เกมอักษรไขว้

Scrabble Game

โดย

นาย โฆษิต เหล่าสุวรรณ

นาย ปกรณ์ วัฒนจตุรพร

นาย วิฑูรย์ กัลยาณวัฒน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2536

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการใช้

033144

ปริญญานิพนธ์ปีการศึกษา 2536

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เกมอักษรไขว้ (Scrabble game)

ผู้จัดทำ

1.	นาย โมษิต	เหล่าสุวรรณ	เลขประจำตัว	33100044
2.	นาย ปกรณ์	วัฒนจตุรพร	เลขประจำตัว	33100195
3.	นาย วิฑูรย์	กัลยาณวัฒน์	เลขประจำตัว	33100346

..... อาจารย์ที่ปรึกษา

(อาจารย์ วิบูลย์ พร้อมพานิชย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกมอักษรไขว้

Scrabble Game

โดย นาย โฆษิต เหล่าสุวรรณ
นาย ปกรณ์ วัฒนจตุรพร
นาย วิฑูรย์ กัลยาณวัฒน์

อาจารย์ที่ปรึกษา อ.วิบูลย์ พร้อมพานิชย์

บทคัดย่อ

การพัฒนาโครงงานนี้ขึ้น เพื่อศึกษาและพัฒนาขั้นตอนการทำงานของโปรแกรม การจัดการข้อมูลขนาดใหญ่ การจัดการการแสดงผล การติดต่อสื่อสารของเครื่องคอมพิวเตอร์ โดยใช้กรณีศึกษาคือ เกมอักษรไขว้

เนื่องจากการเล่นเกมอักษรไขว้นั้นจะต้องใช้ความสามารถในการเลือกคำศัพท์จากตัวอักษรที่มีอยู่ โดยคำนึงถึงคะแนนที่จะได้รับในการวางแต่ละครั้ง และ เวลาที่ใช้ไป เพื่อให้ได้ประสิทธิภาพการทำงานที่ดีจะต้องมีการจัดการข้อมูลที่ดีซึ่งก็คือคำภาษาอังกฤษจากพจนานุกรม รวมทั้งความสามารถในการใช้ข้อมูลเหล่านั้น

ABSTRACT

This project is developed for study and developing programming algorithms, data management, computer graphic display and computer-to-computer communication by using case study, Scrabble game.

The player must use their ability to build vocabulary word from their characters. Each turn, the player must concern about score and time in use. With good data management and high performance algorithm, this game will be the good application.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทที่ 1	บทนำ	1
1.1	ความหมาย	1
1.2	จุดมุ่งหมายของการพัฒนาโปรแกรม	1
1.3	อุปกรณ์การเล่น	1
1.4	กฎวิธีการเล่น	2
1.5	การคิดคะแนน	2
1.6	การสิ้นสุดเกม	3
1.7	ส่วนพิเศษในการเล่น	4
บทที่ 2	การออกแบบโปรแกรม	5
2.1	การจัดเก็บและค้นหาศัพท์	5
2.2	การจัดสรรหน่วยความจำขนาดใหญ่	12
2.3	การจัดการแสดงผลกราฟิค	12
2.4	การติดต่อสื่อสารระหว่างเครื่อง	12
บทที่ 3	การจัดการหน่วยความจำ	14
3.1	ไมโครโปรเซสเซอร์ 80286	14
3.2	ไมโครโปรเซสเซอร์ 80386	16
3.3	การจัดการหน่วยความจำในเครื่องพีซี	16
3.4	หน่วยความจำขยาย (Expanded memory)	17
3.5	หน่วยความจำยืดขยาย (Extended memory)	17
3.6	โหมดป้องกัน (Protected mode)	18
3.7	วิธีการเรียกใช้หน่วยความจำยืดขยายแต่ละแบบ	18
3.8	การติดต่อกับไดรเวอร์	22
3.9	ข้อควรระวังในการเรียกใช้ XMS	22
3.10	รายละเอียดของฟังก์ชันของ XMS	23
บทที่ 4	การจัดการจอภาพ	24
4.1	โหมดของจอภาพ	24
4.2	ประวัติการพัฒนาการแสดงผลบนจอคอมพิวเตอร์	25
4.3	การจัดการโหมดภาพในภาษาซี	26
4.4	การจัดการโหมดภาพในโปรแกรม	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5	ไลบารี GUI	27
4.6	อัลกอริทึมและความหมายของฟังก์ชันที่อยู่ในไลบารี่	28
บทที่ 5	การติดต่อสื่อสาร	34
5.1	แพคเกจข่าวสาร	34
5.2	ส่วนประกอบของระบบเครือข่ายคอมพิวเตอร์ระดับท้องถิ่น	36
5.3	การติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ กับ เครื่องคอมพิวเตอร์	40
5.4	โนเวลล์เนตแวร์ 286	41
5.5	โนเวลล์เนตแวร์ 386	42
5.6	เดต้าแกรมและเซสชัน	42
5.7	IPX (Internetwork Packet Exchange)	43
5.8	SPX (Sequenced Packet Exchange)	46
บทที่ 6	วิจารณ์และสรุปผล	50
6.1	ปัญหาในการทำงาน	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความหมาย

ครอสเวร์ดเกมเป็นเกมกีฬาประเภทหนึ่งที่อาศัยความรู้ภาษาอังกฤษในการเล่นโดยเล่นตั้งแต่ 2 คนขึ้นไป การเล่นเริ่มโดยผู้เล่นต้องประกอบตัวอักษรให้เป็นคำในภาษาอังกฤษลงในกระดานซึ่งจะได้คะแนนแตกต่างกันไปตามตัวอักษรและช่องคะแนนพิเศษต่าง ๆ ในตาราง ผู้ที่มีคะแนนสูงสุดเมื่อจบเกมเป็นผู้ชนะ

1.2 จุดมุ่งหมายของการพัฒนาโปรแกรม

ในการพัฒนาโครงงานนี้ขึ้นมานั้น เพื่อศึกษาและพัฒนาขั้นตอนการทำงานของโปรแกรม การจัดการข้อมูลที่มีขนาดใหญ่ การติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ การจัดการการแสดงผล

1.3 อุปกรณ์การเล่น

1.กระดาน ประกอบด้วยช่องต่าง ๆ รวม 225 ช่อง กว้าง 15 แถว ยาว 15 แถว แบ่งเป็น คะแนนธรรมดา 164 ช่อง (สีเขียว) และช่องคะแนนพิเศษ 61 ช่อง (สีแดง ชมพู น้ำเงิน และ ฟ้า) ซึ่งมีความหมายแตกต่างกัน

2.ตัวเบี้ยอักษร (Tiles) ตัวเบี้ยอักษรครอสเวิร์ดเกมจะมีตั้งแต่ ตัว A ถึงตัว Z และตัวแบลงค์ (Blank) รวมทั้งสิ้น 100 ตัว ตัวอักษรแต่ละตัวจะมีค่าคะแนนและจำนวนแตกต่างกันไปตามความเหมาะสมและสมดุล เช่น ตัวอักษรที่เล่นง่ายใช้ประจำจะมีมากกว่าตัวอักษรที่เล่นยาก

3.แป้นวางตัวอักษร(Rack)/ถุงใส่ตัวอักษร/กระดาษจดคะแนนและบันทึกตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 กฎวิธีการเล่น

1.ผู้เล่นทั้งสองฝ่ายจะต้องเตรียมอุปกรณ์การเล่นทั้งหมดให้พร้อมกระดานแป้นวางตัวอักษร กระดาษจดคะแนนตรวจนับเบี้ยตัวอักษรครบ 100 ตัว และตั้งเวลา 22 นาที นาฬิกาจับเวลา (Chess Clock) ถ้ามี

2.ผู้เล่นจะต้องจับตัวอักษรขึ้นมาฝ่ายละ 1 ตัว เพื่อดูว่าฝ่ายไหนจะเป็นผู้เริ่มเล่นก่อน โดยมีหลักคือ เรียงตามลำดับตัวอักษรเริ่มจาก Blank, A, B, C ไปจนถึง Z ใครได้ใกล่กว่าจะได้เป็นผู้เริ่มเล่นเกมก่อน

3.ผู้เล่นจับตัวอักษรขึ้นมาฝ่ายละ 7 ตัว วางบนแป้น (Rack) โดยผู้ได้เริ่มเล่นก่อนจับก่อน

4.ผู้เริ่มเล่นก่อนจะต้องผสมจัดตัวอักษรตั้งแต่ 2 ตัวขึ้นไป ให้เป็นคำในภาษาอังกฤษ (หมายถึง คำที่ปรากฏอยู่ใน The Standard Dictionary for Crossword Game Players (Tournament Edition)) ลงบนกระดานในแนวนอนหรือแนวตั้ง โดยมีตัวอักษรตัวใดตัวหนึ่งของคำทับอยู่บนกลางกระดาน และคำที่ลงครั้งแรกนี้จะได้คะแนนเป็น 2 เท่า เพราะช่องดาวบนกระดานก็คือช่องสีชมพูนั่นเอง

5.ผู้เล่นคนแรกจะต้องจัดตัวอักษรจากในถุงขึ้นมาแทนใหม่ เท่ากับจำนวนตัวอักษรที่ใช้ไป จากนั้นจะเป็นตาเล่นของผู้เล่นคนที่สอง ซึ่งจะต้องต่ออักษรที่มีอยู่ให้เป็นคำ โดยมีอย่างน้อยหนึ่งตัวสัมผัสกับตัวอักษรที่มีอยู่ในกระดานแล้ว จากนั้นก็จะคิดคะแนนตามคำที่เกิดขึ้นมาทุกคำจากการเล่นในตานั้น ผู้เล่นต้องลงตัวอักษรในแนวใดแนวเดียวติดต่อกันเท่านั้น คำ ๆ ใหม่ อาจเกิดขึ้นโดย

- ผสมตัวอักษรหนึ่งตัวหรือมากกว่ากับคำหรือตัวอักษรที่มีอยู่ในกระดานแล้ว
- ใส่คำในแนวตั้งกับคำที่มีอยู่แล้วในกระดาน โดยมีตัวอักษรตัวใดตัวหนึ่งของคำใหม่สัมผัสหรือเพิ่มเข้าไปในคำที่มีอยู่บนกระดาน
- ใส่คำที่สมบูรณ์แล้วในแนวนอนขนาน โดยตัวอักษรในคำใหม่ก่อให้เกิดคำใหม่อื่น ๆ ขึ้นมาอีก

1.5 การคิดและทำคะแนน

1.คะแนนที่จะทำได้เกิดจากคำที่เกิดขึ้นใหม่ทุกคำในการเล่นในตานั้น อาจจะเป็นการเปลี่ยนรูปให้เป็นพหูพจน์โดยใส่ S หรือทำให้เป็นขั้นกว่าหรือคำในรูปอดีตกาล (Past Tense) ย่อมถือเป็นคำที่เกิดขึ้นใหม่ทั้งนั้น คะแนนแต่ละครั้งจะเท่ากับค่าของตัวอักษรที่เล่นในคำที่สร้างขึ้นใหม่ ประกอบกับช่องพิเศษต่าง ๆ จากตัวอักษรที่วางทับช่อง ตัวอย่างเช่น เมื่อผู้เล่นคนแรกลงคำ PLAY ในแนวนอน ผู้เล่นอีกคนเล่นคำ RAYS ในแนวตั้ง โดยมีตัว S ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อท้ายคำแรกเป็น PLAYS ดังนั้นคะแนนที่คิดจะคิดจากทั้งสองค่า RAYS และ PLAYS รวมกัน

2. จากช่องคะแนนพิเศษ 4 แบบคือช่องสีฟ้า (คูณ 2 เฉพาะตัวอักษร) สีน้ำเงิน (คูณ 3 เฉพาะตัวอักษร) สีชมพู (คูณ 2 ทั้งคำ) สีแดง (คูณ 3 ทั้งคำ) หากเกิดกรณีที่ผู้เล่นลงคำซึ่งมีตัวอักษรที่เล่นใหม่ทับช่องพิเศษมากกว่า 1 ช่องแล้ว คะแนนที่ได้จะนับคะแนนพิเศษให้ตัวอักษร แล้วค่อยนำมารวม สำหรับคะแนนพิเศษของคำ ตัวอย่างเช่น หากผู้เล่นฝ่ายหนึ่งลงคำว่า CLAPS โดยมีตัว C ซึ่งมีค่า 3 แต้ม ทับบนช่องสีน้ำเงิน (คูณ 3 เฉพาะตัวอักษร) และตัว S ซึ่งมีค่า 1 แต้มทับช่องสีชมพู (คูณ 2 ทั้งคำ) คะแนนที่คิดออกมาได้จะเท่ากับ C(3*3) + L(1) + A(1) + P(3) + S(1) = 15 * 2 = 30 แต้ม

3.ช่องตัวอักษรพิเศษและช่องคำพิเศษนั้นสามารถใช้ได้ในการเล่นทับลงไปครั้งแรกเท่านั้นในการเล่นครั้งต่อมาตัวอักษรที่ทับอยู่ในช่องคะแนนแล้วนั้น ให้นับเฉพาะค่าของตัวอักษรเท่านั้น ตัวอย่างเช่น ผู้เล่นคนแรกเล่นคำว่า MONKEY โดยมีตัว Y ทับช่องสีชมพู (Double Word Score) ไปแล้ว ผู้เล่นคนที่สองเล่นคำ CATS โดยมีตัว S ต่อท้ายเป็นคำ MONKEYS โดยตัว S ไม่ได้ทับช่องพิเศษอะไรเลย การคิดคะแนนจะคิดทั้งสองคำ แต่คำ MONKEYS จะได้เฉพาะค่าคะแนนในแต่ละตัวอักษรเท่านั้น

4.เมื่อตัวแบลงค์นั้นตกลงทับในช่องสีชมพู แดง ค่าของคำนั้นจะถูกคูณ 3 ถึงแม้ว่าตัวแบลงค์นั้น จะไม่มีค่าคะแนนก็ตาม

5.เมื่อคำ 2 คำขึ้นไปถูกสร้างในการเล่นครั้งเดียวกัน หากมีตัวอักษรตัวใดที่เป็นตัวประกอบทั้งสองคำ ทับลงช่องพิเศษ ตัวอักษรหรือคำที่เกิดจากตัวอักษรนั้น จะคำนวณคะแนนตามช่องพิเศษนั้นทั้งสองคำ ตัวอย่างเช่น ในกระดานมีคำว่า NOT ผู้เล่นอีกคนเล่นคำ BOX โดยไปพาดกับคำเก่า และก่อให้เกิดคำใหม่เพิ่มอีกสองคำคือ NO, OX โดยมีตัว X ทับลงช่องสีน้ำเงิน (Triple Letter Score) คะแนนจะคิดจากคำ NO, OX และ BOX โดยตัว X ที่ทับช่องพิเศษจะคิดได้ทั้งจากคำ OX และ BOX

1.6 การสิ้นสุดเกม

1.เกมจะสิ้นสุดเมื่อผู้เล่นคนใดคนหนึ่งใช้ตัวอักษรที่ตนมีอยู่จนหมด (หลังจากที่ตัวอักษรในถุงหมดแล้ว)

2.ในขณะเดียวกันผู้เล่นฝ่ายตรงข้ามจะยังคงมีตัวอักษรเหลืออยู่ให้หาคะแนนรวมของตัวอักษรนั้น แล้วคูณด้วย 2 นำไปบวกให้กับผู้เล่นที่เป็นคนลงตัวอักษรหมดก่อน (ยกเว้นแบลงค์ไม่ติดลบ)

3.ในกรณีที่ผู้เล่นทั้งสองฝ่ายไม่สามารถเล่นตัวอักษรที่เหลือในแป้นของเขาทั้งสองแล้ว และบอกผ่านฝ่ายละ 3 ครั้งติดต่อกัน (รวมเป็น 6 ครั้ง) หรือไม่มีการเปลี่ยนแปลงคะแนนของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งสองฝ่ายรวมกันทั้งสิ้น 6 ครั้งติดต่อกัน ก็ถือว่าเกมการแข่งขันสิ้นสุดลง การนับคะแนนจะทำโดยที่เอาคะแนนของตัวอักษรที่เหลืออยู่ในแป้นลบออกจากคะแนนของตนเองโดยไม่ต้องคูณสอง (คะแนนของแบลงค์เท่ากับศูนย์)

1.7 ส่วนพิเศษในการเล่น

1.การขอเปลี่ยนตัว ผู้เล่นสามารถขอเปลี่ยนตัวอักษรได้โดยต้องเสียตาเล่น 1 ตา การเปลี่ยนตัวอักษรสามารถเปลี่ยนได้ตั้งแต่ 1 - 7 ตัว แต่จะไม่สามารถเปลี่ยนตัวได้โดยเด็ดขาด เมื่อมีตัวอักษรในถุงเหลือไม่ถึง 7 ตัวไม่ว่าผู้เล่นจะขอเปลี่ยนกี่ตัวก็ตาม

2.การทำบิงโกครอสเวิร์ด (BINGO CROSSWORD) คะแนนพิเศษ 50 แต้ม ในระหว่างการเล่นถ้าหากผู้เล่นสามารถลงตัวอักษรทั้ง 7 ตัวในแป้นพร้อมกันทำให้เกิดคำขึ้นใหม่ในตาเล่นนั้น ๆ ไม่ว่าคำที่เกิดขึ้นใหม่จะประกอบด้วยตัวอักษรกี่ตัวก็ตาม ผู้เล่นจะได้คะแนนบวกพิเศษอีก 50 แต้ม นอกเหนือจากคะแนนที่ได้ตามปกติ

