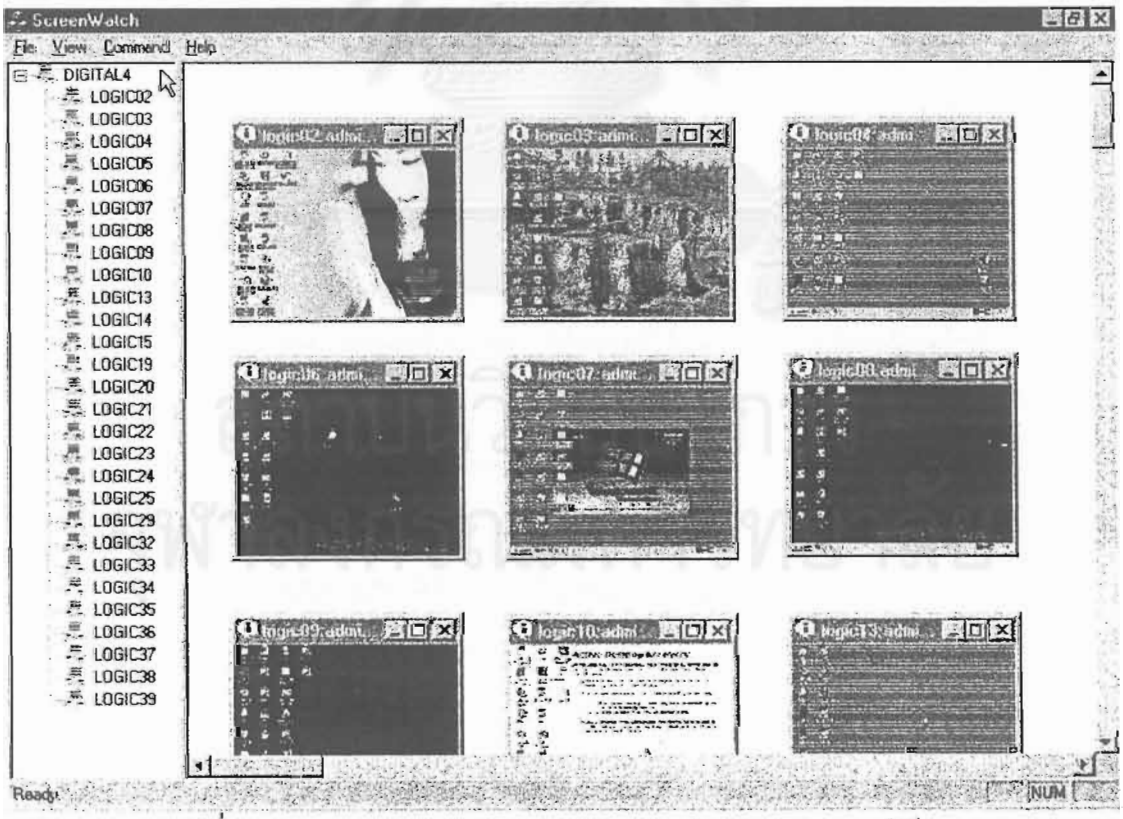
จำนวนเครื่องผู้เรียน	เวลาที่เครื่องผู้สอนใช้ในการส่งข้อมูลทั้งจอภาพให้เครื่องผู้เรียน (ms)
7	3,675
8	4,203
9	4,729
10	5,262
11	5,784
12	6,312
13	6,844
14	7,374
15	7,904
16	16,338
17	16,862
18	17,384
19	17,902
20	18,440
21	18,986
22	19,518
23	20,064
24	20,622
25	21,190
26	21,764
27	22,340



รูปที่ 5.1 แสดงการทดสอบโปรแกรมส่วนที่เฝ้าสังเกตจอภาพของผู้เรียน



รูปที่ 5.2 แสดงกราฟการทดสอบโปรแกรมส่วนที่แสดงจอภาพของเครื่องผู้สอนให้เครื่องผู้เรียน



รูปที่ 5.3 แสดงการทำงานของโปรแกรมขณะเฝ้าสังเกตจอภาพผู้เรียน

บทที่ 6
สรุปและข้อเสนอแนะ

ปัจจุบันห้องปฏิบัติการคอมพิวเตอร์เพื่อการเรียนการสอนที่อยู่ในสภาพแวดล้อมเครือข่ายท้องถิ่นมีการใช้งานกันมาก แต่การติดตามผลการเรียนการสอนในห้องเรียนของผู้เรียนยังคงเป็นวิธีดั้งเดิมคือ การที่ผู้สอนต้องคอยติดตามผลการเรียนการสอนในห้องเรียนด้วยตนเอง เนื่องจากสิ่งที่เรามีอยู่คือทรัพยากร เครื่องคอมพิวเตอร์ที่อยู่ในระบบเครือข่ายท้องถิ่น ซึ่งพอที่จะนำมาประยุกต์ใช้กับโปรแกรมเฝ้าสังเกตจอภาพของผู้เรียนได้ ดังนั้นการพัฒนาโปรแกรมนี้จึงเป็นแนวทางที่สามารถจะช่วยแก้ปัญหานี้ได้

6.1 สรุป

โปรแกรมที่พัฒนาขึ้นใช้ไมโครซอฟท์วิชวล C++ เป็นเครื่องมือ โดยวิธีการโปรแกรมเชิงวัตถุ ใช้คลาสพื้นฐานของไมโครซอฟท์และคลาสที่สร้างขึ้นเอง

ในการพัฒนาโปรแกรมแบ่งโปรแกรมเป็นส่วนๆ ในที่นี้ ให้เครื่องคอมพิวเตอร์ของผู้เรียนเป็นเซิร์ฟเวอร์ และเครื่องคอมพิวเตอร์ของผู้สอนเป็นไคล์เอ็นต์ แบ่งการพัฒนาส่วนต่างๆของโปรแกรม ได้ดังนี้คือ

6.1.1 ส่วนโปรแกรมที่ทำหน้าที่เป็นเซิร์ฟเวอร์

- ส่วนการติดต่อกับไคล์เอ็นต์เพื่อเตรียมการรับส่งข้อมูล
- ส่วนการเตรียมรูปแบบข้อมูลที่จะใช้ในการติดต่อกับไคล์เอ็นต์
- ส่วนการเตรียมส่งข้อมูลหน้าจอของเซิร์ฟเวอร์ให้ไคล์เอ็นต์จากบัฟเฟอร์
- ส่วนการส่งข้อมูลหน้าจอของเซิร์ฟเวอร์ให้ไคล์เอ็นต์ พร้อมทั้งรอรับข้อความจากไคล์เอ็นต์ แล้วตอบสนองตามข้อความนั้น

6.1.2 ส่วนโปรแกรมที่ทำหน้าที่เป็นไคล์เอ็นต์

- ส่วนส่วนที่ต้องแแสดงส่วนต่อประสานกราฟิกกับผู้ใช้ (graphic user interface)
- ส่วนการติดต่อกับเซิร์ฟเวอร์เพื่อเตรียมการรับส่งข้อมูล
- ส่วนการเตรียมรูปแบบข้อมูลที่จะใช้ในการติดต่อกับเซิร์ฟเวอร์
- ส่วนการส่งข้อความ และรับข้อมูลหน้าจอจากเซิร์ฟเวอร์
- ส่วนการแสดงหน้าจอเซิร์ฟเวอร์

จากการทดสอบโปรแกรมที่พัฒนาขึ้นสามารถทำงานได้กับ เครื่องคอมพิวเตอร์ผู้เรียน จำนวน 27 เครื่องในห้องเรียนคอมพิวเตอร์ โดยการทดสอบแบ่งเป็น 2 กรณี กรณีทดสอบการเฝ้าสังเกตจอภาพของผู้เรียนหลายเครื่องพร้อมกัน ใช้ทรัพยากรและเวลาในการรับส่งข้อมูลระหว่างเครื่องคอมพิวตอร์คงที่ ใช้เวลาในการรับส่งข้อมูลเฉลี่ย 17,397.57 มิลลิวินาที เนื่องจากการทำงานมีลักษณะแบ่งหน้าที่กันส่งจอภาพ ของเครื่องคอมพิวตอร์ผู้เรียนทุกเครื่องเท่ากัน สำหรับกรณีการทดสอบการแสดงหน้าจอผู้สอนให้ เครื่องคอมพิวตอร์ผู้เรียนทุกเครื่องจะใช้ทรัพยากร(ใช้หน่วยความจำหลักในการทำที่พักข้อมูลและการใช้งานหน่วยประมวลผลกลาง) และเวลาในการรับส่งข้อมูลระหว่างเครื่องคอมพิวตอร์มากขึ้นตามจำนวนเครื่องที่ติดต่อด้วย เนื่องจากการออกแบบการทำงาน เป็นลักษณะที่เครื่องคอมพิวตอร์ผู้สอนเป็นเครื่องประสานเวลาการรับส่งข้อมูลจอภาพเพียงเครื่องเดียว เพื่อทำให้เครื่องคอมพิวตอร์ผู้เรียนทุกเครื่องใช้ทรัพยากรของตัวเองน้อยที่สุดและสามารถทำงานอื่นได้ปกติ ดังนั้นเครื่องคอมพิวตอร์ผู้สอนจึงใช้ทรัพยากรของตัวเองมากกว่าเครื่องคอมพิวตอร์ผู้เรียน คือใช้หน่วยความจำหลักในการทำที่พักข้อมูลต่อ 1 เครื่องคอมพิวตอร์ผู้เรียนไปประมาณ 4 เมกะไบต์

6.2 ข้อเสนอแนะ

โปรแกรมที่พัฒนาขึ้นเป็นแนวทางหนึ่งซึ่งผู้ที่ต้องการพัฒนาโปรแกรมในลักษณะใกล้เคียงกันสามารถนำไปใช้เป็นแบบอย่างได้ หรือพัฒนาส่วนประกอบต่างๆ ของโปรแกรมในแนวทางอื่นๆ ได้ เช่น ควบคุมเครื่องคอมพิวเตอร์ผู้เรียนไม่ให้ใช้งานขณะสอน การทำโปรแกรมโต้ตอบกับขณะสอน การส่งสัญญาณเสียงไปยังเครื่องคอมพิวเตอร์ผู้เรียน เป็นต้น หากแต่การพัฒนานี้ยังสามารถทำได้ดีขึ้น ถ้าสามารถแก้ไขขั้นตอนวิธีในการส่งข้อมูลการแสดงหน้าจอผู้สอนให้เครื่องคอมพิวตอร์ผู้เรียนให้สามารถใช้ทรัพยากรน้อยลง เนื่องจากวิธีการพัฒนานี้ทำให้ต้องใช้หน่วยความจำหลักมาก ดูได้จากผลการทดลองในรูปที่ 5.2 ซึ่งในการพัฒนายังไม่ได้ทำให้การเข้าถึงข้อมูลจอภาพเป็นในลักษณะของการใช้งานบัฟเฟอร์ของข้อมูลจอภาพร่วมกัน(shared memory) ซึ่งถ้าสามารถทำได้จะทำให้ลดเวลา ในการประมวลผลสำหรับแต่ละเครื่องคอมพิวเตอร์ผู้เรียนที่มีจำนวนเพิ่มขึ้นได้ แต่สำหรับทางแก้ไขอีกวิธีหนึ่งของปัญหาที่เกิดขึ้น ซึ่งไม่ดีนักคือ เพิ่มหน่วยความจำหลักให้กับเครื่องคอมพิวเตอร์ของผู้สอน

รายการอ้างอิง

[1] David, K. ; and Simon , S. Lam.Migration Socket. IEEE/ACM Transaction On Network VOL 6,NO.6 (December 1998): 700-716.