3.การขอชาเลนจ์ (Challenge) หากผู้เล่นฝ่ายหนึ่งฝ่ายใดลงศัพท์ที่ผู้เล่นอีกฝ่ายหนึ่งไม่แน่ใจว่าจะเป็นคำศัพท์ที่ถูกต้องหรือไม่ ผู้เล่นฝ่ายนั้นสามารถขอชาเลนจ์ เพื่อให้กรรมการทำการเปิดพจนานุกรมพิสูจน์ได้ ถ้าหากผลการชาเลนจ์ปรากฏว่าคำศัพท์นั้นมีจริง ผู้ที่ทำการขอชาเลนจ์จะเสียตาเล่นตานั้นหนึ่งตา แต่ถ้าหากคำศัพท์นั้นไม่มีอยู่ในพจนานุกรมผู้ที่ลงคำศัพท์นั้น จะต้องยกตัวอักษรที่เป็นคำนั้น ๆ ออกจากกระดานและคะแนนที่เกิดจากการเล่นครั้งนั้นเป็นศูนย์แต้มพร้อมกับการเสียตาเล่นนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
การออกแบบโปรแกรม

การออกแบบโปรแกรม

ในการพัฒนาโปรแกรมครอสเวร์ด เนื่องจากโปรแกรมครอสเวร์ด ต้องมีความสามารถ ในการ จำคำศัพท์จำนวนมากในภาษาอังกฤษ,การแสดงผลเพื่อให้ผู้เล่นสามารถเข้าใจได้ง่าย,การเล่นหลายคน ต้องที่ต้องการเล่นระหว่างเครื่อง ดังนั้นสิ่งที่จำเป็นต้องใช้ในการพัฒนาโปรแกรมคือ

- การจัดเก็บ และค้นหาคำศัพท์
- การจัดหน่วยสรรความจำขนาดใหญ่
- การจัดการแสดงผลในรูปแบบกราฟิค
- การติดต่อสื่อสารระหว่างเครื่อง

2.1 การจัดเก็บ และค้นหาคำศัพท์

สำหรับคำศัพท์ที่ต้องใช้ในโปรแกรมครอสเวร์ด เป็นคำศัพท์ที่มีความยาวไม่เกิน 15 ตัวอักษร เพราะกระดานที่ใช้การเล่น (สำหรับวางตัวอักษร) มีขนาด 15x15 ตัวอักษร ซึ่งคำศัพท์ดังกล่าวมีประมาณ 100,000 คำ

2.1.1 การจัดเก็บไฟล์คำศัพท์ในหน่วยความจำสำรอง (Disk)

การเก็บเป็นไฟล์ แยกตามตัวอักษรขึ้นต้นเพื่อให้ง่ายต่อการแก้ไขข้อมูล, เพิ่มข้อมูล, ลบข้อมูล แต่ละคำ จะแยกคำศัพท์ต่างๆ ด้วย รหัสขึ้นบรรทัดใหม่ เมื่อจะใช้งานก็อ่านคำศัพท์ต่างๆ ลงในหน่วยความจำ ส่วน XMS (Extened Memory Specification) ทั้งหมดเพียงครั้งแรกครั้งเดียว เพราะการติดต่อกับ หน่วยความจำสำรอง จะทำให้เสียเวลามาก

2.1.2 การอ่านศัพท์จากหน่วยความจำสำรองและเขียนลงในหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากการติดต่อกับหน่วยความจำสำรอง จะทำให้เสียเวลามาก เพราะฉะนั้นก็จะใช้หน่วยความจำ หลัก (Main memory) ,และ หน่วยความจำเสริม (XMS - Extened Memory Specification) โดยจะจอง หน่วยความจำหลักเท่ากับขนาดของตัวอักษร 1 กลุ่ม และ จอง หน่วยความจำส่วน XMS เท่ากับเนื้อที่ที่ ต้องการ ใช้ในการเก็บคำศัพท์ทั้งหมด

เมื่อจะใช้งานก็อ่านคำศัพท์ต่างๆ ลงในหน่วยความจำส่วน XMS ทั้งหมดเพียงครั้งแรกครั้งเดียว แล้วเลือกอ่านเฉพาะกลุ่มคำศัพท์ที่ต้องการค้นหาเพิ่มกลุ่มเดียวจากหน่วยความจำส่วน XMS ไปเขียนในหน่วยความจำหลัก เพราะคำสั่งในการค้นหาและอ้างถึงหน่วยความจำที่มีจะเป็นการติดต่อกับหน่วย ความจำหลัก และเพื่อประโยชน์ในการนำหน่วยความจำ ไปใช้ในส่วนอื่นต่อไป

ในการเก็บข้อมูลบนหน่วยความจำหลักจะใช้เนื้อที่สำหรับเก็บคำศัพท์ 1 คำเท่ากับ 16 ตัวอักษร เพราะคำศัพท์แต่ละคำจะมีความยาวไม่เกิน 15 ตัวอักษร และมีรหัสปิดท้ายตัวอักษรอีก 1 ตัว การเก็บโดยวิธีดังกล่าวนี้จะทำให้สามารถคำนวณตำแหน่งของคำศัพท์ต่างๆ ได้ เพื่อเป็นประโยชน์ในการเรียง แบบ Quick sort และค้นหาข้อมูล แบบ แบ่งครึ่งช่วง (Binary search)

2.1.3 การเรียงข้อมูล

การเรียงข้อมูลมีหลายวิธีแต่ละวิธีมีความเร็ว และความซับซ้อนในการทำงานต่างๆ กันไป

2.1.3.1 Quick sort

2.1.3.2 Bubble sort

2.1.3.3 Selection sort

2.1.3.4 Binary sort

2.1.3.1 Quick sort

Quick sort เป็นวิธีการเรียงข้อมูลแบบหนึ่ง มีหลักการทำงานคือ แบ่งข้อมูลออกเป็น 2 ส่วน แล้วส่งข้อมูลทั้ง 2 ส่วนเข้าไปเรียงอีก โดยเรียงเรียกโปรแกรมตัวเอง การแบ่งข้อมูลเป็น 2 ส่วนนั้นจะมี ข้อมูลตัวซ้ายสุดหรือข้อมูลตัวแรกสุดเป็นเกณฑ์ในการแบ่ง ข้อมูลทั้ง 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มที่แบ่งแล้วจะมีคุณสมบัติคือ ข้อมูลทางด้านซ้ายของตัวแบ่งจะมีค่าน้อยกว่าหรือเท่ากับตัวแบ่ง ส่วนข้อมูลอีกกลุ่มทางด้านขวาของ ตัวแบ่งจะมีค่ามากกว่า ตัวแบ่ง ในการหาจุดแบ่ง (pivot) จะใช้ตัวชี้ข้อมูล 2 ตัวตัวแรกเริ่มชี้ข้อมูลจาก ด้านซ้ายสุดแล้วค่อยเลื่อน ตัวชี้ข้อมูลมาทางด้านขวาเลือกข้อมูลที่มีค่ามากกว่าตัวแบ่ง ส่วนตัวชี้ข้อมูล อีกตัวก็จะเริ่มชี้ข้อมูลจากทางด้านขวาสุดแล้วค่อยเลื่อนมาทางด้านซ้ายเลือกข้อมูลที่มีค่าน้อยกว่าตัวแบ่ง สลับตำแหน่งข้อมูลทั้ง 2 ข้าง แล้วเลื่อนตัวชี้ข้อมูลทั้งสองตามเงื่อนไขไปเรื่อยจนตัวชี้ข้อมูลทั้งสองสวนทางกันก็จะได้จุดแบ่งข้อมูล เมื่อได้ข้อมูลแบ่งเป็น 2 กลุ่มแล้วก็ส่งข้อมูลไปเรียงตามวิธีดังกล่าวทีละกลุ่มทำไปเรื่อยๆ จนไม่สามารถแบ่งข้อมูลได้ก็จะได้ข้อมูลที่เรียงตามลำดับทั้งหมด

ตัวอย่าง ให้ 99>> คือ ตำแหน่งตัวชี้ข้อมูลด้านซ้าย, กำลังเลื่อนตัวชี้ข้อมูลทางด้านซ้าย

99> คือ ตำแหน่งตัวชี้ข้อมูลด้านซ้ายปัจจุบัน

<<99 คือ ตำแหน่งตัวชี้ข้อมูลด้านขวา, กำลังเลื่อนตัวชี้ข้อมูลทางด้านขวา

<99 คือ ตำแหน่งตัวชี้ข้อมูลด้านขวาปัจจุบัน

ข้อมูลเริ่ม	25>>	57	48	37	12	92	86	<33	
ตัวแบ่ง=25	25>>	57	48	37	12	92	86	<33	
	25	57>	48	37	12	92	86	<33	
	25	57>	48	37	12	92	86	<<33	
	25	57>	48	37	12	92	<<86	33	
	25	57>	48	37	12	<<92	86	33	
	25	57>	48	37	<12	92	86	33	
	25	12>	48	37	<57	92	86	33	สลับข้อมูล
	25	12>>	48	37	<57	92	86	33	
	25	12	48>	37	<<57	92	86	33	
	25	12	48>	<<37	57	92	86	33	
	25	12	<<48	37	57	92	86	33	
	25	<12	48>	37	57	92	86	33	
	12	<25	48>	37	57	92	86	33	สลับข้อมูล
	12	25	48	37	57	92	86	33	จุดแบ่งที่ 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.2 Bubble sort

Bubble sort เป็นการเรียงข้อมูลที่อาศัยหลักการสลับข้อมูลตัวที่มีค่ามากกว่าไปทางด้านขวาของตัวเอง การสลับจะเริ่มตรวจข้อมูลจากซ้ายสุดไปขวาสุด เมื่อพบข้อมูลตัวที่มากกว่าข้อมูลทางขวาก็สลับไปทางขวา การตรวจข้อมูลจะวนตรวจไปเรื่อยๆ โดยขอบเขตทางด้านขวาก็จะค่อยๆ ขยับเข้ามาทางซ้ายจนขอบเขตด้านขวาเลื่อนไปถึงขอบด้านซ้าย ก็จะเป็นการสิ้นสุดการเรียงข้อมูล ในระหว่างการตรวจข้อมูลแต่ ละรอบจะได้ข้อมูลที่มีค่ามากไปสุดไปไว้ด้านขวาสุดของขอบเขต

ข้อมูลเริ่มต้น	25	57	48	37	12	92	86	33
รอบที่ 1	25	48	37	12	57	86	33	92
รอบที่ 2	25	37	12	48	57	33	86	92
รอบที่ 3	25	12	37	48	33	57	86	92
รอบที่ 4	12	25	37	33	48	57	86	92
รอบที่ 5	12	25	33	37	48	57	86	92
รอบที่ 6	12	25	33	37	48	57	86	92
รอบที่ 7	12	25	33	37	48	57	86	92

2.1.3.3 Binary tree sort

Binary tree เป็นการนำเสนอข้อมูลที่เรียงตามลำดับรูปแบบหนึ่ง โดยใช้วิธี Binary tree sort ในการเรียงข้อมูล

วิธีการเรียงข้อมูลแบบนี้มีการทำงานคือ ตรวจข้อมูลใหม่แล้วนำไปหาตำแหน่งใส่ข้อมูลใหม่ลง ใน tree (แผนภาพข้อมูลรูปต้นไม้) ในการหาตำแหน่งของข้อมูลตัวใหม่ตำแหน่งที่ได้จะอยู่ข้างซ้ายหรือข้างขวา ของข้อมูลเดิม โดยมีเงื่อนไข ว่า

เมื่อข้อมูลตัวใหม่มากกว่าข้อมูลเดิม และตำแหน่งด้านขวาของข้อมูลเดิมเป็นตำแหน่งว่างก็จะใส่ข้อมูลใหม่ลงในตำแหน่งข้างขวาของข้อมูลเดิม แต่ถ้าข้างขวาของข้อมูลเดิมไม่ว่างก็ให้เปรียบเทียบกับ ข้อมูลด้านขวาของข้อมูลเดิมอีกครั้ง ทำเช่นนี้ไปเรื่อยๆ จนได้ตำแหน่งว่างใส่ข้อมูลใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อข้อมูลตัวใหม่น้อยกว่าข้อมูลเดิม และตำแหน่งด้านซ้ายของข้อมูลเดิมเป็นตำแหน่งว่างก็จะใส่ข้อมูลใหม่ลงในตำแหน่งข้างซ้ายของข้อมูลเดิม แต่ถ้าข้างซ้ายของข้อมูลเดิมไม่ว่างก็ให้เปรียบเทียบกับ ข้อมูลด้านซ้ายของข้อมูลเดิมอีกครั้ง ทำเช่นนี้ไปเรื่อยๆ จนได้ตำแหน่งว่างใส่ข้อมูลใหม่

กรณีที่เป็นข้อมูลตัวแรกก็จะ ใส่ข้อมูลนั้นเป็นตัวแรกสุด อยู่ในตำแหน่ง รากของแผนภาพต้นไม้ (root)

รูปที่ 2.1 แผนภาพข้อมูลรูปต้นไม้

แต่วิธีที่เลือกใช้ก็คือ Quick sort เพราะเป็นการเรียงที่เร็วที่สุดและยังมีคำสั่งในการสั่งเรียงโดยวิธีดังกล่าว ในภาษา C

2.1.4 การค้นหาคำศัพท์

ในการค้นหาข้อมูลเป็นการนำข้อมูลที่มีอยู่เป็นการนำข้อมูลที่มีอยู่มาใช้ประโยชน์ โดยข้อมูลที่นำมาใช้ในการค้นหาอาจจะอยู่ในรูปของ

2.1.4.1. ข้อมูลที่ยังเรียงตามลำดับแล้ว

สำหรับข้อมูลที่เรียงแล้ว ในการค้นหาข้อมูลจะใช้ประหยัดเวลากว่าการค้นหาบนข้อมูลที่ยังไม่ได้เรียงตามลำดับ แต่การที่ต้องเรียงข้อมูลก่อนจะทำให้เสียเวลามาก สำหรับกรณีที่ข้อมูลมีการเปลี่ยนแปลงต้องมีการเรียงข้อมูลใหม่อีกครั้ง ถ้าข้อมูลมีการเปลี่ยนแปลงบ่อยมากและข้อมูลมีจำนวนไม่มากเกินไปก็จะไม่เหมาะสมกับวิธีนี้ และการเรียงข้อมูลไว้เมื่อต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

033144

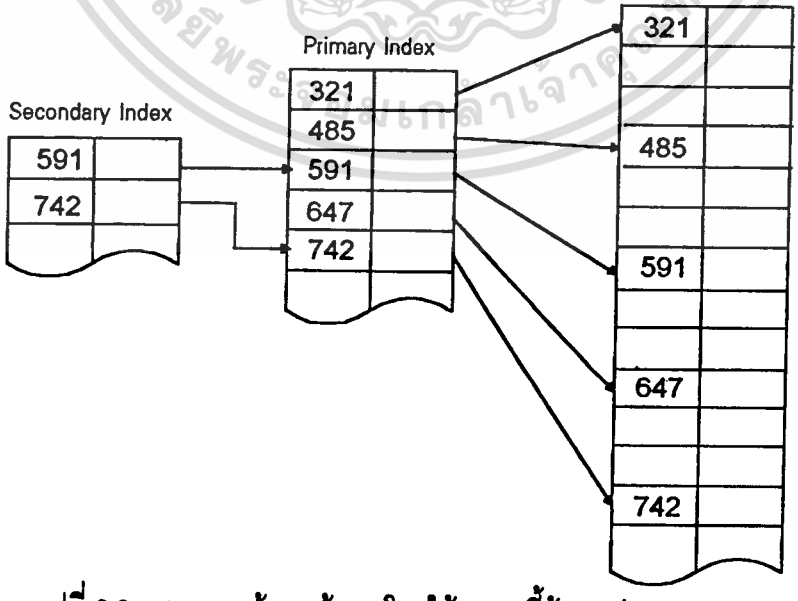
การค้นหาบางวิธีก็จะต้องใช้เนื้อที่ในการเก็บข้อมูลมากขึ้น และสำหรับการค้นหาข้อมูลที่มีเงื่อนไขประกอบด้วย ถ้าข้อมูลไม่ได้เรียงตามเงื่อนไขดังกล่าวก็จะทำให้ไม่สามารถค้นหาข้อมูลตามเงื่อนไขดังกล่าวได้ การที่มีเงื่อนไขหลายแบบต่างๆกันไป ก็เป็นการยากที่จะเรียงข้อมูลไว้ตามเงื่อนไขทุกรูปแบบ เพราะ การเรียงข้อมูล ผลของการเรียงข้อมูลจะให้ผลลัพธ์ 2 แบบคือ

1. เรียงข้อมูลแล้วเปลี่ยนลำดับข้อมูลที่ข้อมูลต้นฉบับเลย จะทำให้เรียงได้เพียงแบบเดียว แต่จะประหยัดเนื้อที่ในการเก็บข้อมูลมี่เรียงตามลำดับแล้ว

2.ให้ผลของการเรียงลำดับข้อมูลโดยใช้ตารางลำดับใหม่ชี้ไปที่ข้อมูลต้นฉบับเดิม วิธีนี้จะทำให้ เรียงข้อมูลตามเงื่อนไขได้มากกว่า 1 เงื่อนไข แต่จะต้องเสียเนื้อที่ในการเก็บผลของการเรียงแต่ละเงื่อนไข และเวลาต้องนำข้อมูลเรียงตามเงื่อนไขนั้นๆ ไปใช้ต้อง ใช้ทั้งข้อมูลต้นฉบับ และ ตารางลำดับข้อมูล

สำหรับวิธีการค้นหาข้อมูลบนข้อมูลที่เรียงตามลำดับแล้ว มีหลายแบบ เช่น

- Indexed Sequential Search (การค้นหาแบบเรียงตามลำดับโดยมีตัวชี้ข้อมูล) ใช้ตารางชี้ข้อมูล (Index table) ช่วยในการประมาณตำแหน่งของข้อมูลที่จะค้นหา หรือเป็นการช่วยกำหนดขอบเขตของข้อมูลที่จะค้นหาเพื่อลดเวลาในการค้นหาข้อมูลไม่ต้องไปตรวจสอบกับข้อมูลทั้งหมด



รูปที่ 2.2 แสดงการค้นหาข้อมูลโดยใช้ตารางชี้ข้อมูลช่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

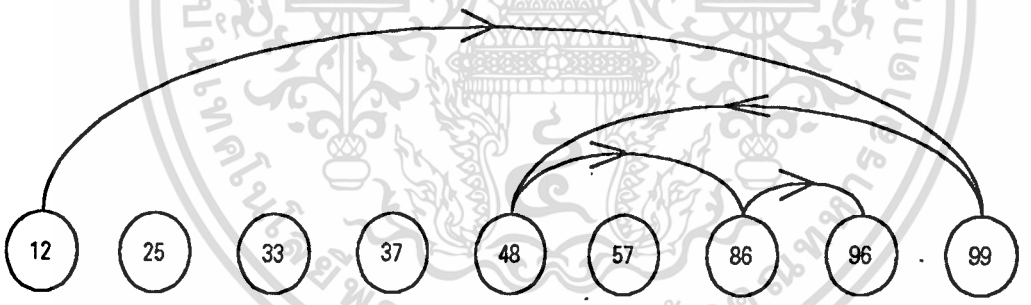
• Binary Search (การค้นแบบแบ่งครึ่งช่วง)

เป็นการค้นหาข้อมูลโดย คำนวณตำแหน่งข้อมูลที่จะค้นหาว่าจะต้องอยู่ระหว่างขอบเขตที่จะค้นหา แล้วกำหนดขอบเขตข้อมูลที่จะค้นหาทางด้านซ้าย และด้านขวา จะได้ขอบเขตของข้อมูลที่เล็กลงเรื่อยๆ จนกระทั่งพบข้อมูลที่ต้องการค้นหา โดยเริ่มต้นให้ขอบเขตข้อมูลด้านซ้ายคือข้อมูลตัวแรก และ ขอบเขตข้อมูลด้านขวาข้อมูลตัวสุดท้าย แล้วเอาข้อมูลที่ต้องการค้นหาเปรียบเทียบกับข้อมูลที่อยู่ระหว่างขอบเขตด้านซ้ายและด้านขวา และมีเงื่อนไขการปรับขอบเขตของข้อมูลที่จะค้นหาคือ

เมื่อข้อมูลที่ต้องการค้นหามากกว่าข้อมูลตรงกลางขอบเขต ให้ปรับขอบเขตด้านซ้ายมาเป็นตรงกลาง แต่เมื่อข้อมูลที่ต้องการค้นหาน้อยกว่าข้อมูลตรงกลางขอบเขต ให้ปรับขอบเขตด้านขวามาเป็นตรงกลาง ทำการเปรียบเทียบไปเรื่อยๆ จนกระทั่งข้อมูลที่ต้องการค้นหาเท่ากับข้อมูลตรงกลางขอบเขต

ข้อมูลที่เรียงแล้ว 12 25 33 37 48 57 86 96 99

เมื่อต้องการค้นหา 96 จากกลุ่มข้อมูล



รูปที่ 2.3 แสดงการค้นหาข้อมูลแบบแบ่งครึ่งช่วง

2.1.4.2 ข้อมูลที่ยังไม่เรียงตามลำดับ

สำหรับข้อมูลที่ยังไม่ได้เรียงวิธีการค้นหาข้อมูลการจะเป็นแบบ Sequential search (เริ่มหาตั้งแต่ข้อมูลตัวแรกไปจนถึงตัวสุดท้าย) จะทำให้ต้องเปรียบเทียบกับข้อมูลที่มีอยู่ทั้งหมด สำหรับข้อมูลที่มีปริมาณมากจะทำให้ ใช้เวลาในการค้นหานานมาก

ใช้วิธีแบ่งครึ่งช่วงโดยมีคำสั่ง Bsearch ของ C ให้บริการในการค้นหาโดยวิธีดังกล่าว ซึ่งการค้นหา ในคำสั่ง ดังกล่าวต้องใช้กับข้อมูลที่เก็บบนหน่วยความจำหลัก และขนาดของข้อมูล แต่ละคำต้องมีขนาดเท่ากัน (เนื้อ ที่จองสำหรับข้อมูลแต่ละคำมีขนาดเท่ากันหมด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การจัดสรรหน่วยความจำขนาดใหญ่

เนื่องจากคำศัพท์ที่ต้องอ้างถึงมีจำนวนมาก และต้องใช้เนื้อที่ในการเก็บมาก จึงไม่พอที่จะใช้หน่วยความจำหลักเพียงอย่างเดียว โดยหน่วยความจำที่ใช้ประกอบด้วยหน่วยความจำหลัก และ หน่วยความ จำเสริม ซึ่งจะกล่าวโดยละเอียดในบทที่ต่อไป

2.3 การจัดการแสดงผลในรูปแบบกราฟิค

สำหรับการแสดงผลในครอสเวร์ดเกมส์ ต้องใช้สีที่แตกต่างกันเพื่อแสดงช่องคะแนนพิเศษ และ คะแนนประจำตัวอักษรต่างๆ กวรเลือกใช้การแสดงผลในรูปแบบกราฟิค จึงเป็นวิธีการหนึ่งในการแก้ปัญหาดังกล่าว และยังสร้างความสวยงามให้เกมส์อีกด้วย

2.4 การติดต่อสื่อสารระหว่างเครื่อง

ในการเล่นเกมส์นี้ เมื่อต้องการเล่นหลายคน สามารถเล่นบนเครื่องเดียวกันหรือเล่นระหว่าง เครื่องก็ได้ แต่การเล่นบนเครื่องเดียวจะทำให้ผู้เล่นมองเห็นตัวอักษรในมือของคนอื่น ในการเล่นระหว่างเครื่อง อาศัยการติดต่อระหว่างเครื่องโดย

2.4.1 ระบบเครือข่ายคอมพิวเตอร์ขนาดท้องถิ่น (LAN : Local Area Network)

คือ การเชื่อมต่อเครื่องคอมพิวเตอร์ โดยที่เครื่องคอมพิวเตอร์แต่ละเครื่องสามารถสื่อสารกันได้โดยใช้ แพคเกจข่าวสาร ในระบบเครือข่ายคอมพิวเตอร์ขนาดท้องถิ่นจะต้องมี ไฟล์เซอร์เวอร์ (File Server) อย่างน้อย 1 เครื่อง การทำงานต่าง ๆ จะกระทำที่เครื่องคอมพิวเตอร์นั้น ๆ ไม่ใช่บนเครื่องไฟล์เซอร์เวอร์ และภายใต้ระบบเครือข่ายคอมพิวเตอร์ขนาดท้องถิ่นนั้นจะมีระบบปฏิบัติการที่ช่วยให้เครื่องคอมพิวเตอร์แต่ละเครื่องสามารถใช้แฟ้มข้อมูลร่วมกัน และการส่งข่าวสารถึงเครื่องคอมพิวเตอร์เครื่องอื่น ๆ

ระบบเครือข่ายคอมพิวเตอร์ขนาดท้องถิ่นเป็นกลุ่มของเครื่องคอมพิวเตอร์ที่เชื่อมต่อกับระบบ โดยใช้สายเคเบิล ในเครื่องคอมพิวเตอร์แต่ละเครื่องจะประกอบไปด้วย อุปกรณ์การเชื่อมต่อระบบเครือข่าย และ ซอร์ฟแวร์สำหรับระบบเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 การติดต่อแบบอนุกรม (Serial communication)

การส่งข้อมูลแบบอนุกรมเป็น การส่งข้อมูลที่ละบิต (bit) มีความเร็วค่อนข้างต่ำคืออยู่ในช่วงประมาณ 1200 บิตต่อวินาที ไปจนถึง 19200 บิตต่อวินาที หรือมากกว่านี้ ซึ่งจะเป็นการส่งระยะใกล้ แต่สำหรับอุปกรณ์ในการส่งข้อมูลแบบอนุกรมที่เครื่องคอมพิวเตอร์ทั่วไปมีให้ใช้จะมีความเร็วอยู่ในช่วง 1200 - 19200 เมื่อต้องการเชื่อมต่อระหว่าง 2 เครื่องก็จะใช้สายส่งข้อมูลแบบอนุกรม ต่อระหว่างพอร์ต (port) อนุกรมทั้งสองเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การจัดการหน่วยความจำ

การทำงานของระบบปฏิบัติการณ์ดอส (DOS : Disk Operating System) ถูกจำกัดการทำงานอยู่ภายในโหมดปกติ (Real Mode) ของสถาปัตยกรรมไมโครโปรเซสเซอร์ตระกูล 80x86 จากบริษัทอินเทล ไมโครโปรเซสเซอร์ 8086 สามารถทำงานได้ภายในโหมดปกติเท่านั้น คือ สามารถใช้หน่วยความจำแรมได้ 1 เมกกะไบต์ ในระยะแรกที่บริษัทไอบีเอ็ม (IBM : International Business Machine) ทำการผลิตเครื่องคอมพิวเตอร์ส่วนบุคคล (PC : Personal Computer) ขึ้นมานั้น ได้มีการจองหน่วยความจำขนาด 384 กิโลไบต์สำหรับอุปกรณ์ควบคุมการทำงานของเครื่อง เช่น หน่วยความจำสำหรับการแสดงผล และหน่วยความจำถาวรของไบออส (BIOS : Basic Input/Output System) และมีเนื้อที่เหลือสำหรับโปรแกรมของผู้ใช้งานอีก 640 กิโลไบต์

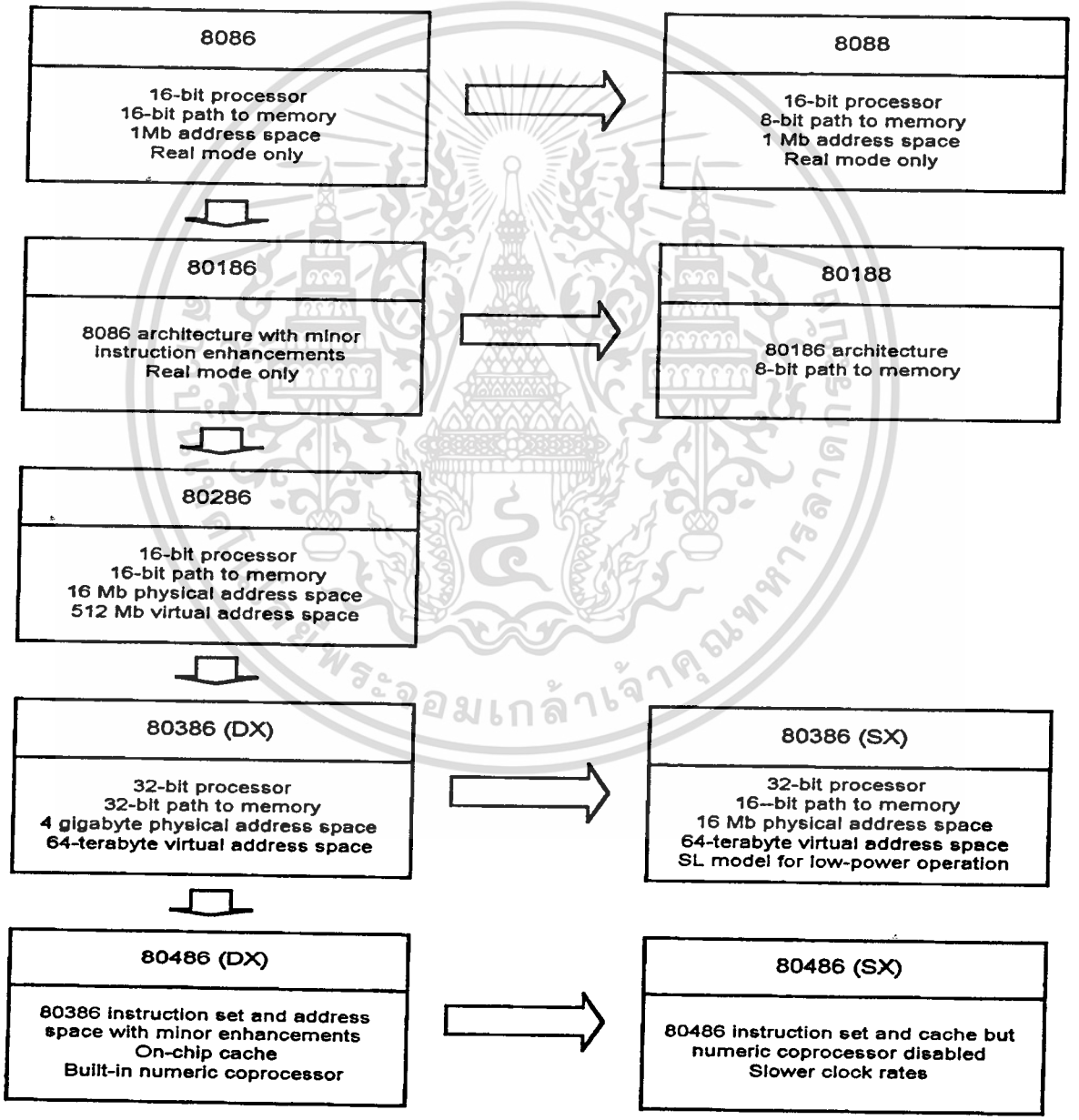
เมื่อมีการใช้เครื่องคอมพิวเตอร์ส่วนบุคคลมากขึ้น ผู้ใช้เริ่มมีการพัฒนาโปรแกรมแบบฝังตัวเองในหน่วยความจำ (TSRs : Terminate-and-Stay-Resident-programs) และมีการใช้เครื่องคอมพิวเตอร์สำหรับเก็บข้อมูลปริมาณมาก ๆ เช่นในโปรแกรมสเปรดชีต (Spreadsheet) ระบบจึงมีความต้องการหน่วยความจำมากขึ้น ระบบแรกที่มีการพัฒนาขึ้นคือ หน่วยความจำขยาย (EMS : Expanded Memory Specification) ระบบนี้จะมีการใช้เนื้อที่ส่วนละ 16 กิโลไบต์ในบริเวณ 384 กิโลไบต์สำหรับโปรแกรมของผู้ใช้ในการติดต่อ การใช้งานหน่วยความจำขยายนี้ใช้สำหรับการเก็บข้อมูล ไม่เหมาะสมสำหรับการเก็บโปรแกรม และ ไม่เหมาะสมกับการเก็บโปรแกรมแบบฝัง ข้อจำกัดอีกประการหนึ่งของการใช้หน่วยความจำแบบขยายคือ ผู้ใช้ไม่สามารถจองเนื้อที่ขนาดใหญ่มาก ๆ ได้ การใช้งานจะใช้ได้ครั้งละ 16 กิโลไบต์ โปรแกรมของผู้ใช้จะต้องเรียกเฟรมของข้อมูลมาใช้ให้ถูกต้อง

3.1 ไมโครโปรเซสเซอร์ 80286

ในช่วงที่บริษัทอินเทลได้ออกไมโครโปรเซสเซอร์ในสายการผลิตขึ้น คือ 80286 และจัดให้อยู่ในกลุ่มการใช้งานเครื่องคอมพิวเตอร์ระดับสูง (PC/AT : Personal Computer / Advanced Technology) มีการพัฒนาระบบการประมวลผลแบบใหม่เกิดขึ้นเรียกว่า โหมดการป้องกัน (Protected virtual address mode) ซึ่งยอมให้มีการใช้หน่วยความจำได้ถึง 16 เมกกะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไบต์ ในโหมดการป้องกันจะมีการตั้งสมมติฐานว่า โปรแกรมใช้งานจะไม่ทราบหน่วยความจำที่แท้จริง ระบบปฏิบัติการณ์จะเข้ามาทำหน้าที่ในการจัดการหน่วยความจำ โดยเมื่อมีการใช้งานหน่วยความจำ ระบบปฏิบัติการณ์จะทำหน้าที่โอนการเรียกใช้หน่วยความจำไปยังตำแหน่งที่ถูกต้อง แต่เพื่อให้การพัฒนาโปรแกรมสะดวกยิ่งขึ้นจึงมีการพัฒนาระบบปฏิบัติการณ์ขนาดเล็กขึ้น เมื่อโปรแกรมของผู้ใช้มีการเรียกใช้งานขึ้น ระบบปฏิบัติการณ์ขนาดเล็กจะโอนการทำงานไปยังดอสอีกทอดหนึ่ง



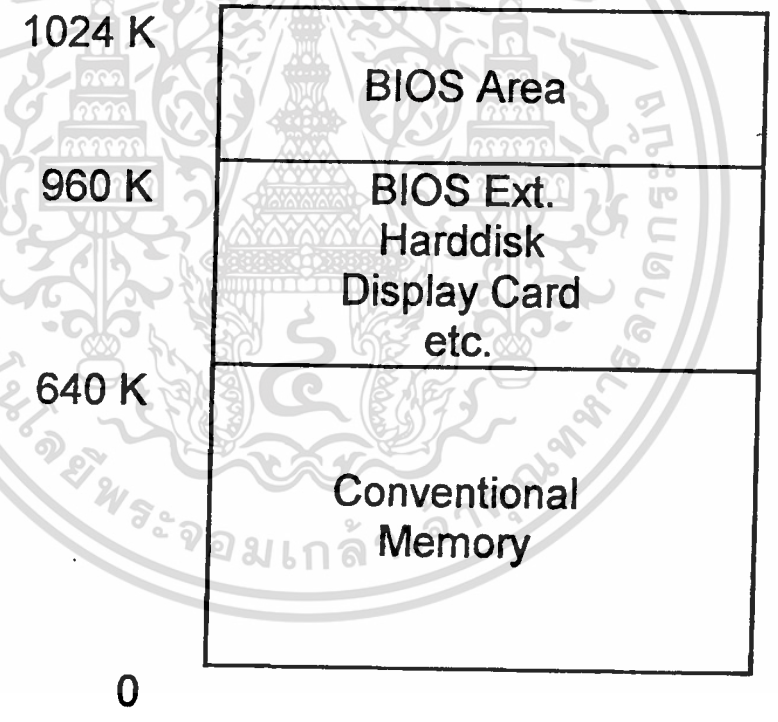
รูปที่ 3.1 แสดงพัฒนาการไมโครโปรเซสเซอร์ตระกูล 80x86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ไมโครโปรเซสเซอร์ 80386