[2] Tristan, R. I.; Quentin, S. T.; Kenneth, R. W.; and Andy, H. W. Virtual Network Computing. IEEE Internet Computing VOL. 2, NO.1 (January/Febuary 1998): 32-38.

[3] Andrew, S.; and Tanenbaum. Computer Networks. Prentice-Hall,Inc.,1996.

[4] Cypser, R.J. Communications for Cooperating Systems OSI, SNA and TCP/IP. Addison-Wesley Publishing Company,Inc.,1992.

[5] Morris, S. Network and Distributed Systems Management. Addison-Wesley Publishing Company,Inc.,1994.

[6] Fred, H. Data Communications Computer Networks and Open Systems. Addison-Wesley Publishing Company,Inc.,1994.

[7] สื่อสารข้อมูลโดยใช้ชุดโปรโตคอล TCP/IP. ไมโครคอมพิวเตอร์ 113 (2537).

[8] ไคล์เอ็นต์ เซอร์ฟเวอร์ เทคโนโลยีการใช้คอมพิวเตอร์ในองค์กร. ไมโครคอมพิวเตอร์ 99 (2536).

[9] Craig, H. TCP/IP Network Administration .O'Reilly & Associates, Inc. ,1993.

[10] Phillip, A. L.; Alexander, D.; and Stoyenko.Real-time Imaging,IEEE PRESS,1996.

[11] Theo, A. A PC-Base Real-time Multimedia Tele-Education System,IEEE,1994.

[12] Prabhat, K.; Andleigh; and Kiran, T. Multimedia System Design. Prentice Hall, 1995.

[13] Stuart, B. Data Transfer Techniques.Real-time Computer Control. Prentice Hall, 1994.

[14] Thierry, T.; and Christian, H. Videoconferencing on the Internet.IEEE/ACM Transactions on Networking VOL.4, NO.3 (June 1996): 340-351.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

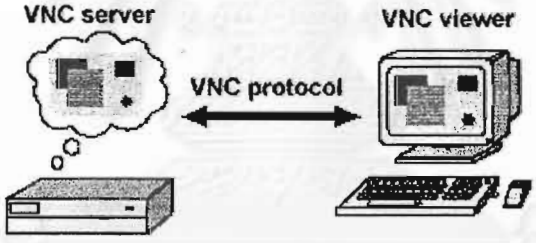
ภาคผนวก ก

โปรโตคอลอาร์ เอฟ บี (RFB Protocol)

1.บทนำ

RFB ("remote framebuffer") คือ โปรโตคอลแบบง่ายใช้สำหรับการเข้าถึงระยะไกล (remote access) ไปยังส่วนต่อประสานกราฟิกกับผู้ใช้ (graphical user interface) เนื่องจากการทำงานของมันอยู่ในระดับเฟรมบัฟเฟอร์ (framebuffer level) จึงสามารถใช้งานได้กับระบบวินโดวส์และแอพพลิเคชันจำนวนมาก ได้แก่ X11, Window3.1/95/98/NT และ Macintosh

ที่จุดปลายทางระยะไกล (remote endpoint) ซึ่งผู้ใช้นั่งอยู่ (หมายถึง หน่วยแสดงผลรวมถึงแผงแป้นอักขระและพอยเตอร์) จะเรียกว่า อาร์ เอฟ บี ไคล์เอ็นต์ (RFB client) ที่จุดปลายทางซึ่งจะเปลี่ยนแปลงเฟรมบัฟเฟอร์ต้นฉบับ (หมายถึงการใช้งานระบบวินโดวส์ และแอพพลิเคชันต่างๆ) จะเรียกว่า อาร์ เอฟ บี เซิร์ฟเวอร์ (RFB server)



รูปที่ ก.1 แสดงการทำงานของโปรโตคอลโดยภาพกว้าง

RFB เป็นโปรโตคอลที่เป็น thin client คือ ส่วนที่เน้นสำหรับการออกแบบของโปรโตคอล RFB คือ เพื่อให้ความต้องการของตัวไคล์เอ็นต์ไม่ต้องมีมาก ด้วยเหตุนี้เอง ไคล์เอ็นต์จึงสามารถทำงานบนฮาร์ดแวร์หลายชนิด และยังทำให้การสร้างให้เกิดไคล์เอ็นต์ง่ายด้วย

โปรโตคอลนี้ยังทำให้ไคล์เอ็นต์ไม่ต้องมีสถานะด้วย ถ้าไคล์เอ็นต์เลิกเชื่อมต่อจากเซิร์ฟเวอร์ใดๆ เซิร์ฟเวอร์หนึ่ง แล้วเชื่อมต่อเซิร์ฟเวอร์เดิมอีกครั้งหนึ่ง สถานะของส่วนต่อประสานกับผู้ใช้จะยังสงวนอยู่ นอกจากนี้ที่จุดปลายทางไคล์เอ็นต์ที่ต่างกัน สามารถจะเชื่อมต่อมาที่ อาร์ เอฟ บี เซิร์ฟเวอร์เดียวกันได้ ที่จุดปลายทางอันใหม่ก็จะสามารถเห็นส่วนต่อประสานกราฟิกกับผู้ใช้เดียวกันกับต้นฉบับได้ ผลที่ได้ก็คือการต่อประสานไปยังแอพพลิเคชันของผู้ใช้ ก็เชื่อมโยงเกี่ยวไปด้วย เมื่อมีการเชื่อมต่อ

แล้ว ผู้ใช้สามารถจะเข้าถึงแอพพลิเคชันของตัวเอง ขณะที่การเข้าถึงระยะไกล ก็จะสงวนสถานะของแอพพลิเคชันเหล่านี้ไว้ด้วยเช่นกัน ทำให้ผู้ใช้รู้สึกคุ้นเคยเหมือนกับที่ตัวเองใช้เองจริง

2. โปรโตคอลของการแสดงผล (Display Protocol)

ในส่วนการแสดงผลของโปรโตคอลจะอยู่บนพื้นฐานอย่างเดียวของกราฟิกเบื้องต้นที่ว่า "วางสี่เหลี่ยมผืนผ้าของข้อมูลจุดภาพลงในตำแหน่ง x, y ตามที่กำหนด" ในขั้นต้นอาจจะดูเหมือนว่าไม่เป็นวิธีที่มีประสิทธิภาพที่จะวาดหลายส่วนประกอบของส่วนต่อประสานกับผู้ใช้ อย่างไรก็ดี เราจะมีวิธีการที่จะเข้ารหัสข้อมูลจุดภาพหลายๆวิธี เพื่อความยืดหยุ่นในการแบ่งรับแบ่งสู้กันระหว่างพารามิเตอร์ (parameters) ต่างๆ ได้แก่ network bandwidth, ความเร็วในการวาดที่ไคล์เอ็นต์ และความเร็วของเซิร์ฟเวอร์ในการประมวลผล

ลำดับของสี่เหลี่ยมผืนผ้า จะทำให้เกิดการปรับให้เป็นปัจจุบันเฟรมบัฟเฟอร์ (framebuffer update) การแทนการปรับให้เป็นปัจจุบัน จะเปลี่ยนจากสถานะเฟรมบัฟเฟอร์หนึ่งไปยังสถานะอื่น เหมือนกับเฟรมในวีดีโอ สี่เหลี่ยมผืนผ้าที่ปรับให้เป็นปัจจุบันปกติจะไม่เป็นผืนเดียวกัน แต่ก็ไม่จำเป็นเสมอไป

โปรโตคอลการปรับให้เป็นปัจจุบันเป็นไปในลักษณะเกิดขึ้นมากน้อยขึ้นอยู่กับความต้องการของไคล์เอ็นต์ นั่นคือ การปรับให้เป็นปัจจุบันจะถูกส่งจากเซิร์ฟเวอร์ไปไคล์เอ็นต์ เมื่อมีการร้องขอจากไคล์เอ็นต์เท่านั้น ถ้าไคล์เอ็นต์หรือระบบเครือข่ายช้า อัตราการปรับให้เป็นปัจจุบันจะช้าไปด้วย ปกติในแอฟพลิเคชันทั่วไป การเปลี่ยนแปลงในบางพื้นที่หนึ่งเฟรมบัฟเฟอร์ ณ ตำแหน่งเดียวกันจะเปลี่ยนแปลงไปด้วยในเวลาสั้นๆต่อมา ด้วยไคล์เอ็นต์หรือระบบเครือข่ายที่ช้า สถานะของเฟรมบัฟเฟอร์จะกำลังเปลี่ยนแปลงอาจถูกละเลยไป ผลก็คือทำให้การส่งข้อมูลในระบบเครือข่ายเกิดแออัด และทำให้การวาดที่ไคล์เอ็นต์ช้าไปได้

3. โปรโตคอลของส่วนนำเข้า (Input Protocol)

ในส่วนของการนำเข้าของโปรโตคอลจะอยู่บนพื้นฐานของรูปแบบมาตรฐานสถานีงาน (standard workstation model) ของแผงแป้นอักขระและอุปกรณ์ตัวชี้แบบหลายปุ่ม เหตุการณ์การนำเข้า(Input events) จะถูกส่งไปเซิร์ฟเวอร์โดยไคล์เอ็นต์ เมื่อไรก็ตามที่มีการกดแป้นอักขระหรือปุ่มตัวชี้ หรือเมื่อไรก็ตามที่อุปกรณ์ตัวชี้ถูกเคลื่อนย้าย เหตุการณ์นำเข้ายังถูกสังเคราะห์ขึ้นจากอุปกรณ์

ไอโอ (I/O device) อื่นๆได้อีก ตัวอย่างเช่น pen-base handwriting อาจจะสังเคราะห์เหตุการณ์นำเข้าแบบแผงแป้นอักขระก็ได้

4. การแทนข้อมูลจุดภาพ (Representation of pixel data)

เมื่อเริ่มการติดต่อระหว่าง RFB client และ server จะมีการส่ง format และ encoding พร้อมกับข้อมูลจุดภาพไปด้วย การติดต่อกันนี้ถูกออกแบบเพื่อให้การทำงานของฝั่งไคล์เอ็นต์เป็นไปโดยง่าย สิ่งที่สำคัญ ก็คือ เซิร์ฟเวอร์ต้องสามารถผลิตและส่งข้อมูลจุดภาพในรูปแบบที่ไคล์เอ็นต์ต้องการได้อย่างสม่ำเสมอ อย่างไรก็ดีไคล์เอ็นต์สามารถที่จะเลือกรูปแบบ (formats) หรือ การเข้ารหัส (encodings) ซึ่งทำให้เซิร์ฟเวอร์ผลิตข้อมูลจุดภาพได้ง่ายขึ้น

รูปแบบข้อมูลจุดภาพ หมายถึง การแทนสีต่างๆด้วยค่าของจุดภาพ รูปแบบจุดภาพปกติจะเป็น 24 บิต หรือ 16 บิต "true colour" ซึ่งมีเขตข้อมูลบิต (bit-fields) ในค่าจุดภาพเป็นตัวบอกความเข้มสีแดง เขียว และน้ำเงินตรงๆ และรูปแบบ 8 บิต "colour map" ซึ่งมีวิธีส่ง (mapping) ใช้แปลค่าจุดภาพให้เป็นค่าความเข้ม RGB

การเข้ารหัส (Encoding) หมายถึง วิธีที่จะจัดการส่งข้อมูลจุดภาพสี่เหลี่ยมผืนผ้าไปบนสายส่ง ทุกๆ ข้อมูลจุดภาพสี่เหลี่ยมผืนผ้าจะมีส่วนหัวข้อมูลบอกพิกัด x, y ของสี่เหลี่ยมผืนผ้าบนจอภาพ ความกว้าง และความยาวของสี่เหลี่ยมผืนผ้า และชนิดการเข้ารหัส ซึ่งบอกการเข้ารหัสของข้อมูลจุดภาพ ดังนั้นข้อมูลที่ตามมาก็จะใช้การเข้ารหัสตามที่ระบุ

โปรโตคอลนี้ยังสามารถเพิ่มการเข้ารหัสได้อีก สำหรับวิธีที่มีกำหนดไว้ได้แก่ raw encoding, copy encoding, RRE (rise and run length) encoding, CoRRE (Compact RRE) encoding และ hextile encoding

4.1 Raw encoding

เป็นวิธีเข้ารหัสแบบง่ายที่สุด คือ ข้อมูลที่ประกอบด้วยค่าจุดภาพ n ค่า เมื่อ n คือค่าความกว้างคูณความยาวของสี่เหลี่ยมผืนผ้า ค่าเหล่านี้ก็จะแทนจุดภาพแต่ละจุด จากซ้ายไปขวาตามบรรทัด ตามลำดับ ทุก RFB client จะสามารถใช้การเข้ารหัสแบบนี้ได้และ RFB server จะใช้วิธีนี้เข้ารหัสถ้าไม่มีการร้องขอวิธีเข้ารหัสอื่นจากไคล์เอ็นต์

4.2 Copy Rectangle encoding

เป็นวิธีที่ง่ายและมีประสิทธิภาพเมื่อไคล์เอ็นต์มีข้อมูลจุดภาพอยู่แล้วในเฟรมบัฟเฟอร์ การเข้ารหัสโดยปกติจะประกอบด้วยพิกัด x, y ซึ่งการบอกจุดตำแหน่งในเฟรมบัฟเฟอร์จะทำให้สามารถ

สำเนาข้อมูลจุดภาพในเฟรมบัฟเฟอร์ได้ วิธีนี้ใช้ได้กับหลายเหตุการณ์ ที่เห็นได้ชัดคือ เมื่อผู้ใช้เครื่องย้ายวินโดว์และเมื่อเลื่อนข้อมูลในวินโดว์ด้วยการ สครอล (Scroll) นอกจากนั้นก็เช่น การพิมพ์หรือวาดรูปอะไรในรูปแบบซ้ำ เซิร์ฟเวอร์ ที่ฉลาดอาจสามารถส่งรูปแบบต้นฉบับแค่ครั้งเดียวหลังจากทราบตำแหน่งเริ่มต้นในเฟรมบัฟเฟอร์แล้วก็สามารถ ส่งรูปแบบที่มีการเกิดขึ้นซ้ำไปได้ด้วย copy rectangle encoding

4.3 RRE encoding

RRE ย่อมาจาก rise-and-run-length encoding เป็นการเข้ารหัสที่คล้ายกับ run-length encoding ในแกน 2 มิติ มิได้หมายความว่า รูปสี่เหลี่ยมผืนผ้าที่เข้ารหัสแบบ RRE จะบีบอัดได้ถึงขั้นดี หรือดีกว่า run-length encoding แต่สาระสำคัญอยู่ที่ ข้อมูลของสี่เหลี่ยมผืนผ้า ที่ส่งมาถึงไคล์เอ็นต์จะอยู่ในรูปแบบซึ่งสามารถให้แสงและเงาได้ทันที และมีประสิทธิภาพ ด้วยกราฟิกเอนจิน (graphics engines) แบบง่ายๆ ดังนั้น RRE จึงถูกใช้ในสถานการณ์ซึ่งการบีบอัดข้อมูลเป็นที่ยอมรับแล้วแต่ RFB client ยังไม่มีประสิทธิภาพในการคลายการบีบอัดให้รวดเร็วพอที่จะแสดงผลได้

แนวคิดพื้นฐานของ RRE คือการแบ่งข้อมูลจุดภาพสี่เหลี่ยมผืนผ้าเป็นสี่เหลี่ยมผืนผ้าย่อยๆ โดยแต่ละสี่เหลี่ยมผืนผ้าย่อยก็ประกอบด้วยจุดภาพซึ่งมีค่าเดียวกัน เมื่อรวมเอาสี่เหลี่ยมผืนผ้าย่อยๆ ก็จะได้สี่เหลี่ยมผืนผ้าเดิมการแบ่งสี่เหลี่ยมผืนผ้าให้เป็นสี่เหลี่ยมผืนผ้าย่อยๆก็จะสามารถคำนวณได้โดยง่าย

ผลจากการเข้ารหัสในสายส่งจะประกอบด้วยค่าของจุดภาพที่เป็นพื้นหลังภาพ V_b (ปกติจะเป็นสีส่วนมากของสี่เหลี่ยมผืนผ้า) จำนวน N คือจำนวนสี่เหลี่ยมผืนผ้าย่อยที่จะตามมา ซึ่งแต่ละผืนสี่เหลี่ยมจะประกอบด้วยลำดับ $<v,x,w,h>$ เมื่อ v (ไม่ใช่ V_b) คือค่าจุดภาพ ส่วน (x,y) คือตำแหน่งพิกัดมุมบนซ้ายของสี่เหลี่ยมผืนผ้าย่อย และ (w,h) คือความกว้างและความยาวของสี่เหลี่ยมผืนผ้าย่อยนั้น ไคล์เอ็นต์ก็จะให้แสงและเงาได้เหมือนกับรูปต้นฉบับด้วยการระบายสีพื้นหลังภาพ และวาดสี่เหลี่ยมผืนผ้าย่อยตรงตำแหน่งต่างๆตามที่ระบุในสี่เหลี่ยมผืนผ้าย่อยนั้น

4.4 CoRRE encoding

CoRRE เป็นวิธีของ RRE อีกแบบหนึ่งซึ่งประกันว่าสี่เหลี่ยมผืนผ้าที่ใหญ่ที่สุดที่ถูกส่งจะไม่ใหญ่กว่า 255×255 จุดภาพ เซิร์ฟเวอร์ซึ่งต้องการส่งสี่เหลี่ยมผืนผ้าที่ใหญ่กว่านี้ต้องแบ่งข้อมูลก่อนส่งเป็นสี่เหลี่ยมย่อยๆ (RFB rectangles) ภายในแต่ละสี่เหลี่ยมผืนผ้าที่เล็กลงนี้จะใช้เพียง 1 ไบต์แทนตำแหน่งของสี่เหลี่ยมย่อยๆอีก การบีบอัดด้วยวิธีนี้กับหน้าจอปกติจะให้ผลดีกว่าวิธี RRE อันที่จริงการบีบอัดจะให้ผลดีที่สุดเมื่อเราจำกัดขนาดของสี่เหลี่ยมผืนผ้า แต่ที่ใช้กันอยู่สำหรับวิธีนี้มีขนาดจำกัดที่ใหญ่ที่สุด 48×48 จุด ทั้งนี้เพราะว่าสี่เหลี่ยมผืนผ้าที่ไม่ได้เข้ารหัส (ประกอบด้วยข้อมูลรูป

ภาพ) จะส่งแบบ raw ขณะที่ส่วนที่เข้ารหัสจะเป็นแบบ CoRRE ดังนั้นการกำหนดให้ขนาดสี่เหลี่ยมผืนผ้ามีขนาดจำกัดที่เล็กกว่าก็จะช่วยให้การเข้ารหัสดีขึ้น สำหรับ RRE การส่งจะเป็นแบบ RRE ทั้งหมดหรือไม่ก็ต้องเป็นแบบ raw ทั้งหมด อย่างไรก็ดีแต่ละ RFB rectangle จะมีการสิ้นเปลือง (overhead) ดังนั้นการทำให้ขนาดจำกัดของสี่เหลี่ยมมีขนาดเล็ก (ทำให้เพิ่มจำนวน RFB rectangle) จะทำให้ผลการบีบอัดแย่ลง

4.5 Hextile encoding

Hextile เป็นวิธีของ RRE อีกแบบหนึ่งซึ่งสี่เหลี่ยมผืนผ้าจะถูกแบ่งออกเป็นหลายผืนๆหนึ่งมีขนาด 16×16 จุด แต่ละผืนสี่เหลี่ยมย่อยมี 4 บิตเป็นตัวบอกพิกัดตำแหน่ง รวมใช้ 16 บิต ต่างจาก CoRRE ที่ผืนสี่เหลี่ยมจะไม่เป็นลักษณะ RFB rectangle ระดับบน (top-level RFB rectangle) เมื่อแบ่งสี่เหลี่ยมผืนผ้าต้นแบบเป็นส่วนๆ การแบ่งจะเหมือนได้ถูกกำหนดเอาไว้แล้ว กล่าวคือตำแหน่งและขนาดของแต่ละผืนสี่เหลี่ยมไม่ต้องระบุอีก สิ่งที่ต้องใช้เข้ารหัสของสี่เหลี่ยมผืนผ้าก็เป็นเพียงลำดับของข้อมูลผืนสี่เหลี่ยมย่อยที่เรียงตามกัน ลำดับของผืนสี่เหลี่ยมย่อยจะเริ่มจากมุมบนซ้าย ลำดับจากซ้ายไปขวาและบนลงล่าง ถ้าขนาดความกว้างของสี่เหลี่ยมย่อยไม่พอดี 16 จุด ในผืนสี่เหลี่ยมย่อยสุดท้ายก็จะเล็กกว่าผืนอื่น เช่นเดียวกับความสูงทั้งหมดของสี่เหลี่ยมผืนผ้าย่อย

แต่ละผืนสี่เหลี่ยมย่อยจะถูกเข้ารหัสเป็น raw pixel data หรือในแนวทางของ RRE ก็ได้ขึ้นอยู่กับการระบุแต่ละผืนสี่เหลี่ยมย่อยจะมีค่าจุดภาพของพื้นหลังบอกก่อน อย่างไรก็ดีค่าจุดภาพของพื้นหลังอาจไม่ต้องมีถ้าเหมือนกับลำดับผืนก่อนหน้า ถ้าทุกสี่เหลี่ยมผืนผ้าย่อยมีค่าจุดภาพเพียงค่าเดียวเหมือนกัน ก็สามารถกำหมดค่าจุดภาพของพื้นหน้าเพียงค่าเดียวได้เช่นกัน

5.ข้อความของโปรโตคอล (Protocol Messages)

โปรโตคอล RFB ที่ใช้บนการรับส่งที่น่าเชื่อถือไม่ว่าจะเป็นกระแสข้อมูล (byte stream) หรือข้อความ (messages) จะมี 2 ขั้นตอนของโปรโตคอล คือ ขั้นตอนเริ่มการติดต่อ ตามด้วยขั้นตอนติดต่อตามปกติ

ขั้นตอนเริ่มการติดต่อ ประกอบด้วยข้อความ ProtocolVersion, Authentication, ClientInitialisation และ ServerInitialisation ดังจะได้กล่าวต่อไป สังเกตว่าทั้งไคล์เอ็นต์และเซิร์ฟเวอร์ต่างก็ส่งข้อความ ProtocolVersion

ขั้นตอนติดต่อตามปกติ จะเกิดขึ้นหลังจากข้อความ ServerInitialisation เกิดขึ้นในขั้นตอนนี้ไคล์เอ็นต์จะส่งข้อความไปร้องขอข้อความตามที่ต้องการจากเซิร์ฟเวอร์ ทุกข้อความจะเริ่มด้วยไบต์ชนิดของข้อความ (message-type) แล้วตามด้วยข้อมูลของข้อความนั้น

ตามคำอธิบายของข้อความของโปรโตคอลจะใช้รูปแบบ CARD8, CARD16, CARD32, INT8, INT16, INT32 ซึ่งแทนรูปแบบจำนวนเต็มบวก (unsigned integer) 8, 16, 32 บิต ตามลำดับ และ 8, 16, 32 signed integer bit ตามลำดับ ทุกๆ multi byte integer จะเก็บในลักษณะ big endian order (เริ่มเรียงจากไบต์หน้า)