ทางบริษัทอินเทลได้เพิ่มขีดความสามารถให้กับไมโครโปรเซสเซอร์ขึ้นอีก โดยเพิ่มความสามารถในการทำงานเป็นการทำงานแบบ 32 บิท และ การอ้างถึงหน่วยความจำขนาด 32 บิทหรือ 4 จิกะไบต์ รวมทั้งได้มีการความสามารถการเพจจิง (Paging) และ ระบบปฏิบัติการณ์แบบใหม่ขึ้น คือ V86 หรือ โหมดการทำงานเสมือน 8086 (Virtual 8086 mode) ความสามารถทั้งสองนี้มีเฉพาะในการทำงานระบบป้องกัน

3.3 การจัดการหน่วยความจำในเครื่องพีซี



รูปที่ 3.2 การจัดหน่วยความจำในเครื่องพีซี

การจัดหน่วยความจำตามรูปที่ 3.2 เป็นการจัดหน่วยความจำของเครื่องคอมพิวเตอร์ส่วนบุคคลทุกแบบตั้งแต่ 8086/8088, 80286, 80386 และ 80486 ที่ใช้ระบบปฏิบัติการดอส เครื่องคอมพิวเตอร์ส่วนบุคคลที่ใช้ไมโครโปรเซสเซอร์ 8086/8088 สามารถอ้างถึงหน่วยความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำได้เพียง 1 เมกกะไบต์เท่านั้น เพราะข้อจำกัดทางฮาร์ดแวร์คือ มีสายแอดเดรสเพียง 20 เส้น (2 ยกกำลัง 20 ได้ประมาณ 1 ล้าน) ดังนั้นดอสจึงสามารถจัดการหน่วยความจำได้เพียง 1 เมกกะไบต์และใน 1 เมกกะไบต์นี้ก็มีส่วนที่เป็นพื้นที่ใช้สอยเพียง 640 กิโลไบต์ ส่วนที่เหลือต้องสำรองไว้สำหรับอุปกรณ์อื่น ๆ และถึงแม้ว่าเครื่องที่ใช้ไมโครโปรเซสเซอร์ตั้งแต่ 80286 ขึ้นไปจะสามารถอ้างถึงหน่วยความจำได้มากขึ้นกว่า 1 เมกกะไบต์ก็ตาม ตัวดอสเองก็ยังคงสนับสนุนเฉพาะการทำงานในโหมดปกติที่มีลักษณะเหมือนกันกับ 8086/8088 คือ อ้างหน่วยความจำได้เพียง 1 เมกกะไบต์ จุดนี้เองที่เป็นปัญหาที่เกิดขึ้นในการพัฒนาซอฟต์แวร์บนระบบดอส วิธีการแก้ไขปัญหานี้เช่น สลับการทำงานไปทำงานในโหมดป้องกันซึ่งสามารถอ้างหน่วยความจำได้เต็มที่ แต่ก็มีปัญหาในการใช้ฟังก์ชันของดอส ซึ่งยังคงมีความจำเป็นในการพัฒนาโปรแกรมหากต้องเขียนฟังก์ชันต่าง ๆ เหล่านั้นขึ้นมาเองก็เปรียบเสมือนว่าเป็นการพัฒนาระบบปฏิบัติการณ์ขึ้นมาใหม่ทีเดียว

3.4 หน่วยความจำขยาย (Expanded Memory)

จากการความต้องการหน่วยความจำที่มากขึ้นบริษัท โลตัส อินเทล และ ไมโครซอฟต์ ได้ร่วมกันตั้งมาตราฐานสำหรับการติดต่อหน่วยความจำขยายขึ้นเรียกว่า EMS (Expanded Memory Specification) ซอร์ฟแวร์ที่ใช้ในการจัดการเรียกว่า โปรแกรมจัดการหน่วยความจำขยาย (Expanded Memory Manager) โปรแกรมจัดการจะจัดการเพจที่กำลังมีการใช้ให้สามารถใช้ได้ และ ทำการเก็บเพจที่ไม่มีการใช้แล้วไว้ การใช้งานหน่วยความจำขยาย โปรแกรมจะขอการใช้จากโปรแกรมจัดการโดยจะระบุถึงจำนวนที่ต้องการ และ เมื่อใช้งานเสร็จเรียบร้อยโปรแกรมจะต้องยกเลิกการใช้งานก่อนหยุดการทำงาน ข้อดีของระบบหน่วยความจำขยายคือ สามารถใช้ได้กับเครื่องคอมพิวเตอร์ส่วนบุคคลในทุก ๆ เครื่อง สำหรับเครื่อง 80386 และ 80486 นั้นไม่จำเป็นต้องมีการเพิ่มเติมอุปกรณ์ใด ๆ โดยจะมีซอฟต์แวร์ทำหน้าที่แทนอุปกรณ์นั้น ๆ

3.5 หน่วยความจำยืดขยาย (Extented Memory)

หน่วยความจำยืดขยาย เป็นหน่วยความจำที่เกินจาก 1 เมกกะไบต์ขึ้นไป หน่วยประมวลผลกลาง จะสามารถอ้างถึงได้เฉพาะการทำงานในโหมดป้องกันเท่านั้น ดังนั้นในการที่จะใช้งานหน่วยความจำในส่วนนี้จึงจำเป็นต้องสวิตช์ให้หน่วยประมวลผลกลางไปทำงานใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้งานหน่วยความจำในส่วนนี้จึงจำเป็นต้องสวิตช์ให้หน่วยประมวลผลกลางไปทำงานในโหมดป้องกันเสียก่อน เมื่อใช้เสร็จแล้วก็ต้องสวิตช์กลับมาที่โหมดปกติ เพราะข้อจำกัดของดอสที่จะทำงานเฉพาะในโหมดปกติเท่านั้น การสวิตช์ไปและกลับนั้นจะต้องระวังปัญหาที่จะตามมาอีก เช่น ต้องจัดการกับระบบของการอินเทอร์รัปต์ใหม่ทั้งหมด

3.6 โหมดป้องกัน (Protected Mode)

ซีพียูในตระกูล 80x86 ของอินเทล จะมีลักษณะการจัดการหน่วยความจำเหมือนกัน คือมอง หน่วยความจำเป็นไบต์ที่ต่อเนื่องกันไป โดยใน 8086/88 ซึ่งเป็นไมโครโปรเซสเซอร์ตัวแรกในตระกูลนี้ สามารถที่จะอ้างถึงหน่วยความจำได้เพียง 1 เมกกะไบต์เท่านั้น และมีสายแอดเดรสอยู่ 20 เส้นเรียกว่า A0 ถึง A19 เมื่อพัฒนามาถึง 286 ที่มีสายแอดเดรส 24 เส้น สามารถอ้างหน่วยความจำได้ 16 เมกกะไบต์ และ 386 อ้างได้ถึง 4096 เมกกะไบต์ หรือ 4 จิกะไบต์

เมื่อซีพียูตั้งแต่ 286 ขึ้นไปจะแบ่งการทำงานได้เป็น 2 โหมดหลัก ๆ คือ โหมดปกติ และ โหมดป้องกัน ในโหมดปกติซีพียูจะมีความสามารถในการอ้างถึงหน่วยความจำได้เท่ากับ 8086/8088 คือ อ้างได้ 1 เมกกะไบต์ ในโหมดปกติจะมีวิธีการอ้างแอดเดรสโดยการรวมกันของเซ็กเมนต์และออฟเซ็ต เพราะ รีจิสเตอร์มีขนาดเพียง 16 บิต แต่แอดเดรสมีขนาด 20 บิต เมื่อซีพียูทำงานในโหมดป้องกันก็จะสามารถใช้หน่วยความจำได้เต็มความสามารถที่มีอยู่ได้

3.7 วิธีการเรียกใช้หน่วยความจำยืดขยายแต่ละแบบ

วิธีการเข้าถึงหน่วยความจำยืดขยายที่จะกล่าวถึงต่อไปนี้ ไม่ใช่วิธีการที่จะเข้าถึงได้โดยไม่ต้องเข้าไปทำงานอยู่ในโหมดป้องกัน แต่เป็นการหลีกเลี่ยงไม่ให้โปรแกรมของเราต้องเป็นผู้จัดการกับการสวิตช์ไปและกลับจากโหมดป้องกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.1. เรียกใช้บริการของไบออส

ในไบออสของเครื่องระดับเอทีขึ้นไปจะมีฟังก์ชันบริการสำหรับการใช้หน่วยความจำยืดขยายอยู่ 3 ฟังก์ชัน ฟังก์ชันที่ 88h ใช้สำหรับขอทราบขนาดของหน่วยความจำยืดขยายที่มีอยู่ในระบบ ส่วนฟังก์ชันที่ 89h ใช้สำหรับผู้ที่ต้องการสวิตช์ไปทำงานในโหมดป้องกันเลย จะเห็นว่าไม่มีฟังก์ชันในการที่จะสวิตช์กลับมาในโหมดปกตินั้นหมายความว่ามันเป็นหน้าที่ของผู้ใช้ที่จะต้องทำถ้าคุณเลือกที่จะทำงานในโหมดป้องกัน ปกติจะไม่ใช้ฟังก์ชันนี้ ฟังก์ชัน 87h เป็นฟังก์ชันที่สำคัญที่สุดสำหรับการใช้หน่วยความจำยืดขยาย มีหน้าที่คัดลอกข้อมูลเป็นบล็อคจากหน่วยความจำที่หนึ่งไปยังอีกที่หนึ่งโดยตำแหน่งของหน่วยความจำที่ส่งไปให้กับฟังก์ชันที่ 87h นี้จะต้องเป็นแบบ ตำแหน่งเชิงเส้น ขนาด 24 บิต คือ เป็นแอดเดรสที่ใช้ในโหมดป้องกันไม่มีการมาบวกกันของเซ็กเมนต์และออฟเซ็ตแบบในโหมดปกติ (ฟังก์ชันนี้จะสวิตช์ไปทำงานในโหมดป้องกันแล้วทำการคัดลอกหน่วยความจำ จากนั้นก็สวิตช์กลับมาทำงานในโหมดปกติ)

วิธีการเรียกใช้ไบออสมีจุดอ่อนอยู่ที่ เมื่อมีหลายโปรแกรมที่ทำงานไปพร้อม ๆ กันและมีการเรียกใช้หน่วยความจำยืดขยาย โปรแกรมทั้งสองจะเกิดปัญหาเพราะไบออสจะไม่สนใจว่าหน่วยความจำส่วนไหนใครเป็นเจ้าของ

3.7.2. การเรียกใช้ผ่านไบออสโดยใช้ข้อมูลจากโปรแกรม VDISK

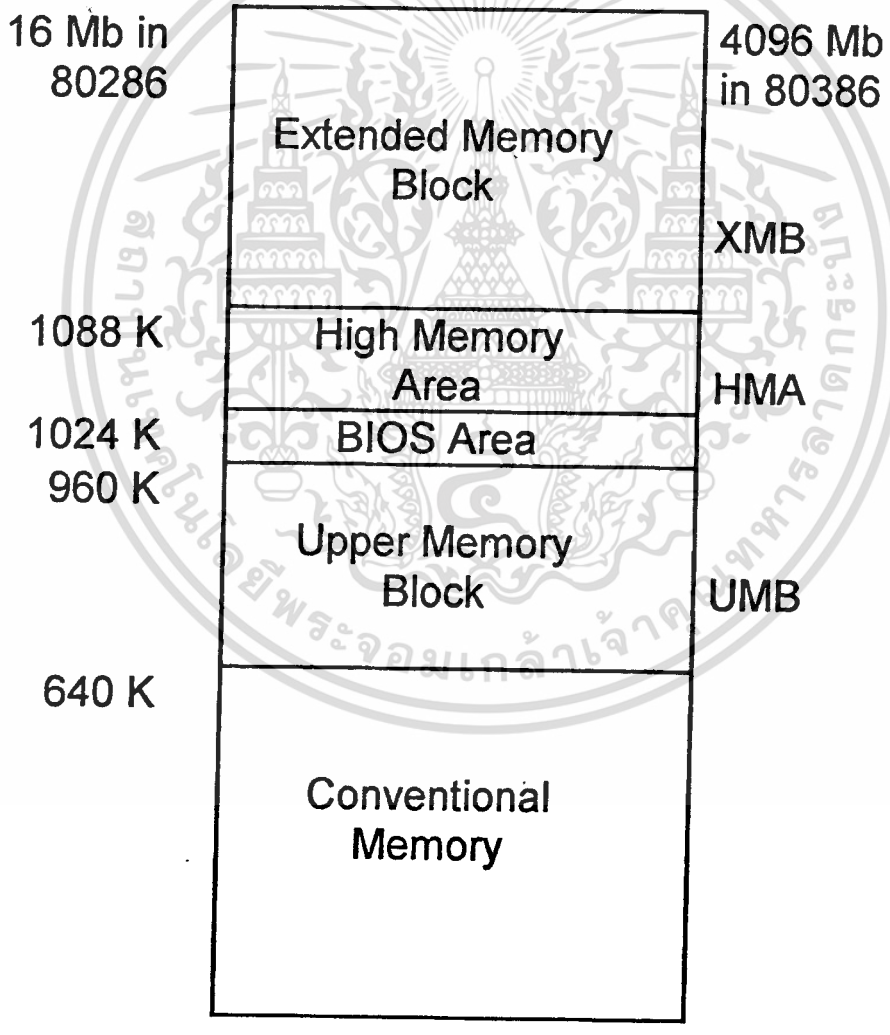
โปรแกรมดีไวซ์ไดรเวอร์ที่ชื่อว่า VDISK.SYS นั้นสามารถสร้างดิสต์จำลองจากหน่วยความจำทั้งแบบธรรมดาและแบบยืดขยายได้ ในการใช้หน่วยความจำยืดขยายของ VDISK จะมีการเก็บข้อมูลเอาไว้ว่ามีหน่วยความจำยืดขยายอยู่ที่ไหน บริเวณใดถูกใช้ไปแล้ว เหลืออีกเท่าไร เพื่อที่ว่าหากมีการเรียกโปรแกรม VDISK สำหรับสร้างดิสต์จำลองเพิ่มขึ้นอีก จะได้ไม่มีการใช้หน่วยความจำซ้ำซ้อนกัน อย่างไรก็ตาม วิธีการนี้ก็ยังคงประสบปัญหาอยู่นั่นเอง คือ ถ้าไม่มีการติดตั้ง VDISK ในระบบแล้ว ก็จะเหมือนกับการใช้ ไบออสแบบปกติ ถ้ามีการเปลี่ยนเวอร์ชันของ VDISK แล้วในเวอร์ชันใหม่มีการเปลี่ยนแปลงตำแหน่งของข้อมูลก็จะทำให้โปรแกรมทำงานผิดพลาดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.3. การใช้ข้อกำหนด XMS (eXtended Memory Specification)

ดีไวซ์ไดร์เวอร์ HIMEM.SYS ใช้สำหรับเป็นตัวจัดการหน่วยความจำแบบยืดขยาด เช่น โปรแกรม วินโดส์ จะเรียกใช้เพื่อที่จะได้เป็นตามข้อกำหนดของ XMS หรือ ไมโครซอฟต์ดอสเวอร์ชัน 5.0 สามารถโหลดตัวเองและดีไวซ์ไดร์เวอร์ไปอยู่ในหน่วยความจำส่วนบนได้

ไมโครซอฟต์ได้ประกาศข้อกำหนด XMS เพื่อให้มีการใช้กันอย่างแพร่หลายมากขึ้น ในข้อกำหนด ของ XMS จะมีตัวไดรเวอร์ซึ่งก็คือ HIMEM.SYS เข้ามาจัดการ ในข้อกำหนดของ XMS นี้มีหน่วยความจำให้ใช้ได้อยู่ 3 ชนิด ได้แก่ UMB (Upper Memory Block), HMA (High Memory Area), และ XMS (eXtended Memory Block)



รูปที่ 3.3 แสดงหน่วยความจำตามข้อกำหนดของ XMS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.3.1 UMB

คือ หน่วยความจำที่อยู่เหนือกว่าหน่วยความจำ 640 กิโลไบต์ของดอส แต่ยังอยู่ในช่วง 1 เมกกะไบต์ ที่สามารถอ้างถึงได้ในโหมดปกติ โดยปกติแล้วหน่วยความจำบริเวณนี้จะเป็นพื้นที่ที่ถูกใช้โดยไบออส ของระบบและพวกอุปกรณ์ต่าง ๆ ที่ต่ออยู่กับเครื่อง แต่ก็ยังคงมีบางส่วนที่ยังคงว่างอยู่สามารถที่จะนำมาใช้งานได้ ในการใช้ UMB นี้ต้องมีการจอง คือ ขอจากตัวไดรเวอร์ก่อน เมื่อขอแล้วจะได้รับค่าเซ็กเมนต์กลับมา จากนั้นจึงเข้าไปใช้หน่วยความจำที่บริเวณของเซ็กเมนต์นั้นได้ เมื่อใช้เสร็จแล้วก็ทำการคืนหน่วยความจำเพื่อให้โปรแกรมอื่นสามารถใช้ได้ ถ้าไม่มีการคืนหน่วยความจำ ถึงแม้ว่าโปรแกรมจะจบการทำงานแล้วก็ตาม ก็จะไม่มีโปรแกรมใดสามารถใช้หน่วยความจำส่วนนี้ได้