5.1 ข้อความเริ่มการติดต่อ (Initial Handshaking Messages)

5.1.1 ProtocolVersion

การติดต่อเริ่มขึ้นโดยเซิร์ฟเวอร์ส่งข้อความ ProtocolVersion ไปที่ไคล์เอ็นต์ ทำให้ไคล์เอ็นต์รู้ว่าเลขที่ของโปรโตคอล (RFB protocol version number) ที่สนับสนุนโดยเซิร์ฟเวอร์เป็นเท่าไร ไคล์เอ็นต์ก็จะตอบข้อความเดียวกันนี้ด้วยเลขรุ่นของโปรโตคอลที่จะใช้จริง (อาจจะแตกต่างจากที่ระบุโดยเซิร์ฟเวอร์ก็ได้)

วัตถุประสงค์ของกลไกนี้ก็เพื่อให้เกิดความเข้ากันได้ของทั้ง 2 ฝ่าย ในกรณีที่เลขรุ่นโปรโตคอลไม่ตรงกันก็สามารถย้อนกลับไปใช้เลขรุ่นของโปรโตคอลที่ต่ำกว่าที่สนับสนุนได้ ตัวอย่างเช่น ถ้าไคล์เอ็นต์ขอใช้เลขรุ่น 3.1 แต่เซิร์ฟเวอร์ใช้เลขรุ่น 3.0 กรณีนี้อาจทำให้เกิดการไม่เข้าใจกันสำหรับการร้องขอใช้การเข้ารหัสที่รู้จักได้ แต่ทุกอย่างก็ยังใช้ได้อยู่ เนื่องจากเลขรุ่นหลักของโปรโตคอล (major version number)

ProtocolVersion ประกอบด้วย 12 ไบต์ เป็นสายอักขระ (string) ของแอสกี (ASCII American Standard Code for Information Interchange) อยู่ในรูปแบบ "RFB xxx.yyy\n" เมื่อ xxx และ yyy คือเลขรุ่นหลัก (major version number) และ เลขรุ่นรอง (minor version number) ตามลำดับ เติมเต็มด้วยเลข 0

ตารางที่ ก.1 แสดงรูปแบบของ ProtocolVersion

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

5.1.2 Authentication

เมื่อเลือก ProtocolVersion เสร็จแล้วเซิร์ฟเวอร์จะส่งรูปแบบการพิสูจน์ตัวจริง (authentication scheme) เพื่อใช้ในการติดต่อ

ตารางที่ ก.2 แสดงรูปแบบ authentication scheme

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

connection failed – มีบางอย่างทำให้การติดต่อขัดข้อง (นั่นคือเซิร์ฟเวอร์ไม่สามารถสนับสนุน protocol version ตามที่ต้องการได้) แบบนี้จะตามด้วยสายอักขระอธิบายเหตุผลที่ทำให้ขัดข้อง (ระบุด้วยความยาวของสายอักขระ ตามด้วยสายอักขระของแอสกี)

ตารางที่ ก.3 แสดงรูปแบบ connection failed

No. of bytes	Type [Value]	Description
4	CARD32	*reason-length*
reason-length	CARD8 array	*reason-string*

no authentication – ไม่ต้องมีการพิสูจน์ตัวจริง โปรโตคอลจะเริ่มต้นส่งข้อความ ClientInitialisation เลย

VNC authentication – ใช้การพิสูจน์ตัวจริงของ VNC แบบนี้จะตามด้วยเขตข้อมูล challenge ที่สุ่มค่ามาซึ่งยาว 16 ไบต์

ตารางที่ ก.4 แสดงรูปแบบของ VNC authentication challenge

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

ไคล์เอ็นต์จะเข้ารหัสลับ challenge ด้วยวิธี DES โดยการใช้รหัสลับที่ผู้ใช้ป้อนข้อมูลมาเป็น key และส่งเขตข้อมูล response เป็นผลลัพธ์

ตารางที่ ก.5 แสดงรูปแบบของ VNC authentication response

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

เซิร์ฟเวอร์จะส่งคำบอกแจ้งไคล์เอ็นต์ว่าการพิสูจน์ตัวจริงสำเร็จหรือไม่ ถ้าสำเร็จโปรโตคอลจะเริ่มส่งข้อความ ClientInitialisation ถ้าไม่สำเร็จเซิร์ฟเวอร์จะปิดการติดต่อ

ตารางที่ ก.6 แสดงรูปแบบของ VNC authentication ส่วนที่เซิร์ฟเวอร์แจ้งการพิสูจน์ตัวจริง

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

ถ้าเซิร์ฟเวอร์แจ้งมาเป็นเขตข้อมูล too-many แสดงว่ามีการผิดหลายครั้งเกินไปและจะไม่ยอมให้มีการติดต่อจากไคล์เอ็นต์เดิมอีก

5.1.3 ClientInitialisation

เมื่อไคล์เอ็นต์และเซิร์ฟเวอร์แน่ใจว่าจะติดต่อกันได้ ไคล์เอ็นต์จะเริ่มส่งข้อความดังนี้

ตารางที่ ก.7 แสดงรูปแบบข้อความ Initialisation

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

Share-flag จะไม่เป็น 0 ถ้าเซิร์ฟเวอร์ยอมให้ไคล์เอ็นต์หลายเครื่องติดต่อมาพร้อมกัน แต่ถ้าเป็น 0 หมายความว่ายอมให้ไคล์เอ็นต์ติดต่อมาเพียงเครื่องเดียว

5.1.4 ServerInitialisation

หลังจากรับข้อความ ClientInitialisation แล้ว เซิร์ฟเวอร์จะส่งข้อความ ServerInitialisation เพื่อบอกไคล์เอ็นต์เกี่ยวกับความกว้างและความยาวเฟรมบัฟเฟอร์ของเซิร์ฟเวอร์, รูปแบบจุดภาพ, ชื่อเครื่องและชื่อผู้ใช้

ตารางที่ ก.8 แสดงรูปแบบข้อความ ServerInitialisation

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*

เมื่อ PIXEL_FORMAT คือ

ตารางที่ ก.9 แสดงรูปแบบ PIXEL_FORMAT

No. of bytes	Type [Value]	Description
1	CARD8	*bits-per-pixel2*
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 จะระบุรูปแบบจุดภาพของเซิร์ฟเวอร์ รูปแบบจุดภาพนี้จะถูกใช้ถ้าไคล์เอ็นต์ไม่ร้องขอใช้รูปแบบที่แตกต่างไปจากนี้ด้วยข้อความ SetPixelFormat (5.2.1)

Bit-per-pixel เป็นจำนวนบิตที่ใช้สำหรับแต่ละจุดภาพ จะต้องมากกว่าหรือเท่ากับ depth ซึ่งหมายถึงจำนวนบิตต่อจุดภาพที่ใช้ได้ โดยปกติ bit-per-pixel ต้องเป็น 8, 16 หรือ 32 –ไม่สนับสนุนจำนวนบิตที่น้อยกว่า 8 บิต Big-endian-flag จะไม่เป็น 0 ถ้าใช้ big-endian กับ multi-byte pixel

ถ้า true-colour-flag ไม่เป็น 0 แล้ว ให้ดูเขตข้อมูลที่เหลืออีก 6 เขตข้อมูล จะบอกวิธีอ่านรูปแบบความเข้มสีแดง, เขียว และน้ำเงิน จากค่าจุดภาพ Red-max เป็นค่าสูงสุดของสีแดง ($=2^n-1$ เมื่อ n คือจำนวนบิตที่ใช้สำหรับสีแดง) ค่าเหล่านี้มักจะอยู่ในลำดับ big-endian Red-shift คือจำนวนของการเลื่อนบิตเพื่อให้ได้ค่าสีแดง ทำนองเดียวกับ green-Max, green-shift และ blue-max, blue-shift ตัวอย่างเช่น ต้องการหาค่าสีแดง (มีค่าตั้งแต่ 0 ถึง red-max) จากจุดภาพหนึ่ง ให้ทำดังนี้

- สลับค่าจุดภาพตามค่า big-endian-flag ให้ถูกต้อง (ถ้า big-endian-flag เป็น 0 แต่ลำดับไบต์เป็น big-endin ให้สลับ)
- เลื่อนบิตทางขวาด้วย red-shift
- AND ด้วย red-max (ตามลำดับไบต์)

ถ้า true-colout-flag เป็น 0 แล้วเซิร์ฟเวอร์จะใช้จุดภาพซึ่งไม่ได้ประกอบด้วยสีแดง, เขียว และน้ำเงินโดยตรง แต่จะใช้วิธีบอกด้วยการแทนสี (colour map) หน่วยข้อมูลในการแทนสีสามารถตั้งค่าโดยไคล์เอ็นต์ด้วยข้อความ FixColourMapEntries (5.5.2) หรือโดยเซิร์ฟเวอร์ด้วยข้อความ SetColourMaoEntries (5.3.2)

5.2 ข้อความจากไคล์เอ็นต์ไปยังเซิร์ฟเวอร์ (Client to Srver Messages)

5.2.1 SetPixelFormat

กำหนดรูปแบบค่าจุดภาพที่จะส่งโดยข้อความ FramebufferUpdate ถ้าไคล์เอ็นต์ไม่ได้ส่งข้อความ SetPixelFormat แล้วเซิร์ฟเวอร์จะส่งค่าจุดภาพตามที่ระบุไว้ตามข้อความ ServerInitialisation (5.1.4)

ถ้า true-colour-flag เป็น 0 แสดงว่าใช้ “colour map” ไคล์เอ็นต์สามารถแก้หน่วยข้อมูลในเขตข้อมูล colour map โดยใช้ข้อความ FixColourMapEntries (5.5.2) หน่วยข้อมูลใดที่ไม่ได้แก้ไขด้วยไคล์เอ็นต์ อาจจะกำหนดได้แบบไดนามิก (dynamically) ตามต้องการโดยเซิร์ฟเวอร์ด้วยข้อความ SetColourMapEntries (5.3.2) ทันทีที่ไคล์เอ็นต์ส่งข้อความนี้แล้ว colour map จะถูกกำหนดให้มีค่าว่าง แม้ว่าหน่วยข้อมูลเดิมถูกแก้ไขโดยไคล์เอ็นต์หรือถูกกำหนดไว้โดยเซิร์ฟเวอร์ก็ตาม

ตารางที่ ก.10 แสดงรูปแบบข้อความ SetPixelFormat

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

เมื่อ PIXEL_FORMAT อธิบายไว้แล้วใน 5.1.4

5.2.2 FixColourMapEntries

เมื่อรูปแบบจุดภาพใช้ "colour map" ข้อความนี้จะบอกเซิร์ฟเวอร์ว่าค่าจุดภาพที่ระบุมาจะแทนค่าเป็นค่าความเข้มสี RGB ใด หมายความว่าเซิร์ฟเวอร์อาจจะไม่ได้กำหนดค่าความเข้มสี RGB สำหรับค่าจุดภาพโดยข้อความ SetColourMapEntries (5.3.2)

ถ้าไคล์เอ็นต์มีการแก้ไขการแทนค่าสีก็สามารถจะส่งข้อความ FixColourMapEntries เพื่ออธิบายการแทนค่าสีทั้งหมดและเซิร์ฟเวอร์จะแปลทุกค่าจุดภาพให้เหมาะสมสำหรับการแทนค่าสีของไคล์เอ็นต์ ในกรณีนี้เซิร์ฟเวอร์จะไม่ส่งข้อความ SetColourMapEntries