3.7.3.2 HMA บริเวณความจำสูง

ในหน่วยความจำแบบยืดขยายจะมีอยู่บริเวณหนึ่งที่พิเศษกว่าบริเวณอื่น คือ ส่วนที่มีขนาด 64 กิโลไบต์เหนือ 1 เมกกะไบต์ (บริเวณ 100000h - 110000h) หน่วยความจำบริเวณนี้สามารถอ้างถึงได้ในขณะที่ซีพียูจะทำงานอยู่ในโหมดปกติ พิจารณาจากวิธีการคำนวณแอดเดรส ถ้าเราให้เซ็กเมนต์มีค่าเป็น FFFFh ให้ออฟเซ็ตมีค่า 0010h ซึ่งจะได้ 100000h ซึ่งเป็นหน่วยความจำไบต์แรกที่เกิน 1 เมกกะไบต์ขึ้นไป การจะอ้างถึงหน่วยความจำ HMA ได้ต้องมีการจัดการกับสัญญาณ A20 มิฉะนั้นแล้วจะเกิดอาการที่เรียกว่า การวนกลับของแอดเดรส เพราะถ้าสัญญาณ A20 ไม่ทำงานเสียแล้วค่าแอดเดรสที่ 100000h ก็จะกลายเป็น 00000h และอ้างถึง HMA ก็จะกลายเป็นการอ้างถึงหน่วยความจำบริเวณ 000000h - 00FFEFh (0000:0000 - 0000:FFEF) ซึ่งเป็นบริเวณส่วนสำคัญของระบบ คือ เป็นที่อยู่ของตารางอินเทอร์รัปต์ และข้อมูลของไบออส ดังนั้นจึงต้องใช้ความระมัดระวังเป็นพิเศษ การจัดการสัญญาณ A20 นี้สามารถทำได้โดยการเรียกผ่านตัวไดรเวอร์ของ XMS

3.7.3.3 XMB หน่วยความจำยืดขยายแท้

ส่วนนี้เป็นส่วนที่มีขนาดใหญ่ที่สุด และ เป็นส่วยของหน่วยความจำยืดขยายอย่างแท้จริง อยู่ในบริเวณตั้งแต่ 110000h เป็นต้นไป การกล่าวถึงข้อกำหนด XMS จะหมายถึงหน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความจำในส่วนนี้เป็นสำคัญ ในการใช้งานเมื่อขอจองหน่วยความจำไปโดยบอกขนาดที่ต้องการไปยังตัวไดรเวอร์แล้ว ก็จะได้ค่าแฮนเดิล (Handle) กลับมาเพื่อที่จะใช้ในการอ้างถึงต่อไป ในการใช้งานหน่วยความจำยืดขยาย ผู้ใช้ไม่สามารถที่จะเขียนและอ่านหน่วยความจำนี้ได้ โดยตรงเหมือนกับการใช้ UMB หรือ HMA เพราะอยู่นอกเหนือความสามารถของโหมดปกติที่จะทำได้ สิ่งที่ทำได้คือการนำเองข้อมูลเก็บไว้ในบัฟเฟอร์ในหน่วยความจำปกติเสียก่อน จากนั้นจึงเรียกให้ตัวไดรเวอร์ทำการคัดลอกข้อมูลไปไว้ในหน่วยความจำยืดขยาย เมื่อต้องการใช้ก็ขอให้ไดรเวอร์คัดลอกกลับมาในหน่วยความจำปกติ เมื่อใช้งานเสร็จแล้วก็ต้องทำการยกเลิกเช่นเดียวกันหน่วยความจำ UMB และ HMA

3.8 การติดต่อกับไดรเวอร์

การติดต่อกับไดรเวอร์ของ XMS จะใช้วิธีการเรียกไปยังตำแหน่งที่อยู่โดยตรง โดยค่าแอดเดรสนี้จะได้มาจากการใช้ฟังก์ชัน 2Fh หลังจากได้ค่ากลับมาแล้วก็สามารถเรียกใช้ฟังก์ชันของ XMS ได้โดยให้หมายเลขฟังก์ชันอยู่ที่รีจิสเตอร์ AH เช่นเดียวกับการเรียกใช้ฟังก์ชันบริการของไบออสและดอส และมีขั้นตอนดังนี้

- ตรวจสอบว่าเวกเตอร์ของอินเทอร์รัปต์หมายเลข 2Fh มีค่าเป็น NULL หรือชี้ไปยังคำสั่ง IRET หรือไม่ ถ้าใช่แสดงว่าไม่มี XMS ไดรเวอร์อยู่ในระบบ
- ใช้ฟังก์ชัน INT 2Fh AX=4300h ตรวจสอบ ถ้า AL ไม่ใช่ 80h แสดงว่าไม่มี XMS ไดรเวอร์
- ใช้ฟังก์ชัน INT 2Fh AX=4310h ได้ XMS Driver Entry Point อยู่ใน ES:BX
- ตรวจสอบว่ามี HMA หรือไม่ ทำได้โดยการใช้ฟังก์ชันขอเวอร์ชันของ XMS (AH=00) ถ้าค่าของ DX เป็น 0 แสดงว่าไม่มีการติดตั้ง HMA อยู่ในระบบ
- ตรวจสอบว่ามี UMB หรือไม่ ส่วนของ UMB นี้ไม่จำเป็นต้องตรวจสอบ เพราะ ก่อนที่จะใช้ต้องมีการขอ ขนาดของ UMB ก่อน

3.9 ข้อควรระวังในการเรียกใช้ XMS

- ขอทราบขนาดของหน่วยความจำที่ยังว่างอยู่เสียก่อนว่าพอต่อความต้องการหรือไม่ ก่อนที่จะทำการของจองหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตรวจสอบการคืนค่าทุกครั้งที่เรียกใช้ฟังก์ชันของ XMS เพื่อให้แน่ใจว่าการทำงานสำเร็จตามที่ ต้องการ
- คืนหน่วยความจำที่เลิกใช้แล้วให้แก่ไดรเวอร์ เพื่อให้โปรแกรมอื่นสามารถเรียกใช้ได้ ทั้งนี้รวมถึงกรณีที่โปรแกรมจบการทำงานแบบไม่ปกติ (Abnormal terminated) ด้วย

3.10 รายละเอียดของฟังก์ชันของ XMS

AH	FUNCTION
00h	Get XMS version number
01h	Request High Memory Area (1M to 1M + 64k)
02h	Release High Memory Area
03h	Global enable A20 , for using the HMA
04h	Global disable A20
05h	Local enable A20, for direct access to extended memory
06h	Local disable A20
07h	Query A20 state
08h	Query free extended memory, not counting HMA
09h	Allocate extended memory block
0Ah	Free extended memory block
0Bh	Move extended memory block
0Ch	Lock extended memory block
0Dh	Unlock extended memory block
0Eh	Get handle information
0Fh	Reallocate extended memory block
10h	Request upper memory block (nonEMS memory above 640k)
11h	Release upper memory block

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การจัดการจอภาพ

ในบทนี้ จะกล่าวถึงวิธีการเขียนโปรแกรมควบคุมจอภาพบนระบบปฏิบัติการดอส โดยเน้นที่การ์ดแสดงผลวีจีเอ (VGA) ขึ้นไป เนื่องจากในปัจจุบัน ราคาของการ์ดแสดงผลวีจีเอมีราคาต่ำลงมาก จนทำให้การ์ดแสดงผลวีจีเอกลายเป็นมาตราฐานต่ำสุดของไมโครคอมพิวเตอร์ในปัจจุบัน

4.1 โหมดของจอภาพ

การแสดงผลบนจอภาพ แบ่งตามกรรมวิธีในการสร้างภาพบนจอได้ 2 วิธีคือ

1. โหมดตัวอักษร (Text mode) ในโหมดนี้ รหัสของตัวอักษรและรูปแบบการแสดงตัวอักษรซึ่งเรียกว่า แอตตริบิวต์ (Attribute) จะเก็บเรียงกันไปในวิดีโอแรม (VideoRAM) รหัสจะถูกนำไปใช้ในการเปิดตารางภาพของตัวอักษร (Character generators) ภาพที่ได้จากการเปิดตารางจะถูกนำไปแสดงบนจอภาพตามรูปแบบการแสดงที่กำหนด ในโหมดนี้เราจะเห็นว่าบนจอภาพจะถูกแบ่งพื้นที่ออกเป็นส่วนย่อยๆ ในแต่ละส่วนจะทำหน้าที่แสดงภาพตัวอักษรหนึ่งตัว เช่น ในโหมด 80 คูณ 25 ตัวอักษรก็จะมีพื้นที่สำหรับแสดงภาพตัวอักษรได้ 80 คอลัมน์ (จำนวนตัวอักษรในแนวนอน) และ 25 บรรทัด (จำนวนตัวอักษรในแนวตั้ง)

2. โหมดภาพ (Graphic mode) ในโหมดนี้ ข้อมูลที่อยู่ภายในวิดีโอแรมจะหมายถึงการแสดงจุดแต่ละจุดบนจอภาพโดยตรง กรรมวิธีการแสดงจุดนี้มีสองลักษณะย่อยคือ

2.1 ใช้การเปิดตารางค่าสี (Color look-up table) ข้อมูลที่อยู่ภายในวิดีโอแรมจะถูกนำไปเปิดตารางค่าสี เพื่อให้ได้ค่าความสว่างของแม่สีแดง สีเขียว และสีน้ำเงินของจุดภาพแต่ละจุด

2.2 ข้อมูลที่อยู่ภายในวิดีโอแรมเป็นค่าแม่สี (True color) ข้อมูลที่อยู่ภายในวิดีโอแรมจะหมายถึงค่าความสว่างของแม่สีโดยตรง ดังนั้น ในวิธีนี้จะต้องใช้พื้นที่หลายบิตของข้อมูลสำหรับแทนจุดหนึ่งจุดบนจอภาพ เพื่อให้ได้ภาพที่มีความสมจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ประวัติการพัฒนาการแสดงผลบนจอคอมพิวเตอร์

การควบคุมการแสดงผลบนจอภาพ จะกระทำโดยการ์ดแสดงผล (Video card) ภายในเครื่องคอมพิวเตอร์ ตัววิดีโอแรมเองก็จะอยู่บนการ์ดแสดงผลนี้ด้วย

การ์ดแสดงผลมีการพัฒนากันมาอย่างต่อเนื่อง นับตั้งแต่ไมโครคอมพิวเตอร์รุ่นแรกๆ ที่กำเนิดขึ้น โดยเริ่มจากการใช้จอโทรทัศน์ธรรมดาเป็นตัวแสดงภาพ การ์ดแสดงผลมีหน้าที่แปลงข้อมูลตัวอักษรให้เป็นสัญญาณภาพ สำหรับนำไปแสดงบนจอโทรทัศน์ ทำให้ลดต้นทุนการผลิตเครื่องไมโครคอมพิวเตอร์ลงได้

ต่อมา เมื่อคอมพิวเตอร์ในคระกูล XT ของไอบีเอ็มได้ออกสู่ท้องตลาดก็ได้มีการออกแบบการ์ดแสดงผลที่สามารถแสดงตัวอักษรได้ 80 คอลัมน์ 25 บรรทัด แต่ยังไม่สามารถแสดงเป็นรูปภาพตามต้องการได้ ในที่สุด ก็ได้มีบริษัท Hercules Computer Technology ผลิตการ์ดแสดงผลชื่อ Hercules Monochrome Adapter หรือที่รู้จักกันในนามของการ์ดเฮอร์คิวลิส ซึ่งนอกจากจะแสดงผลในโหมดตัวอักษร 80 ตัวอักษร 25 บรรทัดแล้ว ยังสามารถแสดงในโหมดภาพด้วยความละเอียด (resolution) 720 คูณ 350 จุด ใช้แทนการ์ดเดิมของไอบีเอ็มได้ทันที

แต่การแสดงผลเฮอร์คิวลิสยังคงเป็นภาพขาวดำที่มีระดับความสว่างระดับเดียวจึงได้มีผู้สร้างการ์ดแสดงผลแบบสีขึ้นมา ให้ชื่อว่า การ์ด CGA (Color Graphics Adapter) สามารถแสดงโหมดภาพด้วยความละเอียด 320 คูณ 350 จุด 16 สี

การ์ดแสดงผลทั้งสามชนิดที่กล่าวมานี้ จะต้องใช้จอภาพที่ออกแบบมาเฉพาะรุ่น และเนื่องจากการ์ดแสดงผลและจอภาพเหล่านี้ต่างก็ใช้สัญญาณภาพในรูปดิจิตอลทั้งหมด ทำให้เป็นข้อจำกัดในการพัฒนาคุณภาพของการ์ดและจอภาพ จึงได้มีการพัฒนาการ์ดแสดงผลแบบใหม่ ที่ให้สัญญาณออกมาเป็นสัญญาณที่ต่อเนื่อง (อนาล็อก) เพื่อให้สามารถพัฒนาการแสดงผลให้มีความใกล้เคียงกับภาพจริงมากขึ้น

การ์ดแสดงผลตัวแรกที่ให้สัญญาณอนาล็อกก็คือ การ์ด VGA (Video Graphics Adapter) ซึ่งจะต้องใช้จอภาพ VGA (Video Graphics Array) หรือจอภาพที่มีคุณสมบัติสูงกว่าในการแสดงผล ถึงแม้จอภาพ VGA จะสามารถรับสัญญาณที่เป็นอนาล็อก และการ์ด VGA ก็ให้สัญญาณออกมาเป็นอนาล็อก แต่ด้วยข้อจำกัดที่แหล่งผลิตสัญญาณก็เป็นเครื่องคอมพิวเตอร์ซึ่งทำงานในรูปสัญญาณดิจิตอล จึงทำให้จำนวนสีที่ได้ก็มีจำนวนจำกัดด้วย การ์ด VGA ซึ่งเป็นการ์ดอนาล็อกรุ่นแรก สามารถแสดงสีได้คราวละ 256 สีจากจำนวนสีที่สามารถแสดงได้ 262,144 สี ในความละเอียด 320 คูณ 200 จุด และเพื่อให้การ์ดรุ่นใหม่ๆ ที่ผลิตขึ้น สามารถใช้กับโปรแกรมที่ได้เขียนขึ้นสำหรับการ์ดรุ่นก่อนๆ การ์ดรุ่นใหม่จึงต้องมี

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถใช้กับโปรแกรมที่ได้เขียนขึ้นสำหรับการ์ดรุ่นก่อนๆ การ์ดรุ่นใหม่จึงต้องมีโหมดการแสดงสำหรับให้เข้ากันได้กับการ์ดรุ่นเก่าๆ ดังนั้น การ์ด VGA จึงมีโหมดการแสดงของการ์ด CGA และ EGA ได้

หลังจากนั้น มีการพัฒนาการ์ดให้มีความละเอียดในการแสดงภาพสูงขึ้น เช่นการ์ด SuperVGA (SVGA) ที่มีความละเอียดสูงถึง 1024 คูณ 768 จุดที่ 256 สี และการ์ด Extended VGA (XGA) ที่สามารถแดงสีได้ถึง 32,768 สี 65,536 สี และ 16.7 ล้านสีที่ความละเอียด 640 คูณ 480 จุด เป็นต้น

4.3 การจัดการโหมดภาพในภาษาซี

โดยปกติ การทำงานของโปรแกรมจะอยู่ในโหมดตัวอักษร หากเราจะต้องการให้โปรแกรมทำงานในโหมดภาพ จะต้องใช้ฟังก์ชันสำหรับเปลี่ยนโหมดให้มาอยู่ในโหมดภาพ และเมื่อต้องการเปลี่ยนกลับไปสู่โหมดตัวอักษร จะต้องใช้ฟังก์ชันปิดโหมดภาพ ฟังก์ชันที่ใช้ในการเปลี่ยนโหมดมีดังนี้คือ

```
#include <graphics.h>
void far initgraph(int far *graphdriver,int far *graphmode,char far *pathtodriver);
void far closegraph(void);
```

ฟังก์ชัน initgraph() เป็นฟังก์ชันสำหรับเปลี่ยนโหมดการทำงานมาเป็นโหมดภาพ ส่วนฟังก์ชัน closegraph() ทำหน้าที่ปิดโหมดภาพ ให้กลับไปสู่โหมดตัวอักษร