อย่างไรก็ดี วิธีที่ไคล์เอ็นต์สามารถจะกำหนดค่าจุดภาพโดยไม่ต้องให้แก้ไขการแทนค่าสีอีกก็โดยการส่งข้อความ SetPixelFormat ซึ่งใช้ได้กับทุกการแทนค่าสี แต่สำหรับข้อความ FixColourMapEntries เวลานี้ยังไม่สนับสนุนโดยทุกเซิร์ฟเวอร์

ตารางที่ ก.11 แสดงรูปแบบข้อความ FixColourMapEntries

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

ตามด้วยเซตข้อมูล number-of-colour ที่ซ้ำๆกันดังนี้

ตารางที่ ก.12 แสดงรูปแบบเขตข้อมูล number-of-colour

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

5.2.3 SetEncodings

กำหนดรูปแบบการเข้ารหัสในข้อมูลจุดภาพที่จะให้เซิร์ฟเวอร์ส่ง

ตารางที่ ก.13 แสดงรูปแบบข้อความ SetEncodings

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

ตามด้วยเขตข้อมูล number-of-encodings ที่ซ้ำๆกันดังนี้

ตารางที่ ก.14 แสดงรูปแบบเขตข้อมูล number-of-encodings

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

5.2.4 FramebufferUpdateRequest

บอกเซิร์ฟเวอร์ว่าไคล์เอ็นต์สนใจในพื้นที่ใดของเฟรมบัฟเฟอร์โดยตำแหน่งพิกัด x-position, y-position, width และ height เซิร์ฟเวอร์มักจะตอบสนองกับข้อความ FramebufferUpdateRequest โดยส่งข้อความ FramebufferUpdate โดยที่ FramebufferUpdate อาจจะส่งเพียงครั้งเดียวเพื่อตอบข้อความ FramebufferUpdateRequest ที่ส่งมาหลายๆครั้งก็ได้

เซิร์ฟเวอร์จะสันนิษฐานไว้ก่อนว่าไคล์เอ็นต์ได้เก็บพื้นที่สนใจไว้ ซึ่งเซิร์ฟเวอร์ก็เพียงต้องส่งส่วนที่มีการเปลี่ยนแปลงให้ไคล์เอ็นต์เท่านั้น

อย่างไรก็ดี ถ้ามีเหตุผลบางอย่างทำให้ไคล์เอ็นต์สูญเสียส่วนของพื้นที่ที่มีอยู่ไป ไคล์เอ็นต์จะส่งข้อความ FramebufferUpdateRequest และกำหนดให้ค่า incremental เป็น 0 ซึ่งจะทำให้เซิร์ฟเวอร์ส่งพื้นที่ทั้งหมดกลับมาให้เร็วที่สุด แต่พื้นที่นั้นจะไม่เปลี่ยนแปลงให้ทันเวลาถ้าใช้การเข้ารหัสแบบ copy rectangle

ถ้าไคล์เอ็นต์ยังไม่สูญเสียส่วนของพื้นที่ที่สนใจไป ก็จะส่งข้อความ FramebufferUpdateRequest ด้วยค่า incremental ที่ไม่เป็น 0 เมื่อมีเปลี่ยนแปลงในพื้นที่นั้นในเฟรมบัฟเฟอร์ เซิร์ฟเวอร์ก้จะส่งข้อความ FramebufferUpdate ส่วนระยะเวลาระหว่างการส่งข้อความ FramebufferUpdateReques และ FramebufferUpdate ไม่ได้กำหนดไว้

ในกรณีที่ไคล์เอ็นต์มีความรวดเร็วอาจจะ ต้องการส่งเป็นระยะๆโดยข้อความ FramebufferUpdateRequest เพื่อหลีกเลี่ยงการใช้เครือข่ายอย่างเห็นแก่ตัว

ตารางที่ ก.15 แสดงรูปแบบข้อความ FramebufferUpdateRequest

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*

5.2.5 KeyEvent

คือเหตุการณ์ที่มีการกดหรือปล่อยแป้นอักขระ Dawn-flag จะไม่เป็น 0 ถ้าแป้นอักขระกดอยู่ และเป็น 0 ถ้าปล่อยแป้นอักขระอยู่ เขตข้อมูล key จะถูกระบุด้วย “keysym” ซึ่งกำหนดไว้ในระบบเอกซ์วินโดว์ (X Window System)

ตารางที่ ก.16 แสดงรูปแบบของ KeyEvent

No. of bytes	Type	[Value]	Description
1	CARD8	4	*message-type*
1	CARD8		*down-flag*
2			*padding*
4	CARD32		*key*

สำหรับ "keysym" จะมีค่าตามเป็น ASCII ดังนี้

ตารางที่ ก.17 แสดงค่าของ "keysym"

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
F1	0xffbe
F2	0xffbf
F3	0xffc0
F4	0xffc1

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Key name	Keysym value
F5	0xffc2
F6	0xffc3
F7	0xffc4
F8	0xffc5
F9	0xffc6
F10	0xffc7
F11	0xffc8
F12	0xffc9
Shift (left)	0xffe1
Shift (right)	0xffe2
Control (left)	0xffe3
Control (right)	0xffe4
Meta (left)	0xffe7
Meta (right)	0xffe8
Alt (left)	0xffe9
Alt (right)	0xffea

5.2.6 PointerEvent

จะเป็นสิ่งที่บอกทั้งการเคลื่อนไหวและการกดหรือปล่อยพอยเตอร์ โดยบอกตำแหน่งพิกัด (x-position, y-position) และสถานะของปุ่มพอยเตอร์ตั้งแต่ค่า 0 ถึง 7 ในเซตข้อมูล button-mask เมื่อ 0 หมายถึงปล่อย และ 1 หมายถึงกด

ตารางที่ ก.18 แสดงรูปแบบ PointerEvent

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

5.2.7 ClientCutText

เมื่อไคล์เอ็นต์มีข้อความ ASCII ใหม่ในบัฟเฟอร์ ส่วนท้ายบรรทัดจะแทนด้วยอักขระป้อนบรรทัด (line feed) และอักขระขึ้นบรรทัดใหม่ (new line) เท่านั้น ไม่จำเป็นต้องมีอักขระปัดแคร่ (cariage-return)

ตารางที่ ก.19 แสดงรูปแบบ ClientCutText

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

5.3 FramebufferUpdate

ประกอบด้วยลำดับของข้อมูลจุดภาพของสี่เหลี่ยมผืนผ้า ซึ่งไคล์เอ็นต์จะใส่มาในเฟรมบัฟเฟอร์ของตัวเอง การส่งนี้จะใช้สำหรับตอบสนองข้อความ FramebufferUpdateRequest จากไคล์เอ็นต์ สำหรับระยะเวลาระหว่างข้อความ FramebufferUpdateRequest และข้อความ FramebufferUpdate ยังไม่มีข้อจำกัดแน่นอน

ตารางที่ ก.20 แสดงรูปแบบข้อความ FramebufferUpdate

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

ข้อความนี้ตามด้วยเขตข้อมูล number-of-rectangles ซึ่งเป็นข้อมูลจุดภาพของรูปสี่เหลี่ยมผืนผ้าหลายๆรูป แต่ละรูปประกอบด้วย

ตารางที่ ก.21 แสดงรูปแบบ number-of-rectangles

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

No. of bytes	Type	[Value]	Description
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

สำหรับข้อมูลจุดภาพมีการเข้ารหัสดังนี้

- Raw encoding ประกอบด้วยค่าจุดภาพขนาด width× height ค่าเหล่านี้จะแทนแต่ละจุดภาพจากซ้ายไปขวาของแต่ละสแกนไลน์ (scanline)
- Copy Rectangle ประกอบด้วย

ตารางที่ ก.22 แสดงรูปแบบ Copy Rectangle

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

- RRE ประกอบด้วยข้อมูลส่วนหัวดังนี้

ตารางที่ ก.23 แสดงรูปแบบ RRE ส่วนหัว

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

เมื่อ $8n$ เป็นจำนวนบิตของเขตข้อมูล bit-per-pixel ที่ตกลงกันโดยไคล์เอ็นต์และเซิร์ฟเวอร์แล้วด้วยข้อความ ServerInitialisation (5.1.4) หรือ SetPixelFormat (5.2.1) ก็ได้ ข้อมูลเหล่านี้จะตามด้วยข้อมูลแต่ละสี่เหลี่ยมย่อย ซึ่งมีโครงสร้างข้อมูลดังนี้

ตารางที่ ก.24 แสดงรูปแบบสี่เหลี่ยมย่อยของ RRE

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*

- CoRRE ประกอบด้วยข้อมูลส่วนหัวดังนี้

ตารางที่ ก.25 แสดงรูปแบบ CoRRE ส่วนหัว

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

เช่นเดียวกับการเข้ารหัส RRE ข้อมูลนี้จะตามด้วยข้อมูลแต่ละสี่เหลี่ยมย่อย ซึ่งมีโครงสร้างดังนี้

ตารางที่ ก.26 แสดงรูปแบบสี่เหลี่ยมย่อยของ CoRRE

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*

- Hextile

การเข้ารหัสแบบนี้สี่เหลี่ยมผืนผ้าจะถูกแบ่งเป็นขนาด 16×16 จุดภาพ โดยเริ่มจากจุดมุมบนซ้ายจากซ้ายไปขวาและบนลงล่าง ถ้าความกว้างของสี่เหลี่ยมผืนผ้าไม่ใช่มีขนาดเท่าผลคูณของ 16 จะทำให้ความกว้างของสี่เหลี่ยมย่อยแถวสุดท้ายเล็กกว่า 16 จุดภาพ เช่นเดียวกับความยาวอาจจะยาวน้อยกว่า 16 จุดภาพในแถวสุดท้าย

แต่ละสี่เหลี่ยมย่อยจะเริ่มต้นด้วย subencoding ดังนี้

ตารางที่ ก.27 แสดงรูปแบบ subencoding type byte

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

ดังนั้นถ้ามีการกำหนดค่าบิตเป็น Raw ข้อมูลที่ตามมาจะเป็นค่าจุดภาพขนาด width×height มิเช่นนั้นข้อมูลที่ตามมาจะมี

BackgroundSpecified – กำหนดค่าสีพื้นหลังของสี่เหลี่ยมย่อย ดังนี้

ตารางที่ ก.28 แสดงรูปแบบ background-pixel-value

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

เมื่อ 8*n* เป็นจำนวนบิตของเซตข้อมูล bit-per-pixel ที่ตกลงกันโดยไคล์เอ็นต์และเซิร์ฟเวอร์แล้วด้วยข้อความ ServerInitialisation (5.1.4) หรือ SetPixelFormat (5.2.1) ก็ได้ ในสี่เหลี่ยมย่อยแรกที่ไม่ใช่การเข้ารหัสแบบ raw ต้องมีการกำหนดบิตนี้เสมอ แต่ถ้าไม่มีการจะถือว่าพื้นหลังมีสีเหมือนสี่เหลี่ยมย่อยที่แล้ว

ForegroundSpecified – ถ้ามีการกำหนดค่าไว้ ค่าจุดภาพที่ตามมาจะมีสีพื้นหลังสำหรับทุกสี่เหลี่ยมย่อยดังที่กำหนดดังนี้

ตารางที่ ก.29 แสดงรูปแบบ foreground-pixel-value

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

ถ้ามีการกำหนดบิตแล้วเซตข้อมูล SubrectsColoured ต้องกำหนดค่าเป็น 0

AnySubrects – ถ้ามีการกำหนดค่าไว้ ไบต์ที่ตามมาจะบอกจำนวนของสี่เหลี่ยมย่อยที่มี ดังนี้