พารามิเตอร์ใน initgraph() มีรายละเอียดดังนี้

graphdriver เป็นพอยน์เตอร์สำหรับชี้ไปยังหมายเลขโหมดของจอภาพที่ต้องการโดยทั่วไป เราจะกำหนดให้เท่ากับ DETECT ซึ่งเป็นค่าคงที่ที่ได้นิยามไว้ใน graphics.h หมายถึงให้ใช้โหมดภาพที่มีความละเอียดสูงสุดที่การ์ดสามารถจะทำงานได้

graphmode เป็นพอยน์เตอร์สำหรับชี้ไปยังหมายเลขโหมดย่อย บอกระดับความละเอียดที่ต้องการ ในการใช้งานทั่วไป เราจะไม่เซ็ตค่านี้เมื่อเราใช้ค่าคงที่ DETECT ในการกำหนดโหมดจอให้สูงสุดอยู่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

pathtodriver คือข้อมูลสตริงที่หมายถึงชื่อไดเรกตอรีของไดรเวอร์จอภาพในการเขียนโปรแกรมภาษาซีและปาสคาลโดยใช้คอมไพเลอร์จกาบริษัทบอร์แลนด์ซีนั้น จะแยกส่วนควบคุมการแสดงผลออกมาต่างหาก เราเรียกส่วนควบคุมนี้ว่า กราฟิกไดรเวอร์ (Graphic driver) มีนามสกุลเป็น BGI และยังมีไฟล์รูปแบบตัวอักษรพิเศษที่มีนามสกุล CHR การกำหนดไดเรกตอรีในส่วนนี้จะทำให้โปรแกรมที่เขียนขึ้น เรียกกราฟิกไดเวอร์และไฟล์รูปแบบตัวอักษรพิเศษจากไดเรกตอรีที่กำหนด

4.4 การจัดการโหมดภาพในโปรแกรม

จากที่ได้กล่าวไว้ในหัวข้อ 4.3 การจะเปลี่ยนโหมดตัวอักษรให้เป็นโหมดภาพนั้นจะต้องมีการเรียกใช้คำสั่ง initgraph ซึ่งเป็นฟังก์ชันที่เตรียมเอาไว้ให้แล้ว แต่ในโปรแกรมจะมีการสร้างเป็นไลบารี่เกี่ยวกับคำสั่งที่จำเป็นต้องใช้ ในเรื่องของกราฟิกไว้ รายละเอียดต่างๆ ของฟังก์ชันที่อยู่ในไลบารี่ จะกล่าวในตอนหลัง นอกจะในไลบารี่นี้จะมีฟังก์ชันที่จำเป็นในการเรียกใช้งานแล้ว ในไลบารี่นี้ยังได้มีการรวมเอา กราฟิกไดรเวอร์ ซึ่งมีนามสกุลเป็น BGI และไฟล์ที่เก็บรูปแบบตัวอักษรพิเศษที่มีนามสกุลเป็น CHR เข้าไว้ในไลบารี่ด้วย ซึ่งเมื่อทำการคอมไพล์โปรแกรมแล้ว จะไม่จำเป็นที่จะต้องเรียก หรือค้นหาไฟล์จำพวกกราฟิกไดรเวอร์ หรือไฟล์ที่เก็บรูปแบบตัวอักษรพิเศษอีกเลย

4.5 ไลบารี่ GUI

ไลบารี่ GUI เป็นไลบารี่ที่จะเก็บฟังก์ชันที่จำเป็นในการใช้งาน โดยทั่วไปจะเป็นฟังก์ชันที่ทำงานจำพวก การสร้างเมนู (Menu) การตรวจสอบเมาส์ (Mouse) การสร้างวินโดว์ส (Window) เป็นต้น

นอกจากนั้นแล้ว ในไลบารี่นี้จะทำการรวมเอากราฟิกไดรเวอร์ เข้าไปไว้ 2 ตัวคือ

egavga.bgi

herc.bgi

และได้ทำการรวมเอาไฟล์ที่เก็บรูปแบบตัวอักษรพิเศษไว้อีกเป็นจำนวน 3 ไฟล์ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

trip.chr

litt.chr

sans.chr

นอกจากจะมีการรวมเอาไฟล์ดังกล่าวข้างต้นเข้าไว้ในไลบารี่นี้แล้ว ยังจะมีการเพิ่มไฟล์ที่เก็บตัวอักษรแบบปกติเข้าไปไว้ด้วย

จะสังเกตได้ว่ามีไฟล์เกี่ยวกับกราฟิกไดรเวอร์อยู่เพียง 2 ไฟล์เท่านั้นที่นำมารวมเข้ากับไลบารี่ GUI ซึ่งจะหมายความว่า เมื่อทำการรันโปรแกรมจะสามารถใช้ได้กับจอภาพชนิดเฮอร์คิวลิส (Hercules) และวีจีเอ (VGA) หรืออีจีเอ (EGA) เท่านั้น โดยจะใช้โหมดจอที่สูงที่สุดเท่าที่เป็นไปได้ คือถ้าเป็นจอภาพชนิดวีจีเอ หรืออีจีเอ จะสามารถแสดงสีได้ถึง 16 สี และมีความละเอียดเท่ากับ 640 จุดคูณ 480 จุด ถ้าเป็นจอภาพชนิดเฮอร์คิวลิสนั้น จะสามารถแสดงสีได้ 2 สี และมีความละเอียดเท่ากับ 720 จุดคูณ 348 จุด

4.6 อัลกอริทึมและความหมายของฟังก์ชันที่อยู่ในไลบารี่ GUI

The Choose File function

chooseFile

ฟังก์ชัน chooseFile จะแสดงเป็นหน้าต่าง (window) เพื่อใช้ในการเลือกชื่อไฟล์เพื่อใช้สำหรับอ่าน หรือเพื่อใช้สำหรับการเก็บข้อมูลลงไฟล์นั้น และยังมีที่จะทำการเปลี่ยนไดรฟ์และไดเรกทอรี่ได้ ฟังก์ชันจะคืนค่า (return value) เป็นจริง (true) ถ้ามีการเลือกไฟล์ (ยกตัวอย่างเช่น มีการดับเบิลคริก (double-clicked) หรือกดปุ่ม OK เพื่อเลือกไฟล์)

General dialog functions

KeyBoard

ฟังก์ชัน Keyboard ไว้ใช้สำหรับการตรวจสอบการกดคีย์บอร์ด และยังนำมาใช้ในฟังก์ชัน Editfield ในการรับคีย์บอร์ดอีกด้วย โดยการนำไปใช้นั้น ฟังก์ชัน Keyboard จะต้องถูกวนเรียกตลอดเวลาเพื่อให้สามารถที่จะรับการกดคีย์ได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Editfield functions

addEditfield	ฟังก์ชัน addEditfield จะทำการเพิ่ม editfield เข้าไปยังหน้าต่าง
drawEditfield	ฟังก์ชันนี้จะทำการวาด editfield ลงไปยังหน้าต่าง
findEditfield	ตรวจสอบตำแหน่งของการกดเมาส์ โดยจะมีการคืนค่ากลับเป็นพอยน์เตอร์
trackEditfield	ทำให้ editfield เริ่มทำงานเนื่องจากการกดเมาส์

Checkbox functions

addCheckbox	เพิ่ม checkbox เข้าไปยังหน้าต่างที่กำลังกระทำอยู่
drawCheckbox	วาดรูปของ checkbox ลงบนหน้าต่างที่ได้ทำการเพิ่ม checkbox ไว้ก่อนหน้านี้
findCheckbox	ตรวจสอบว่าได้มีการกดเมาส์ที่ checkbox ไหน โดยจะทำการคืนค่ากลับเป็นพอยน์เตอร์
trackCheckbox	ให้ checkbox นั้นเริ่มทำงาน

Scroll bar functions

addHorScroll	เพิ่ม scoll bar ตามแนวขวางเข้าไปยังหน้าต่างที่กำลังกระทำอยู่
addVertScroll	เพิ่ม scoll bar ตามแนวตั้งเข้าไปยังหน้าต่างที่กำลังกระทำอยู่
drawHorScroll	ทำการวาด scoll bar ตามแนวขวางลงบนหน้าต่าง ที่ได้มีการเพิ่ม scoll bar ไว้ก่อนหน้านี้
drawVertScroll	เช่นเดียวกับฟังก์ชัน drawHorScroll จะทำการวาด scoll bar แต่จะเป็นในลักษณะแนวตั้งแทน โดยที่จะทำการวาดลงบนหน้าต่างที่ได้มีการเพิ่ม scoll bar ไว้ก่อนหน้านี้แล้วเท่านั้น
findScrollbar	จะทำการตรวจสอบว่าได้มีการกดเมาส์ที่ scoll bar ไหน ซึ่งจะคืนค่ากลับเป็นพอยน์เตอร์ เพื่อบอกว่าเป็น scoll bar ไหน
trackScrollbar	ให้ scoll bar ที่ได้มีการกดเมาส์ เริ่มทำงานตามที่ได้รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

List functions

addlist	ทำการเพิ่ม list เข้าไปยังหน้าต่างนั้นๆ
drawList	วาด list ลงไปยังหน้าต่าง
findList	หาตำแหน่งว่าการกดเมาส์นั้น ได้กดลงไปยัง list ตัวไหน โดยจะมีการคืนค่าเป็นพอยน์เตอร์ ที่ชี้ไปยัง list ตัวนั้น
trackList	เริ่มทำงาน

Text and Textfield functions

addText	เพิ่มการควบคุมตัวอักษรเข้าไปยังหน้าต่าง
.addTextfield	เพิ่มการควบคุม textfield เข้าไปยังหน้าต่าง
drawText	วาดตัวอักษร
drawTextfield	วาด textfield

Bitmap functions

addBitmap	เพิ่มการควบคุม bitmap เข้าไปยังหน้าต่าง
drawBitmap	วาด bitmap เข้าไปยังหน้าต่าง ซึ่งจะต้องมีการเพิ่มการควบคุม bitmap ไว้แล้วก่อนหน้านี้
findBitmap	ตรวจสอบหาตำแหน่งการกดเมาส์ว่าได้ทำการกดที่ bitmap ตัวไหน
trackBitmap	bitmap ที่ถูกกดโดยเมาส์ เริ่มทำงาน

Button functions

addButton	เพิ่มการควบคุมเกี่ยวกับปุ่มเข้าไปยังหน้าต่างนั้นๆ
drawButton	วาดรูปปุ่มกดลงไปบนหน้าต่างที่ได้มีการ เพิ่มการควบคุมเกี่ยวกับปุ่มนี้เข้าไป
findButton	ค้นหาว่าการกดเมาส์นั้น ได้ทำการกดลงไปบนปุ่มไหน และจะคืนค่ากลับเป็นพอยน์เตอร์ ที่ชี้ไปยังปุ่มกดนั้น
trackButton	ปุ่มที่ถูกกดเริ่มทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Menu functions

addMenu	เพิ่มการควบคุมเกี่ยวกับเมนู เข้าไปยังหน้าต่าง
dispatchMenuItem	เป็นฟังก์ชันที่ใช้ในการตรวจสอบการคีย์ เนื่องจากในการทำเมนูนั้น จะมีการกำหนดฮอตคีย์ (hot-key) เพื่อความสะดวกรวดเร็วในการทำงาน หรือการเรียกใช้เมนู ดังนั้นจะต้องมีการคอยตรวจสอบคีย์บอร์ดด้วย การฟังก์ชันนี้ไปใช้งาน จะต้องมีการเรียกฟังก์ชันนี้ตลอดเวลา โดยใช้การวนลูปเพื่อคอยตรวจสอบการกดคีย์
doMenu	แสดงเมนู เมื่อมีการกดเมาส์ที่แถบแสดงรายการของเมนู จะมีการคืนค่าเป็นจริง ถ้าเมนูนั้นถูกเปิด และจะคืนค่าเท็จ เมื่อไม่สามารถที่จะทำการเปิดเมนูนั้นได้ อันเนื่องมาจากไม่มีหน่วยความจำเพียงพอในการจองเพื่อแสดงเมนูนั้นๆ
drawMenuBar	วาดแถบแสดงรายการของเมนูที่ตำแหน่งบนสุดของจอภาพ
idle	เป็นฟังก์ชันที่เรียกใช้ เมื่อไม่ต้องการใช้กระทำคำสั่งใดๆ คือจะไม่ทำงานอะไร
InitMenuManager	ทำการกำหนดค่าเริ่มต้นต่างๆ ให้กับเมนู โดยจะต้องการทำการเรียกฟังก์ชันนี้เมื่อเริ่มต้นโปรแกรม

Window functions

closeWindow	ทำการลบหน้าต่างนั้นออกจากหน้าจอ จะมีการนำภาพบนหน้าจอที่ถูกหน้าต่างนั้นทับหรือปิดบังไว้ กลับคืนเป็นอย่างเดิม และจะมีการคืนหน่วยความจำที่ได้ทำการจองเอาไว้เมื่อตอนเปิดหน้าต่างกลับคืนด้วย
InitWindowManager	ทำการกำหนดค่าเริ่มต้นต่างๆ ให้กับการสร้างหน้าต่าง ซึ่งฟังก์ชันนี้จะต้องทำการเรียกใช้เมื่อเริ่มต้นโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

openWindow	ใช้เมื่อต้องการสร้างหน้าต่างขึ้นมา โดยจะมีการจองเนื้อที่ในหน่วยความจำที่มีขนาดเท่ากับขนาดของหน้าต่างนั้น ใช้สำหรับเก็บหน้าจอภาพที่จะต้องถูกหน้าต่างนั้นทับหรือถูกบัง ฟังก์ชันนี้จะทำการคืนค่าเป็นจริง เมื่อหน้าต่างสามารถเปิดได้สำเร็จ และจะทำการคืนค่าเป็นเท็จเมื่อหน้าต่างไม่สามารถที่จะเปิดได้ เนื่องมาจากไม่สามารถทำการจองหน่วยความจำได้
Graphic functions	
deinitGraphics	เป็นฟังก์ชันที่ใช้สำหรับการเปลี่ยนโหมดภาพให้กลายเป็นโหมดตัวอักษร
initGraphics	เป็นฟังก์ชันที่ใช้สำหรับการเปลี่ยนโหมดตัวอักษรให้กลายเป็นโหมดภาพ
Mouse functions	
InitMouse	เป็นฟังก์ชันที่ใช้ในการกำหนดค่าเริ่มต้นให้กับการเรียกใช้เมาส์ โดยการใช้เมาส์นี้จะสามารถใช้ได้ในโหมดภาพเท่านั้น ฟังก์ชันนี้จะทำการคืนค่าเป็นจริงเมื่อมีไดร์เวอร์ของเมาส์คือสามารถที่จะทำการใช้เมาส์ได้นั้นเอง และจะคืนค่าเป็นเท็จเมื่อไม่สามารถใช้เมาส์ได้
MouseButton	จะทำการคืนค่าเป็นจำนวนเต็ม ซึ่งจะบอกสถานะปุ่มของเมาส์ในขณะนั้น โดยบิตแรกจะแสดงถึงสถานะปุ่มทางด้ายซ้ายของเมาส์ และบิตที่สองจะบอกสถานะปุ่มของเมาส์ทางด้านขวา โดยถ้าบิตนั้นๆ ถูกเซ็ตไว้ แสดงว่าได้มีการกดปุ่มของเมาส์ทางด้านนั้น
MouseDown	เป็นฟังก์ชันที่ใช้ในการตรวจสอบหาตำแหน่งของเมาส์ ที่ได้มีการกดปุ่ม โดยจะทำการคืนค่ากลับมาเป็นตำแหน่งของเมาส์ในขณะนั้น
MouseLoc	ใช้สำหรับหาตำแหน่งของเมาส์ในขณะนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MouseOff	ทำการซ่อนเมาส์ไว้
MouseOn	แสดงเมาส์ออกมายังจอภาพ
MoveMouse	เปลี่ยนตำแหน่งของเมาส์ไปยังตำแหน่งที่กำหนด
SetupMouse	กำหนดขอบเขตของเมาส์
arrowCursor	เป็นรูปของเมาส์ ที่ใช้ในการแสดงบนจอภาพ จะมีลักษณะเป็นรูปลูกศรธรรมดา
waitCursor	เป็นรูปของเมาส์ ที่ใช้ในการแสดงบนจอภาพ แต่จะมีลักษณะเป็นรูปนาฬิกาทราย
whereMouse	เป็นฟังก์ชันที่ใช้สำหรับตรวจสอบว่า ได้มีการกดปุ่มเมาส์ ที่ตำแหน่งได้บนหน้าต่างนั้น โดยจะทำการคืนค่ากลับมาเป็นลักษณะของสิ่งของที่เมาส์นั้นได้กดปุ่มลงไป เช่น ได้กดปุ่มลงไปยังตำแหน่งของเมนู ก็จะทำการคืนค่าโดยบอกว่าเป็นแถบเมนู เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การติดต่อสื่อสาร

ระบบเครือข่ายคอมพิวเตอร์ขนาดท้องถิ่น (LAN : Local Area Network) คือ การเชื่อมต่อเครื่องคอมพิวเตอร์ โดยที่เครื่องคอมพิวเตอร์แต่ละเครื่องสามารถสื่อสารกันได้โดยใช้แพคเกจข่าวสาร ในแพคเกจ จะมีข้อมูลซึ่งบอกตำแหน่งของผู้ที่ส่งแพคเกจนั้นและผู้รับแพคเกจ ซึ่งจะนำส่วนดังกล่าวมาใช้เพื่อการจัดเส้นทาง (Routing) ในระบบเครือข่ายคอมพิวเตอร์ขนาดท้องถิ่นจะต้องมี ไฟล์เซอร์เวอร์ (File Server) อย่างน้อย 1 เครื่อง อย่างไรก็ตามการทำงานต่าง ๆ จะกระทำที่เครื่องคอมพิวเตอร์นั้น ๆ ไม่ใช่บนเครื่องไฟล์เซอร์เวอร์ และภายใต้ระบบเครือข่ายคอมพิวเตอร์ขนาดท้องถิ่นนั้นจะมีระบบปฏิบัติการที่ช่วยให้เครื่องคอมพิวเตอร์แต่ละเครื่องสามารถใช้แฟ้มข้อมูลร่วมกัน การป้องกันเรคอร์ด การกำหนดตั้งชื่อเครื่อง และการส่งข่าวสารถึงเครื่องคอมพิวเตอร์เครื่องอื่น ๆ