ตารางที่ ก.30 แสดงรูปแบบ number-of-subrectangles

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

ถ้าไม่มีการกำหนดค่าไว้ แสดงว่าไม่มีสี่เหลี่ยมย่อยเลย(หมายความว่าสีพื้นหลังเหมือนกันหมด)

SubrectsColoured – ถ้ามีการกำหนดค่าไว้ แต่ละสี่เหลี่ยมย่อยจะกำหนดค่าจุดภาพให้มีสีดังนี้

ตารางที่ ก.31 แสดงรูปแบบการกำหนดค่าสีของสี่เหลี่ยมย่อย

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

ถ้าไม่มีการกำหนดค่าไว้ ทุกสี่เหลี่ยมย่อยจะมีสีพื้นหน้าเดียวกัน มิเช่นนั้นถ้าเซตข้อมูล ForegroundSpecified ไม่ได้กำหนดค่า สีพื้นหน้าจะเหมือนกับสี่เหลี่ยมย่อยที่อยู่ก่อนหน้า โดยจะมีรูปแบบดังนี้

ตารางที่ ก.32 แสดงรูปแบบการบอกตำแหน่งสี่เหลี่ยมย่อยโดยใช้ค่าสีเหมือนกับสี่เหลี่ยมย่อยที่อยู่ก่อนหน้า

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

ตำแหน่งและขนาดของแต่ละสี่เหลี่ยมย่อยจะถูกกำหนดไว้ใน 2 ไบต์ ในเขตข้อมูล x-and-y-position และ width-and-height โดยที่ 4 บิตแรกของเขตข้อมูล x-and-y-position จะบอกตำแหน่งในแนวนอน อีก 4 บิตที่เหลือจะบอกตำแหน่งในแกนตั้ง และ 4 บิตแรกของเขตข้อมูล width-and-height จะบอกความกว้าง – 1 และที่เหลือจะบอกความยาว –1

5.3.2 SetColourMapEntries

การแทนสีปัจจุบันยังสนุนแค่ระบบ X-base server

ตารางที่ ก.33 แสดงรูปแบบข้อความ SetColourMapEntries

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

โดยที่เขตข้อมูล number-of-colours ที่ตามมาจะมีรูปแบบที่ซ้ำกัน ดังนี้

ตารางที่ ก.34 แสดงรูปแบบเขตข้อมูล number-of-colours

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

5.3.3 Bell

ส่งเสียงที่ไคล์เอ็นต์

ตารางที่ ก.35 แสดงรูปแบบข้อความ Bell

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

5.3.4 ServerCutText

ตารางที่ ก.36 แสดงรูปแบบข้อความ ServerCutText

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

จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข

คู่มือการใช้โปรแกรมเฝ้าสังเกตจอภาพของผู้เรียนผ่านระบบเครือข่ายห้องเรียน

โปรแกรมเฝ้าสังเกตจอภาพของผู้เรียนผ่านระบบเครือข่ายห้องเรียนเป็นโปรแกรมสำหรับแสดงภาพหน้าจอของเครื่องคอมพิวเตอร์ผู้เรียนในห้องเรียนคอมพิวเตอร์ โดยจะติดตามเฝ้าสังเกตการทำงานของจอภาพผู้เรียนเสมือนสังเกตที่หน้าเครื่องคอมพิวเตอร์ผู้เรียนได้ทีละหลายเครื่อง และในทางกลับกันสามารถแสดงจอภาพของผู้สอนให้ผู้เรียนได้สามารถติดตามจอภาพที่ผู้สอนสังเกตได้เช่นกัน

ส่วนประกอบการทำงานที่สำคัญ

1. ส่วนการเพิ่มจอภาพที่ต้องการสังเกตใหม่
   1.1. การเพิ่มจอภาพที่ต้องการสังเกตโดยทราบชื่อเครื่องคอมพิวเตอร์ที่อยู่ในกลุ่มงาน (work group) เดียวกัน
   1.2. การเพิ่มจอภาพที่ต้องการสังเกตโดยไม่ทราบชื่อเครื่องคอมพิวเตอร์ที่อยู่ในกลุ่มงาน (work group) เดียวกัน
2. ส่วนการแสดงจอภาพที่ต้องการสังเกต
3. ส่วนการแสดงจอภาพของผู้สอนให้ผู้เรียนสังเกต
4. ส่วนตัวเลือกของระบบ
   4.1. การเวียนจอภาพที่ต้องการสังเกตและยกเลิกการเวียน
   4.2. การอนุญาตให้มีการส่งจอภาพจากผู้เรียนมาปรากฏที่ผู้สอน
   4.3. การตรวจสอบรายชื่อผู้เรียน (รายชื่อผู้ใช้งาน)
   4.4. ตัวเลือกของระบบอื่น ๆ
      4.4.1. ตัวเลือกของวินโดว์จอภาพเฝ้าสังเกต
      4.4.2. ตัวเลือกของโปรแกรมที่ฝ่ายถูกเฝ้าสังเกต

วินโดว์หลัก

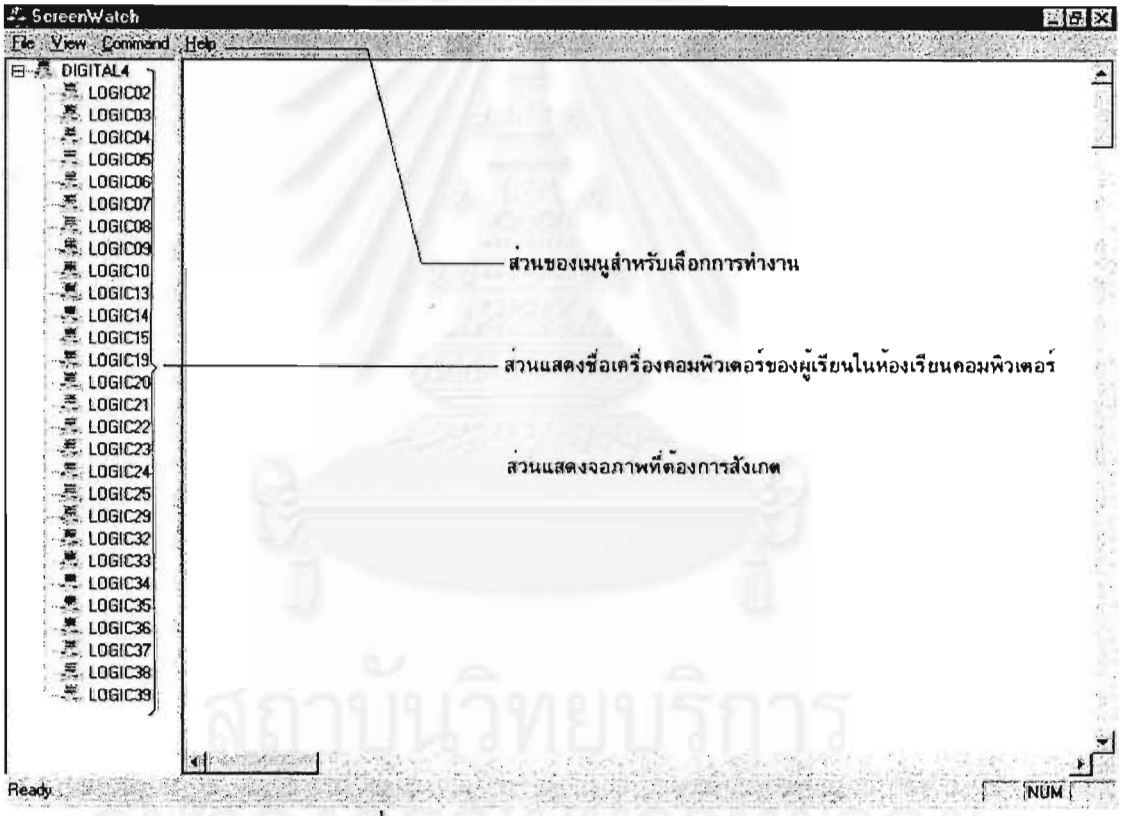
ประกอบด้วยส่วนของเมนูสำหรับเลือกการทำงาน ส่วนแสดงชื่อเครื่องคอมพิวเตอร์ของผู้เรียนในห้องเรียนคอมพิวเตอร์ และส่วนแสดงจอภาพที่ต้องการสังเกต

1. การเพิ่มจอภาพที่ต้องการสังเกตใหม่

เมื่อต้องการจอภาพที่ต้องการสังเกตใหม่ สามารถทำได้โดยกดปุ่มขวาเม้าส์ที่ส่วนแสดงชื่อเครื่องคอมพิวเตอร์ของผู้เรียนในห้องเรียนคอมพิวเตอร์ ที่ชื่อเครื่องคอมพิวเตอร์ที่ต้องการสังเกต หรือ เลือกจากเมนูสำหรับเลือกการทำงานโดยใส่ชื่อเครื่องที่ต้องการสังเกต รายละเอียดขั้นตอนที่สามารถทำได้มีดังนี้

1.1. การเพิ่มจอภาพที่ต้องการสังเกตโดยทราบชื่อเครื่องคอมพิวเตอร์ที่อยู่ในกลุ่มงาน (work group) เดียวกัน

ถ้าต้องการสังเกตโดยทราบชื่อเครื่องคอมพิวเตอร์ที่อยู่ในกลุ่มงาน สามารถทำได้โดยการเลือกเมนู "Command" – "New Connection" (ตามรูปที่ ข.2) จะปรากฏวินโดว์สำหรับใส่ชื่อเครื่องคอมพิวเตอร์ที่จะเฝ้าสังเกตโดยใส่ชื่อเครื่อง หรือ ที่อยู่ไอ พี ก็ได้ จะปรากฏวินโดว์สำหรับใส่รหัสผ่าน

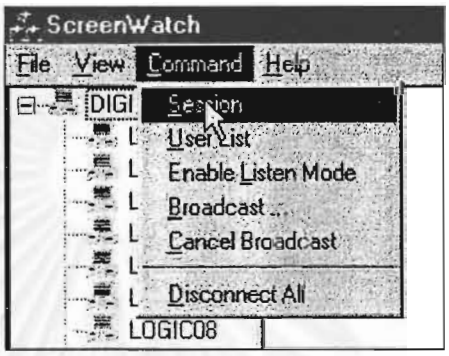


รูปที่ ข.1 แสดงวินโดว์หลักของโปรแกรม

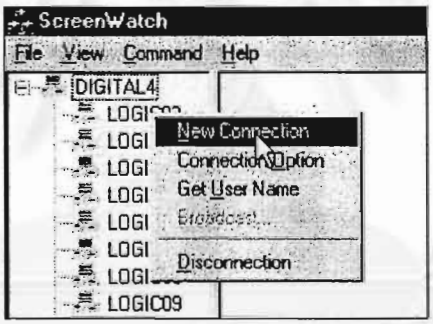
ของเครื่องคอมพิวเตอร์ที่จะเฝ้าสังเกต ถ้าใส่รหัสผ่านถูกต้องจะปรากฏวินโดว์จอภาพที่ต้องการเฝ้าสังเกต

1.2. การเพิ่มจอภาพที่ต้องการสังเกตโดยไม่ทราบชื่อเครื่องคอมพิวเตอร์ที่อยู่ในกลุ่มงาน (work group) เดียวกัน

ถ้าต้องการสังเกตโดยไม่ทราบชื่อเครื่องคอมพิวเตอร์ที่อยู่ในกลุ่มงาน สามารถทำได้โดยการเลือกชื่อเครื่องคอมพิวเตอร์ที่จะเฝ้าสังเกตที่ส่วนแสดงชื่อเครื่องคอมพิวเตอร์ของผู้เรียนในห้องเรียนคอมพิวเตอร์ โดยกดปุ่มขวาเม้าส์ที่ชื่อเครื่องคอมพิวเตอร์ที่ต้องการของต้นไม้แสดงชื่อเครื่อง (ตามรูปที่ ข.3) จะปรากฏวินโดว์สำหรับใส่รหัสผ่านของเครื่องคอมพิวเตอร์ที่จะเฝ้าสังเกต ถ้าใส่รหัสผ่านถูกต้องจะปรากฏวินโดว์จอภาพที่ต้องการเฝ้าสังเกต



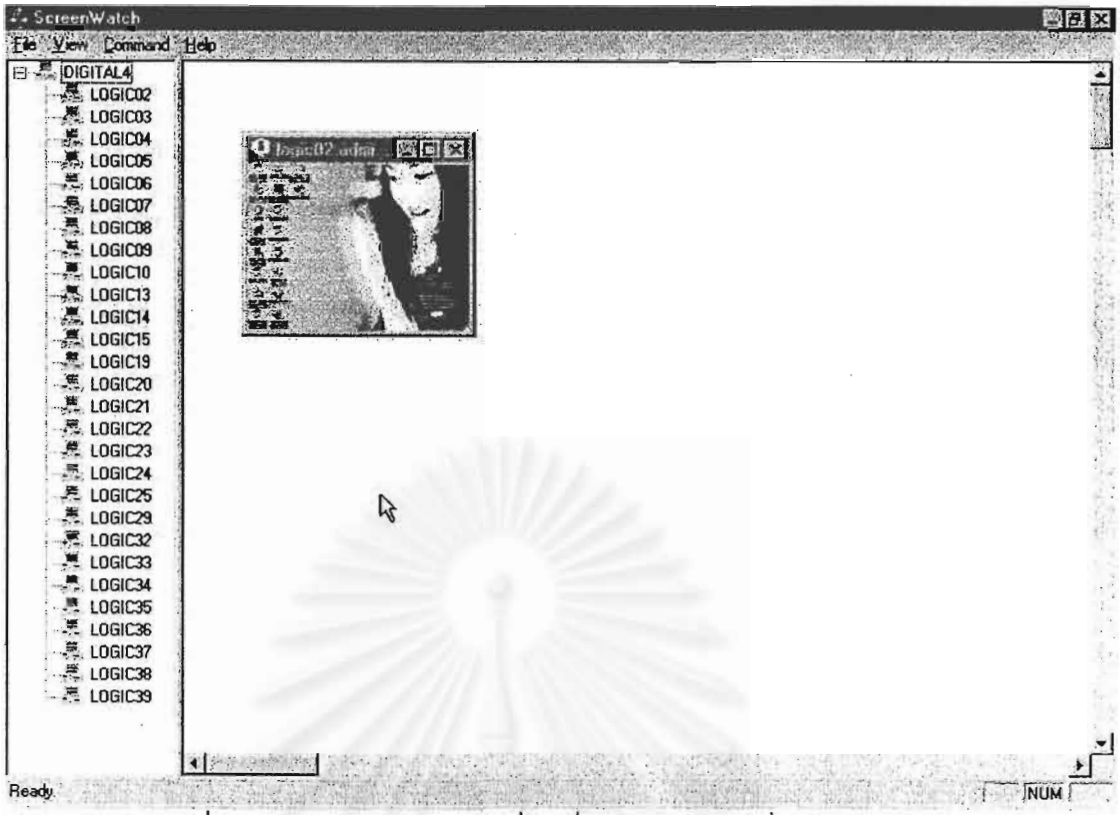
รูปที่ ข.2 แสดงการเพิ่มจอภาพที่ต้องการสังเกตโดยทราบชื่อเครื่องคอมพิวเตอร์ที่อยู่ในกลุ่มงาน



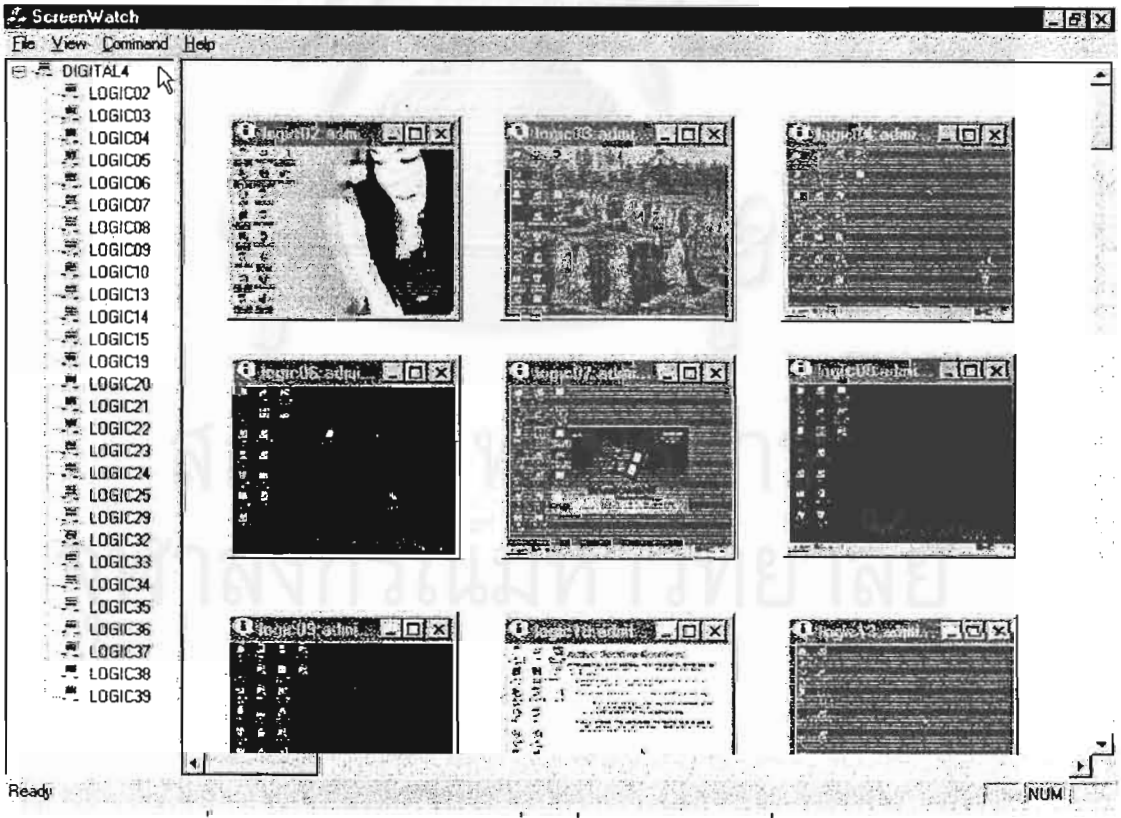
รูปที่ ข.3 แสดงการเพิ่มจอภาพที่ต้องการสังเกตโดยไม่ทราบชื่อเครื่องคอมพิวเตอร์ที่อยู่ในกลุ่มงาน

2. การแสดงจอภาพที่ต้องการสังเกต

หลังจากที่เลือกเมนูเพิ่มจอภาพที่ต้องการสังเกตแล้ว จะปรากฏจอภาพของเครื่องคอมพิวเตอร์ผู้เรียนในอัตราส่วน 1:5 ของจอภาพที่ต้องการสังเกตจริง ที่ส่วนหัวของวินโดว์ที่สังเกตจะปรากฏชื่อเครื่องคอมพิวเตอร์ และชื่อผู้ใช้งานเครื่องคอมพิวเตอร์นั้น



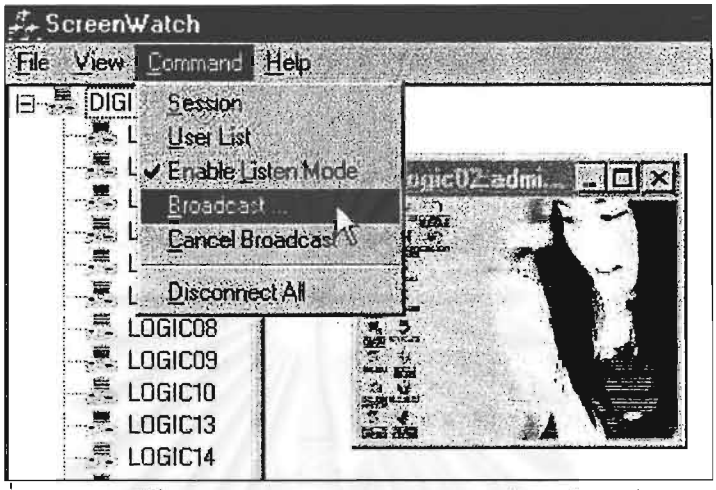
รูปที่ ข.4 แสดงจอภาพที่ต้องการสังเกตหลังจากใส่ชื่อเครื่องคอมพิวเตอร์ที่อยู่ในกลุ่มงานแล้ว



รูปที่ ข.5 แสดงจอภาพที่ต้องการสังเกตหลังจากใส่ชื่อเครื่องคอมพิวเตอร์ที่อยู่ในกลุ่มงานแล้ว หลายเครื่องพร้อมกัน

3. ส่วนการแสดงจอภาพของผู้สอนให้ผู้เรียนสังเกต

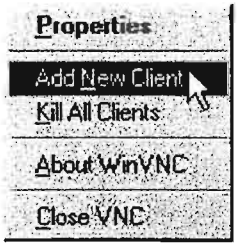
ถ้าต้องการแสดงจอภาพของผู้สอนให้ผู้เรียนสังเกตในกลุ่มงาน สามารถทำได้โดยการเลือกเมนู "Command" – "Broadcast" (ตามรูปที่ ข.6) จะปรากฏสัญรูป (icon) ของโปรแกรมอนุญาตให้มีการสังเกตจากผู้เรียนที่บริเวณทางขวาของทาสก์บาร์ (task bar)



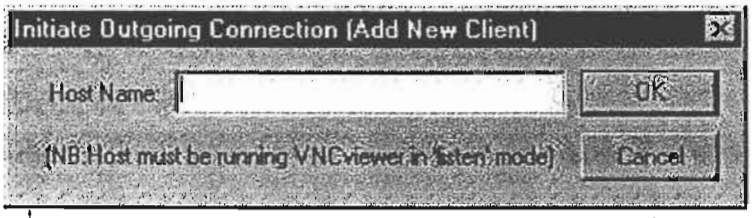
รูปที่ ข.6 แสดงวิธีอนุญาตการแสดงจอภาพของผู้สอนให้ผู้เรียนสังเกต

รูปที่ ข.7 แสดงสัญรูปแสดงจอภาพของผู้สอนให้ผู้เรียนสังเกต(สีเขียว)บริเวณทาสก์บาร์

ถ้าต้องการแสดงจอภาพของผู้สอนให้ผู้เรียนสังเกตในกลุ่มงาน ขณะนั้นสามารถทำได้โดยกดปุ่มขวาเม้าส์ที่สัญรูป (icon) ของโปรแกรมอนุญาตให้มีการสังเกตจากผู้เรียน (มีสีเขียว) ที่บริเวณทางขวาของทาสก์บาร์ (task bar) แล้วเลือก "Add New Client" จากเมนูที่ปรากฏคำโต้ตอบ (dialog) ให้ใส่ชื่อเครื่องคอมพิวเตอร์ที่อยู่ในกลุ่มงานที่ต้องการระบุเป็นเครื่องปลายทาง โดยใช้ "," เป็นเครื่องหมายคั่นระหว่าง 2 ชื่อเครื่องใด ๆ หรือถ้าต้องการแสดงจอภาพของผู้สอนให้ผู้เรียนสังเกตในกลุ่มงานทั้งหมด สามารถทำได้โดยใส่คำ "all" ไปในคำโต้ตอบ ถ้าการแสดงจอภาพของผู้สอนให้ผู้เรียนสังเกตในกลุ่มงานสำเร็จสัญรูปของโปรแกรมอนุญาตให้มีการสังเกตจากผู้เรียนจะเปลี่ยนเป็นสีดำ