ระบบเครือข่ายคอมพิวเตอร์ขนาดท้องถิ่นเป็นกลุ่มของเครื่องคอมพิวเตอร์ที่เชื่อมต่อกับระบบ โดยใช้สายเคเบิล ในเครื่องคอมพิวเตอร์แต่ละเครื่องจะประกอบไปด้วย อุปกรณ์การเชื่อมต่อระบบเครือข่าย และ ซอร์ฟแวร์สำหรับระบบเครือข่าย เครื่องคอมพิวเตอร์แต่ละเครื่องจะมีที่อยู่ประจำเครื่องบนระบบเครือข่ายเรียกว่า โหนด (Node) หรือ เวิร์คสเตชั่น (Work-station) ซอร์ฟแวร์สำหรับระบบเครือข่ายในแต่ละเครื่องจะมีการแบ่งการทำงานเป็นชั้น ๆ ชั้นที่อยู่ต่ำที่สุดจะทำหน้าที่ติดต่อกับอุปกรณ์การเชื่อมต่อระบบเครือข่าย ในชั้นระดับบนที่สุดจะทำหน้าที่ติดต่อกับโปรแกรมของผู้ใช้งานระบบ และ จัดการช่วยเหลือด้านการติดต่อโปรแกรมของผู้ใช้งานระบบ กับ กระบวนการการใช้งานระบบเครือข่าย ในซอร์ฟแวร์แต่ละชั้นจะพัฒนาขึ้นโดยมีการกำหนดหน้าที่ในแต่ละชั้นไว้ก่อนล่วงหน้าแล้ว

5.1 แพคเกจข่าวสาร (Message Packets , Frames)

ในการสื่อสารชั้นล่างสุดนั้นเครื่องคอมพิวเตอร์ในระบบเครือข่ายนั้นทำการติดต่อกับเครื่องอื่น ๆ ในระบบ และ กับเครื่อง ไฟล์เซอร์เวอร์ โดยการใช้แพคเกจข่าวสาร หรือ มักจะเรียกว่า เฟรม ขั้นตอนการทำงานทั้งหมดของระบบเครือข่ายคอมพิวเตอร์ขนาดท้องถิ่นนั้นจะมีการรับส่งโดยใช้อุปกรณ์การเชื่อมต่อระบบเครือข่าย และ การเรียกการใช้งานผ่านซอร์ฟแวร์สำหรับอุปกรณ์นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอร์ฟแวร์สำหรับระบบเครือข่ายนั้น ทำการส่งเฟรมเพื่อการทำงานในหลาย ๆ หน้าที่ดังนี้

- การเริ่มต้นเปิดช่องทางการสื่อสาร
- การส่งข้อมูล เช่น เรคอร์ดจากแฟ้มข้อมูล ไปยังเครื่องคอมพิวเตอร์
- การตอบรับการรับรู้การมาถึงของข้อมูล
- การกระจายข่าวสารไปยังเครื่องอื่น ๆ
- การปิดช่องทางการสื่อสาร

Sender ID	Dest ID	Frame type	Data/message	CRC

รูปที่ 5.1 รูปแบบพื้นฐานของเฟรม

จากรูปที่ 1 แสดงรูปแบบพื้นฐานของเฟรม การพัฒนาระบบเครือข่ายขึ้นมานั้น มีการกำหนดรูปแบบของเฟรมหลายรูปแบบแตกต่างกันไป แต่รูปแบบที่แสดงไว้นั้นเป็นข้อมูลที่มีการใช้เป็นประจำประกอบไปด้วย

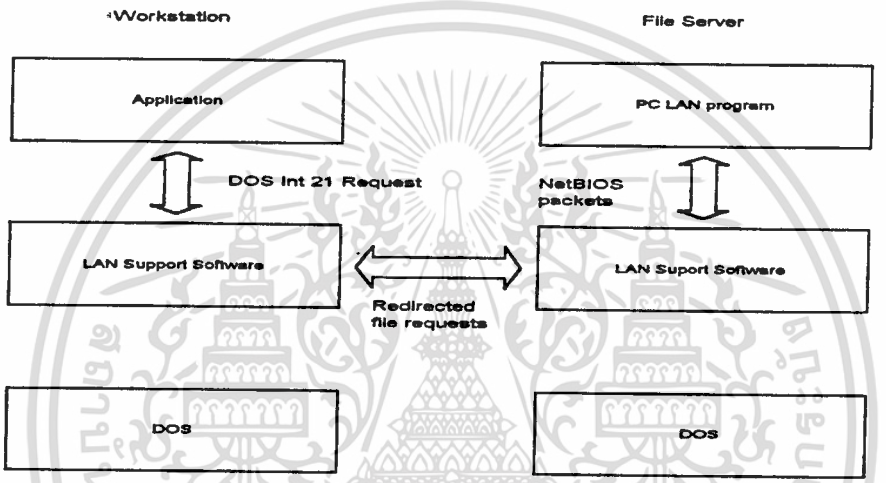
- ตำแหน่งที่อยู่ระบบเครือข่ายของผู้ที่ส่งเฟรมนั้น
- ตำแหน่งที่อยู่ระบบเครือข่ายของผู้รับเฟรมนั้น
- ข้อมูลแสดงประเภทของข้อมูลภายในเฟรม
- ข้อมูลหรือข่าวสาร
- ข้อมูลตรวจสอบความถูกต้อง เช่น การตรวจสอบผลรวมหรือการตรวจสอบแบบ CRC

เมื่อเฟรมข่าวสารต้องการเปิดแฟ้มข้อมูลที่อยู่ในไฟล์เซอร์เวอร์ จะทำเรียกการทำงานนั้นโดยเรียกใช้ การเปลี่ยนทิศทางการทำงานโดยระบบปฏิบัติการ สำหรับเครื่องคอมพิวเตอร์ที่มิได้เชื่อมต่อกับระบบเครือข่าย การทำอินเทอร์รัพท์ 21 เป็นการช่องทางการเรียกการใช้งานระบบปฏิบัติการ ในระบบเครือข่ายคอมพิวเตอร์อินเทอร์รัพท์ 21 จะถูกดักการทำงานโดยซอร์ฟแวร์ระบบเครือข่าย โดยจะดึงการทำงานบางอย่างไปทำบนเครื่องไฟล์เซอร์เวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอร์ฟแวร์ระบบเครือข่ายที่ทำหน้าที่กลั่นกรองการทำงานของอินเทอร์รัพท์ 21 ซอร์ฟแวร์นี้เรียกว่า เชลล์ (Shell) หรือ ตัวเปลี่ยนทิศทาง (Redirector)

เชลล์ จะทำการตรวจสอบแฟ้มที่ต้องการจะเปิด ในกรณีที่แฟ้มนั้นอยู่ในไฟล์เซอร์เวอร์ เชลล์ นั้นจะทำการส่งผ่านข้อมูลโดยใช้ แพคเกจข่าวสาร ในการขนย้ายข้อมูลระหว่างเครื่องคอมพิวเตอร์และเครื่องไฟล์เซอร์เวอร์ ในกรณีที่แฟ้มที่ต้องการเปิดอยู่ในดิสต์ของเครื่องคอมพิวเตอร์นั้นอยู่แล้วก็จะทำการเปิดแฟ้มนั้นโดยไม่มีการเรียกใช้งานระบบเครือข่าย



รูปที่ 5.2 การเปลี่ยนทิศทางโดยการทำงานของระบบปฏิบัติการณ์

การรับส่งเฟรมจะกระทำโดยซอร์ฟแวร์ระบบเครือข่ายมิได้กระทำโดยผู้ใช้ ขั้นตอนการทำงานของเครื่องคอมพิวเตอร์จะเป็นตัวเรียกการทำงานนั้นเองเมื่อมีการใช้แฟ้มข้อมูลที่อยู่ภายในไฟล์เซอร์เวอร์ หรือ โดยการทำงานตามโปรโตคอลเช่น จาก NetBIOS หรือ IPX ที่จะส่งข่าวสารถึงเครื่องคอมพิวเตอร์เครื่องอื่นภายในระบบเครือข่าย

5.2 ส่วนประกอบของระบบเครือข่ายคอมพิวเตอร์ระดับท้องถิ่น

5.2.1 อุปกรณ์เชื่อมต่อระบบเครือข่าย

อุปกรณ์เชื่อมต่อระบบเครือข่ายอาจเป็นได้ทั้งแบบ ตรวจสอบการชน (Collision-sensing) หรือ ประเภท ตรวจสอบการผ่านมาของโทเคน (Token-passing) อุปกรณ์ทั้งสอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภทจะประกอบไปด้วยอุปกรณ์ตรวจสอบสภาวะความพร้อมในการส่งเฟรม และ ตรวจรับเฟรมซึ่งจะส่งมาถึงตัวเอง อุปกรณ์ทั้งสองประเภทจะมีหน้าที่หลัก 7 ขั้นตอนระหว่างการรับและการส่งเฟรม ในกรณีที่มีการส่งข้อมูลขั้นตอนจะเป็นไปดังต่อไปนี้ และ ในกรณีที่มีการรับข้อมูลขั้นตอนจะกลับลำดับกัน

1. การส่งผ่านข้อมูล ข้อมูลจะถูกส่งผ่านมาจากหน่วยความจำของเครื่องคอมพิวเตอร์ไปยังอุปกรณ์เชื่อมต่อระบบเครือข่าย หรือ จากอุปกรณ์เชื่อมต่อระบบเครือข่ายไปยังหน่วยความจำของเครื่องคอมพิวเตอร์ โดยการเข้าถึงโดยตรง การใช้หน่วยความจำร่วมกัน หรือ โปรแกรมการจัดการอุปกรณ์รับส่งข้อมูล
2. การจัดเก็บข้อมูล ในขณะที่อุปกรณ์การเชื่อมต่อระบบเครือข่ายกำลังทำงานอยู่นั้น ข้อมูลจะถูกจัดเก็บในหน่วยความจำชั่วคราว หน่วยความจำนี้จะทำหน้าที่พักข้อมูลเพื่อให้อุปกรณ์การเชื่อมต่อระบบสามารถจัดการข้อมูลแต่ละเฟรม และ จัดการด้านความเร็วที่แตกต่างกันเนื่องจากความเร็วของเครือข่ายและความเร็วของการส่งผ่านข้อมูลในเครื่องคอมพิวเตอร์
3. การจัดรูปแบบของเฟรม อุปกรณ์การเชื่อมต่อระบบเครือข่าย จะจัดแบ่งข้อมูลให้เป็นส่วน ๆ ในเครือข่ายอีเทอร์เนต (Ethernet network) แต่ละส่วนจะมีขนาด 4 กิโลไบต์ ในขณะที่เครือข่ายบางระบบจะใช้ 12 กิโลไบต์โดยทั่วไปขนาดจะมีตั้งแต่ 1 ถึง 4 กิโลไบต์ ส่วนหัวของเฟรมจะเป็นตัวบอกจุดเริ่มต้นของแพคเกจ และส่วนท้ายจะเป็นตัวบอกจุดสิ้นสุดแพคเกจ เมื่อกระทำขั้นตอนดังกล่าวเสร็จจะได้เฟรมที่พร้อมจะทำการส่ง (ถ้าเป็นการรับเฟรม จะทำการถอดส่วนหัวและส่วนท้ายออก)
4. การเชื่อมต่อกับระบบนำสัญญาณ ในระบบที่ใช้วิธีตรวจสอบการชน เช่น อีเทอร์เนต จะทำการตรวจสอบระบบนำสัญญาณว่าระบบนำสัญญาณนั้นไม่มีการใช้แล้วจึงส่ง ในระบบที่ใช้วิธีตรวจสอบการผ่านมาของโทเคน จะทำการตรวจสอบโทเคนว่าว่างหรือไม่ ถ้าว่างจึงบรรจุข้อมูลลงในโทเคนนั้น
5. การเปลี่ยนแปลงข้อมูลเป็นแบบขนานหรือแบบอนุกรม ข้อมูลในแต่ละไบต์จะทำการรับส่งแบบอนุกรม ซึ่งอุปกรณ์การเชื่อมต่อระบบเครือข่ายจะต้องเปลี่ยนรูปแบบข้อมูลให้เหมาะสมเสียก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. การเข้ารหัสและการถอดรหัส รูปแบบของสัญญาณทางไฟฟ้าจะเป็นตัวแสดงข้อมูลทั้งในการรับและการส่ง อุปกรณ์การเชื่อมต่อระบบเครือข่ายจะใช้การเข้ารหัสแบบแมนเชสเตอร์ (Manchester Encoding)
7. การรับและการส่งสัญญาณ สัญญาณทางไฟฟ้าที่ได้รับการเข้ารหัสแล้วจะถูกขยายสัญญาณและทำการส่งไปยังระบบสื่อสัญญาณ

อุปกรณ์การเชื่อมต่อระบบเครือข่ายนอกจากจะแตกต่างกันในด้านของกรรมวิธีการเข้าถึงและขั้นตอนการทำงานแล้ว ยังมีความแตกต่างกันในเรื่องดังต่อไปนี้

- ความเร็วในการรับส่งข้อมูล
- ขนาดหน่วยความจำชั่วคราวในแผ่นวงจร
- การออกแบบเส้นทางการติดต่อ (8 บิท , 16 บิท , ไมโครชัลแนล)
- ความเร็วของเส้นทางการติดต่อ
- ความสามารถในการทำงานร่วมกันได้กับหน่วยประมวลผลที่ต่างชนิดกัน
- การใช้การเข้าถึงหน่วยความจำโดยตรง (DMA : Direct Memory Access)
- การกำหนด IRQ และ ตำแหน่งการเชื่อมต่อ
- ความเชี่ยวชาญพิเศษ (Intelligence)
- การออกแบบส่วนการเชื่อมต่อ (Connector Design)

5.2.2 ระบบการสื่อสัญญาณ

ระบบการสื่อสัญญาณสำหรับระบบเครือข่ายจะแตกต่างกันในด้าน คุณลักษณะ จุดประสงค์การใช้งาน และ ราคาต้นทุน ระบบสื่อสัญญาณที่มีการใช้ มีอยู่ด้วยกัน 3 ประเภทคือ

- สายคู่ตีเกลียวทั้งแบบชิลล์และไม่มีการชิลล์ (Twisted pair shielded and unshielded)
- สายโคแอคเชียล (Coaxial cable)
- ใยแก้วนำแสง (Fiber-optic cable)

5.2.3 ไฟล์เซอร์เวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้สำหรับเก็บแฟ้มข้อมูลเตรียมสำหรับการใช้แฟ้มข้อมูลร่วมกัน โดยทั่วไปจะนำเครื่องคอมพิวเตอร์ส่วนบุคคลมาใช้เป็นเครื่องไฟล์เซอร์เวอร์ หรืออาจจะใช้เครื่องคอมพิวเตอร์ต่างชนิดกันก็ได้ อย่างไรก็ตามไฟล์เซอร์เวอร์จะต้องมีคุณสมบัติดังต่อไปนี้

- สามารถเข้าถึงแฟ้มข้อมูลได้ด้วยความเร็วสูง
- สามารถให้บริการแฟ้มข้อมูลหรือข้อมูลบางเรคอร์ดสำหรับผู้ใช้งานหลาย ๆ คนพร้อมกัน
- มีระบบรักษาความปลอดภัยให้กับแฟ้มข้อมูล
- ระบบมีความเชื่อถือได้

ถ้ามีการใช้เครื่องคอมพิวเตอร์ชนิดอื่น ๆ ทำหน้าที่เป็นไฟล์เซอร์เวอร์จะต้องตรวจสอบการใช้งาน โดยจะต้องสามารถเชื่อมต่อกับระบบได้และเครื่องนั้นสามารถทำหน้าที่เป็นไฟล์เซอร์เวอร์ได้

5.2.4 ฮาร์ดดิสต์สำหรับเครื่องไฟล์เซอร์เวอร์

ความเร็วในการเข้าถึงและความจุของฮาร์ดดิสต์เป็นสิ่งที่จะต้องคำนึงมากที่สุดสำหรับ เครื่องไฟล์เซอร์เวอร์ ปัญหาที่ทำให้ความเร็วของระบบเครือข่ายลดลง บ่อยครั้งที่สุดเกิดจากเวลาที่ดิสต์ใช้ในเครื่องไฟล์เซอร์เวอร์ และปัญหาที่ผู้ใช้งานประสบมากที่สุดคือ มีเนื้อที่บนดิสต์ไม่เพียงพอ ปัญหาด้านความเร็วของดิสต์เกิดจากปัจจัยหลาย ๆ ด้านดังนี้

- กระบวนการการบันทึกข้อมูล (MFM, RLL, ESDI หรือ SCSI)
- ชนิดหรือขีดความสามารถของวงจรควบคุมการทำงานของดิสต์
- ชนิดของฮาร์ดดิสต์ (Stepper band หรือ Voice coil)
- ค่าอินเตอร์ลีฟ (Interleave factor)
- การจัดตำแหน่งการเก็บแฟ้มข้อมูลบนดิสต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.5 ซอร์ฟแวร์สำหรับระบบเครือข่าย

ซอร์ฟแวร์สำหรับระบบเครือข่ายจะทำหน้าที่จัดการให้เครื่องคอมพิวเตอร์สามารถติดต่อกับเครื่องอื่น ๆ ในระบบได้ ซอร์ฟแวร์ดังกล่าวจะต้องสามารถใช้งานได้กับเครื่องที่มีอยู่ และจะต้องมีความสามารถดังนี้

- สามารถเข้าถึงอุปกรณ์เชื่อมต่อระบบเครือข่ายได้
- ยอมให้มีการติดต่อระหว่าง เครื่องคอมพิวเตอร์ กับ เครื่องคอมพิวเตอร์ โดยตรงได้
- จำลองการทำงานของระบบปฏิบัติการณ์ ดังนั้นดิสต์บนเครื่องไฟล์เซอร์เวอร์ จึงสามารถถูกใช้งานได้โดยตรง