รูปที่ ข.8 แสดงการแสดงจอภาพของผู้สอนให้ผู้เรียนสังเกต

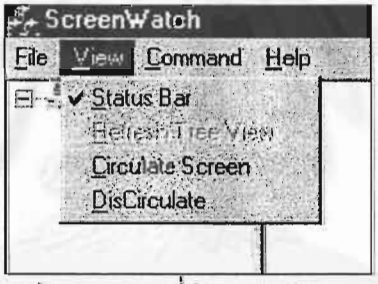


รูปที่ ข.9 แสดงคำโต้ตอบก่อนแสดงจอภาพของผู้สอนให้ผู้เรียนสังเกต

4. ส่วนตัวเลือกของระบบ

4.1. การเวียนจอภาพที่ต้องการสังเกตและยกเลิกการเวียน

ถ้าต้องการเวียนจอภาพที่ต้องการสังเกตจากหน้าจอเครื่องคอมพิวเตอร์ที่เฝ้าสังเกตอยู่แล้ว สามารถทำได้โดยเลือกเมนู "View" – "Circulate Screen" จะปรากฏหน้าจอที่จะเฝ้าสังเกตทีละ 1 หน้าจอในอัตราส่วน 1:6 ของจอภาพที่ต้องการสังเกตจริงที่ส่วนแสดงจอภาพที่ต้องการสังเกตสลับกันแสดงไป ถ้าต้องการยกเลิกการเวียน สามารถทำได้โดยเลือกเมนู "View" – "DisCirculate"



รูปที่ ข.10 แสดงการเวียนจอภาพที่ต้องการสังเกตและยกเลิกการเวียน

4.2. การอนุญาตให้มีการส่งจอภาพจากผู้เรียนมาปรากฏที่ผู้สอน

ถ้าต้องการอนุญาตให้มีการส่งจอภาพจากผู้เรียนมาปรากฏที่ผู้สอน สามารถทำได้โดยเลือกเมนู "Command" – "Enable Listen Mode" จะปรากฏสัญรูป(สีแดง) ขึ้นที่บริเวณทางขวาของทาสก์บาร์ ถ้าทำการเลือกซ้ำจะทำให้สัญรูป(สีแดง)นั้นหายไป และเป็นการไม่อนุญาตให้มีการส่งจอภาพจากผู้เรียนมาปรากฏที่ผู้สอน

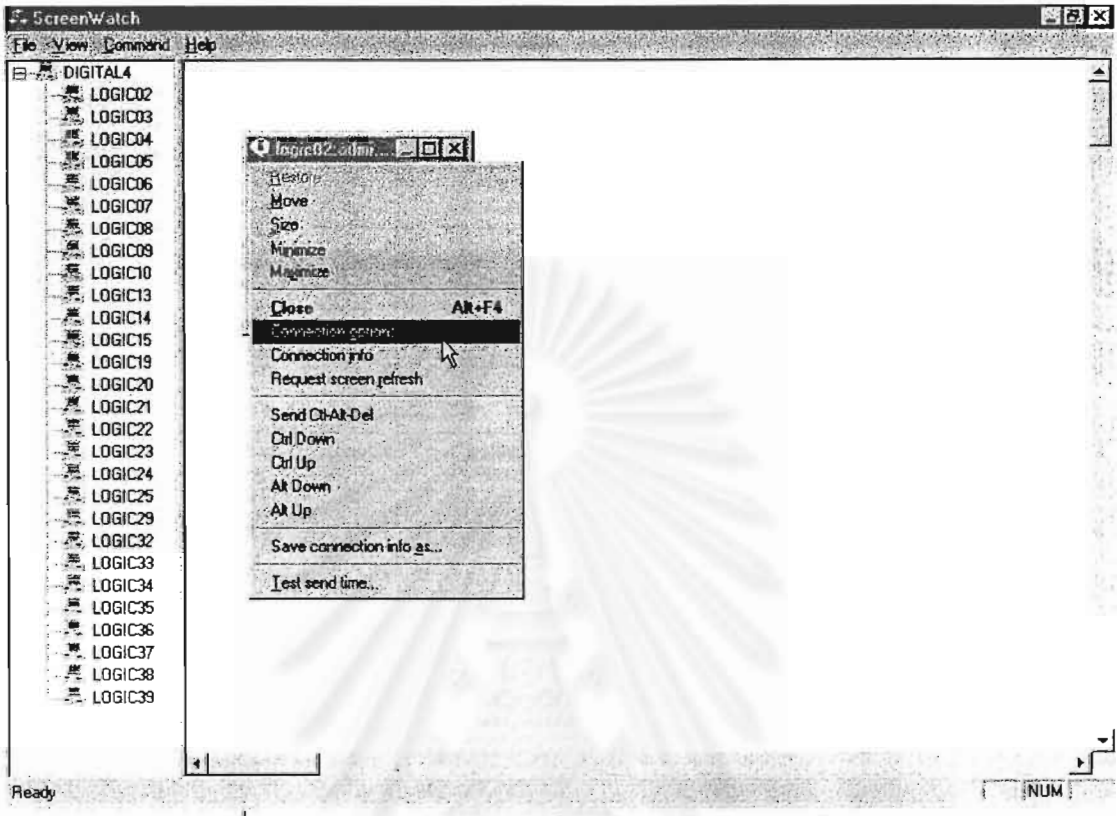
4.3. การตรวจสอบรายชื่อผู้เรียน (รายชื่อผู้ใช้งาน)

ถ้าต้องการตรวจสอบรายชื่อผู้เรียน (รายชื่อผู้ใช้งาน) สามารถทำได้โดยเลือกเมนู "Command" – "User List" จะปรากฏข้อความแสดงผู้ใช้งานของแต่ละเครื่อง

4.4. ตัวเลือกของระบบอื่น ๆ

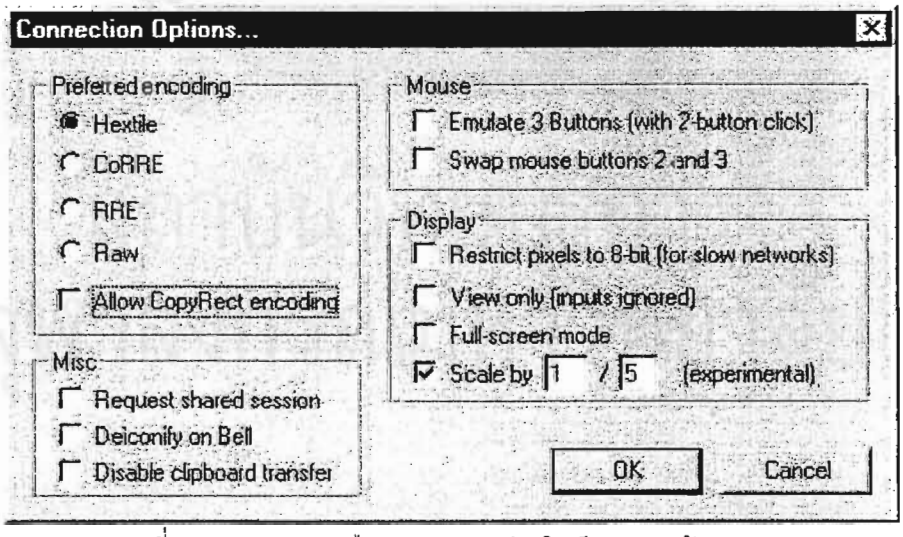
4.4.1. ตัวเลือกของวินโดว์จอภาพเฝ้าสังเกต

ถ้าต้องการเลือกตัวเลือกของแต่ละวินโดว์จอภาพเฝ้าสังเกต สามารถทำได้โดยเลือกเมนูที่มุมบนซ้ายของแต่ละวินโดว์ โดยกดปุ่มขวาเม้าส์ที่สัญรูปที่กรอบวินโดว์แล้วเลือก "Connection option..." (ตามรูปที่ ข.11)



รูปที่ ข.11 แสดงรูปก่อนเข้าตัวเลือกของวินโดว์จอภาพเฝ้าสังเกต

สำหรับรายละเอียดตัวเลือกของวินโดว์ และความหมายสามารถศึกษาได้ในภาคผนวก ก

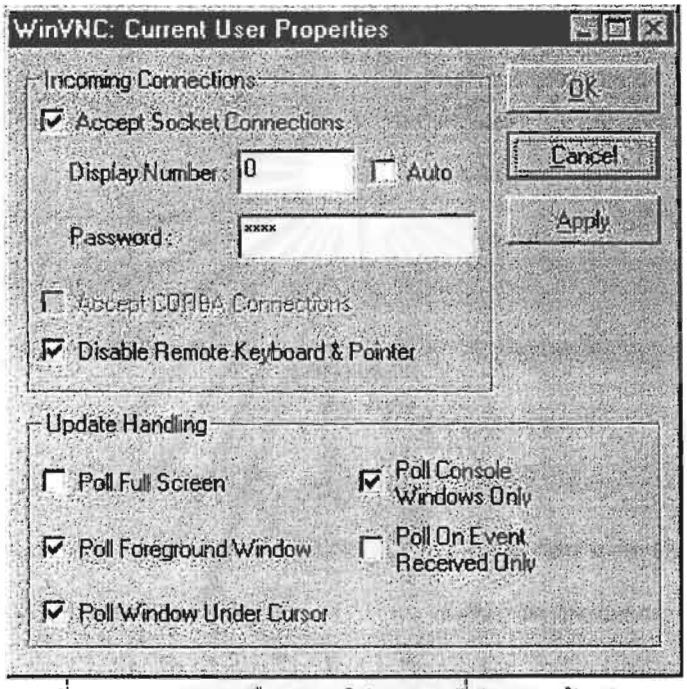


รูปที่ ข.12 แสดงตัวเลือกของแต่ละวินโดว์จอภาพเฝ้าสังเกต

4.4.2. ตัวเลือกของโปรแกรมที่ฝ่ายถูกเฝ้าสังเกต

ในการใช้งานจริง อาจมีการใช้งานในระบบเครือข่ายคับคั่ง หรือต้องการประสิทธิภาพของคอมพิวเตอร์ผู้เรียนที่ดี การเปลี่ยนแปลงค่าตัวเลือกของฝ่ายถูกเฝ้าสังเกต จะสามารถทำได้ ถ้าต้องการตั้งค่าตัวเลือกที่ฝ่ายถูกเฝ้าสังเกต สามารถทำได้โดยกดปุ่มขวาเม้าส์ที่สัญรูป (สีเขียว) เลือกเมนู Properties จะปรากฏวินโดว์ตัวเลือกของโปรแกรมที่ฝ่ายถูกเฝ้าสังเกต (ตามรูปที่ ข.13)

สำหรับรายละเอียดตัวเลือกของวินโดว์ และความหมายสามารถศึกษาได้ในภาคผนวก ก



รูปที่ ข.13 แสดงตัวเลือกของโปรแกรมที่ฝ่ายถูกเฝ้าสังเกต

จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้วิจัย

นายกิติพงษ์ ตันโสภณธนาศักดิ์ เกิดวันที่ 11 มกราคม 2517 ที่อำเภอท่ามะกา จังหวัดกาญจนบุรี สำเร็จการศึกษาปริญญาตรีวิทยาศาสตร์บัณฑิต สาขา วิทยาการคอมพิวเตอร์ จากภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2538 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตร์มหาบัณฑิตที่จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2540

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

จุฬาลงกรณ์มหาวิทยาลัย