5.3 การติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ กับ เครื่องคอมพิวเตอร์

ผู้ผลิตแต่ละบริษัทได้มีการออกแบบโปรโตคอลเป็นของตัวเองสำหรับการสื่อสารระหว่างเครื่องบนระบบเครือข่าย ต่อมาผู้ผลิตได้รวมกันจัดตั้งมาตราฐาน และจัดตีพิมพ์ขึ้นโดยองค์การมาตราฐานสากล (ISO : International Standards Organization) เรียกว่า รูปแบบการเชื่อมต่อระบบเปิด (OSI : Open Systems Interconnection Model) มีคุณสมบัติ ดังต่อไปนี้

- การเริ่มต้นการติดต่อสื่อสาร (Initiating communications) ในแต่ละแบบจะต้องสามารถแสดงเครื่องแต่ละเครื่องได้ โดยการอนุญาติให้ตั้งชื่อเครื่อง หรือ การให้หมายเลขประจำเครื่อง หรือ ใช้ทั้งสองอย่าง คุณลักษณะนี้จะถูกใช้โดยเซลล์ และหรือโปรแกรมที่ใช้งาน การติดต่อสื่อสารระหว่างเครื่องสามารถเกิดขึ้นได้จะต้องมีเครื่องเครื่องหนึ่งซึ่งทำหน้าที่บอกตำแหน่งของเครื่องปลายทางได้ โดยทั่วไปจะเป็นเครื่องไฟล์เซอร์เวอร์ จากนั้นเครื่องต้นทางจะเริ่มทำการส่งข้อมูล ถ้าเป็นการส่งแบบ เดต้าแกรม (Datagram) เฟรมจะถูกกำหนดตำแหน่งที่อยู่และส่งไปยังปลายทางโดยไม่มีการยืนยันหรือการตรวจสอบ เฟรมนั้นอีกเลย ในกรณีที่ใช้ เซสชัน (Session) การเชื่อมต่อจะยืนยันการส่งข่าวสารนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การส่งและการรับข้อมูล ในแต่ละโปรโตคอลนั้นพัฒนาขึ้นเพื่อให้เครื่องต้นทาง และเครื่องปลายทางสามารถทำการรับส่งข่าวสารกัน ข้อกำหนดของแต่ละโปรโตคอลโดยทั่วไปจะเป็นความยาวของแพคเกจที่ต้องการส่ง
- การยกเลิกการสื่อสาร จัดการยกเลิกการสื่อสารเมื่อสิ้นสุดการสื่อสาร

5.4 โนเวลล์เนตแวร์ 286

ในปัจจุบันนี้เนตแวร์ถือได้ว่าเป็นระบบปฏิบัติการณ์เครือข่ายที่ได้รับความนิยมมากที่สุด มากกว่าครึ่งหนึ่งของระบบเครือข่ายคอมพิวเตอร์ขนาดท้องถิ่นเป็นเนตแวร์ ความนิยมนี้เนื่องมาจากเนตแวร์ทำงานได้เป็นอย่างดี สามารถทำงานกับฮาร์ดแวร์ได้หลากหลาย และยังมีการใช้ระบบรักษาความปลอดภัยที่เหมาะสม นอกจากนี้เนตแวร์ยังมีการเตรียมอุปกรณ์และซอร์ฟแวร์ สำหรับการใช้งานเครือข่ายไว้ให้มากเพียงพอ

5.4.1 เนตแวร์เซอร์เวอร์และระบบแฟ้มข้อมูล

ในไฟล์เซอร์เวอร์ของเนตแวร์ 286 จะมีการจัดรูปแบบฮาร์ดดิสต์ ซึ่งเป็นรูปแบบเฉพาะของเนตแวร์ โดยที่ผู้ใช้ไม่สามารถเข้าไปดูข้อมูลในดิสต์ได้โดยตรง แต่ผู้ใช้ระบบจะใช้งานไฟล์เซอร์เวอร์เสมือนไดรฟ์อีกไดรฟ์หนึ่ง ที่เป็นเช่นนี้ได้เนื่องจากการเปลี่ยนทิศทางการทำงานของระบบปฏิบัติการณ์ รูปแบบการจัดเก็บข้อมูลของไฟล์เซอร์เวอร์ของโนเวลล์จะมีการจัดเก็บรายละเอียดของแฟ้มข้อมูลและซับไดเรคทอรีมากกว่าระบบปฏิบัติการณ์บนเครื่องคอมพิวเตอร์ส่วนบุคคล เช่น การบันทึกเจ้าของแฟ้มข้อมูล การยอมให้แฟ้มข้อมูลนั้นมีการใช้ร่วมกันได้หรือไม่ ฯลฯ

5.4.2 การรักษาความปลอดภัยของเนตแวร์

โนเวลล์เนตแวร์มีระบบการรักษาความปลอดภัยอยู่ด้วยกัน 4 ประเภท เพื่อจำกัดหรือการอนุญาติการเข้าถึงไฟล์เซอร์เวอร์ ไดเรคทอรี และ แฟ้มข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การกำหนดชื่อเฉพาะของผู้ใช้แต่ละคนและรหัสผ่าน สำหรับการเข้าถึงไฟล์เซอร์เวอร์
- การให้สิทธิกับผู้อื่น คือ การอนุญาติ หรือ ไม่อนุญาติ และ กำหนดขอบเขตการเข้าถึงในแต่ละไดเรคทอรี
- ในแต่ละไดเรคทอรีจะมีการจำกัดการรักษาความปลอดภัยเฉพาะแต่ละไดเรคทอรี
- แฟ้มข้อมูลสามารถกำหนดให้มีการอ่านได้เพียงอย่างเดียว เพื่อป้องกันการแก้ไขแฟ้มข้อมูล

5.4.3 เนตแวร์เวิร์คสเตชัน (NetWare Workstation)

ซอร์ฟแวร์แต่ละโปรแกรมของเนตแวร์ที่จำเป็นต้องมีการเรียกใช้บนเครื่องเวิร์คสเตชัน เป็น โปรแกรมชนิดฝังในหน่วยความจำ (TSR program) โปรแกรม IPX ทำหน้าที่จัดการการติดต่อสื่อสารระหว่าง เครื่องคอมพิวเตอร์ กับ เครื่องคอมพิวเตอร์ หรือ เครื่องคอมพิวเตอร์ กับ ไฟล์เซอร์เวอร์ โดยการใช้ IPX / SPX โปรโตคอล โปรแกรม NET2 , NET3 , NET4 คือเชลล์ที่ทำหน้าที่ตรวจสอบการเรียกใช้งานแฟ้มข้อมูลโดยจะมีการเรียกใช้งาน IPX อีกครั้งหนึ่ง โดยการทำงานของซอร์ฟแวร์เหล่านี้ทำให้การติดต่อกับไฟล์เซอร์เวอร์เหมือนกับการติดต่อกับระบบปฏิบัติการโดยตรง

5.5 โนเวลล์เนตแวร์ 386

เนตแวร์ 386 จะมีการใช้ขีดความสามารถของหน่วยประมวลผล 80386 ในการเพิ่มประสิทธภาพของเนตแวร์ เช่น สามารถใช้หน่วยความจำขนาด 4 จิกะไบต์สำหรับแคช เพียงพอสำหรับการทำงานกับผู้ใช้งานระบบพร้อมกัน 250 คน และสามารถใช้ดิสก์ขนาด 32 เทราไบต์ (Terabyte) บนเครื่องเซอร์เวอร์ได้อีกด้วย แฟ้มข้อมูลแต่ละแฟ้มสามารถมีขนาดได้ถึง 4 จิกะไบต์ สามารถเปิดแฟ้มข้อมูลพร้อมกันได้ถึง 100000 แฟ้มข้อมูล

5.6 เดต้าแกรมและเซสชัน (Datagram and Session)

การติดต่อสื่อสารทั้งระหว่างเครื่องคอมพิวเตอร์ กับ เครื่องคอมพิวเตอร์ และระหว่างเครื่องคอมพิวเตอร์ กับ เซอร์เวอร์ เป็นเดต้าแกรมและเซสชันเดต้าแกรม คือ การส่งข่าวสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ฝ่ายรับจะไม่ตอบรับ การรับข่าวสาร (ถ้าต้องการการยืนยันการส่งข่าวสาร ฝ่ายรับเมื่อได้รับข่าวสารแล้วจะต้องส่งข่าวสารตอบรับไปยังผู้ส่ง) ดังนั้นถ้าใช้เดต้าแกรมผู้ส่งและผู้รับจะต้องมีโปรโตคอลยืนยันการสื่อสาร เนื่องจากในกรณีที่ต้องการส่งข่าวสาร ระหว่างนั้นมีเครื่องอื่นทำการส่งอยู่ด้วย ทำให้ข่าวสารเกิดการชนขึ้น ดังนั้นถ้าไม่มีการตรวจสอบข่าวสารนั้นก็จะหายไป ในกรณีกลับกัน เซสชันจะเป็นการเชื่อมต่อกันทางลอจิกคอลระหว่างเครื่องสองเครื่อง ซึ่งข่าวสารที่จะทำการส่งจะได้รับการยืนยันการส่ง อย่างไรก็ตามการใช้เดต้าแกรมจะเร็วกว่าการใช้เซสชัน เนื่องจากการใช้เซสชันจะต้องมีการยืนยันการส่งทุกครั้งที่ได้รับข่าวสาร

5.7 IPX (Internetwork Packet Exchange)

เป็นโปรโตคอลของบริษัท โนเวลล์เนตแวร์ ซึ่งได้รับการปรับปรุงมาจาก XNS (Xerox Network Standard) ซึ่งเป็นของบริษัท ซีรอกส์ IPX จะเป็นการส่งข่าวสารแบบเดต้าแกรม อาจเทียบได้กับ ระดับชั้นเนทเวิร์คของ OSI ทำหน้าที่กำหนดตำแหน่งที่อยู่ การกำหนดเส้นทาง และ การส่งข่าวสารไปยังจุดหมาย การส่งแบบเดต้าแกรมจะไม่ยืนยันการส่ง อย่างไรก็ตามทางบริษัท โนเวลล์เนตแวร์ กล่าวยืนยันการส่งของ IPX ว่าสามารถรับได้ถูกต้องประมาณร้อยละ 95

5.7.1 ตำแหน่งที่อยู่ที่ IPX ใช้งาน

ตำแหน่งที่อยู่ที่ใช้โดยของแพคเกจ IPX จะเป็นตำแหน่งที่อยู่สำหรับการเชื่อมต่อเครือข่ายขนาดใหญ่ มีส่วนประกอบสำคัญ 3 ส่วน คือ หมายเลขเครือข่าย ตำแหน่งที่อยู่ของโหนด และ ซอกเกต

- หมายเลขเครือข่าย (Network number) เป็นหมายเลขที่ใช้ในการแบ่งแยกเครือข่ายออกจากเครือข่ายขนาดใหญ่ ผู้บริหารเครือข่ายจะเป็นผู้กำหนดหมายเลขดังกล่าวในขณะที่ทำการติดตั้งระบบ
- ตำแหน่งที่อยู่ของโหนด (Node address) จะเป็นหมายเลขประจำของวงจรการเชื่อมต่อกับเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ซอกเกต (Socket) คือหมายเลขแสดงตำแหน่งปลายทางบนโปรแกรมผู้ใช้งาน โปรแกรมจะต้องทำการเปิดและปิดซอกเกต เช่นเดียวกับเปิดและปิดแฟ้มข้อมูล

5.7.2 การให้บริการของ IPX

ผู้ใช้สามารถจัดการการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ กับ เครื่องคอมพิวเตอร์ โดยการเลือกใช้การบริการของ IPX การให้บริการเหล่านี้จะมีรูปแบบข้อมูลอยู่ 3 ประเภท ที่จะต้องแจ้งให้ IPX ทราบ คือ

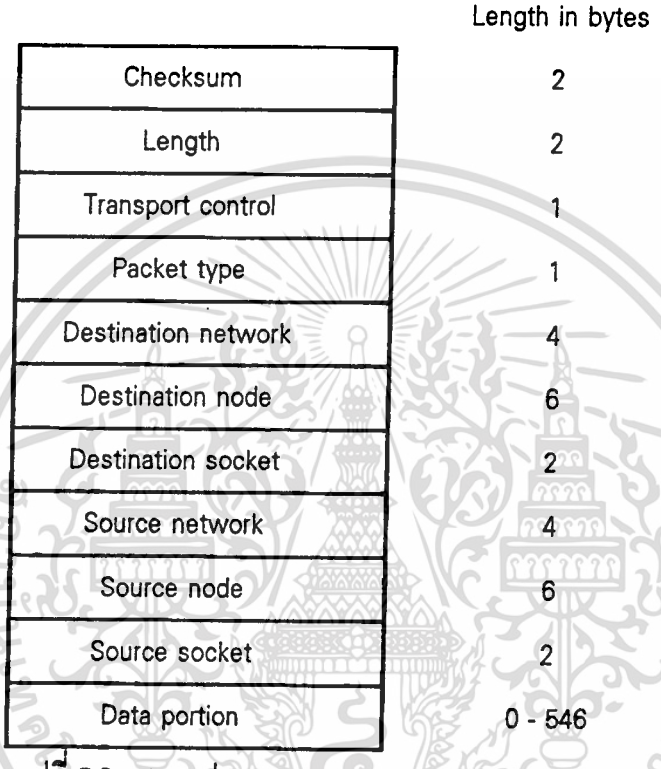
- ส่วนหัวของ IPX จะประกอบไปด้วยข้อมูลขนาด 30 ไบท์
- เรคอร์ดข้อมูลหรือข่าวสารที่ต้องการส่งหรือรับ
- บล็อคควบคุมเหตุการณ์ (ECB : Event Control Block) เป็นส่วนที่แสดงประเภทของข้อมูลให้ IPX ทราบ

5.7.3 การเริ่มต้นและการสิ้นสุดการทำงาน

เมื่อเครื่องคอมพิวเตอร์ 2 เครื่องต้องการส่งข่าวสารถึงกันโดยใช้ IPX จะเริ่มด้วยการเปิด ซอกเกต โดยการใช้ฟังก์ชัน IPXOpenSocket หมายเลขซอกเกตที่ทำการเปิดโดยเครื่อง A ซึ่งเป็นหมายเลขที่มีการตกลงไว้ล่วงหน้าและในกรณีกลับกันก็เช่นเดียวกัน เครื่องทั้งสองจำเป็นต้องรู้ตำแหน่งที่อยู่ของที่หมายปลายทางของเครื่องอีกเครื่องหนึ่ง โปรแกรมจะต้องทำการตรวจสอบหมายเลขเนตเวิร์ค และตำแหน่งที่อยู่ของ โหนด สิ่งที่รู้ทันทีคือหมายเลขประจำตัวของเครื่อง ในการแปลงหมายเลขประจำตัวของเครื่องให้เป็นหมายเลขเนตเวิร์คและตำแหน่งที่อยู่ของโหนดจะใช้ฟังก์ชัน GetObjectConnectionNumbers และ GetInternetworkAddress สิ่งที่จะต้องระมัดระวังในการเรียกงาน คือ เนตแวร์ ยอมให้ผู้ใช้งานคนหนึ่ง สามารถใช้งานระบบได้พร้อมกันจากหลาย ๆ เครื่อง เพื่อเป็นการป้องกันปัญหาดังกล่าว ในการพัฒนาโปรแกรมให้ใช้เฉพาะส่วนแรกของลิสต์ที่ระบบส่งมาให้ และ ไม่สนใจส่วนที่เหลือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.7.4 รูปแบบของ IPX แพคเกจ



รูปที่ 5.3 แสดงรูปแบบของ IPX แพคเกจ

Checksum ข้อมูลในส่วนนี้บริษัทซีรอกส์ ใช้ในโปรโตคอล XNS และยังมีปรากฏในโปรโตคอล IPX อย่างไรก็ตามในโปรโตคอลชั้นล่างได้มีการจัดการส่วนนี้อยู่แล้ว โดยปกติผู้ใช้ไม่ต้องจัดการในส่วนนี้ IPX จะตั้งค่าไว้ที่ 0xFFFF

Length ขนาดของแพคเกจที่สมบูรณ์ คือ ความรวมของส่วนหัวและส่วนข้อมูลขนาดเล็กที่สุดคือ 30 ไบต์ ขนาดใหญ่ที่สุดคือ 576 ไบต์

Transport Control IPX แพคเกจสามารถส่งข้อมูลผ่านเนตแวร์บริดซ์ได้ทั้งหมด 16 ตัว เมื่อแพคเกจเริ่มต้นส่งค่าส่วนนี้จะเป็น 0 และค่าจะเพิ่มขึ้นเรื่อย ๆ เมื่อผ่านแต่ละเครื่อง ถ้าส่วนนี้มีค่าเป็น 16 แพคเกจนี้จะถูกยกเลิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Packet Type	ถ้าต้องการใช้ IPX จะให้ค่าส่วนนี้เป็น 4 และถ้าต้องการใช้ SPX จะให้ค่าส่วนนี้เป็น 5
Destination Network	ทำหน้าที่บอกตำแหน่งจุดหมายของแพคเกจ เป็นส่วนแรกจาก 3 ส่วนที่ผู้ใช้จะต้องกำหนดลงไป
Destination Node	ทำหน้าที่บอกตำแหน่งจุดหมายของแพคเกจ โดยจะเป็นหมายเลขประจำวงจรของวงจรเชื่อมต่อระบบเครือข่าย เป็นส่วนที่ 2 ที่ผู้ใช้จะต้องกำหนด
Destination Socket	เป็นส่วนสุดท้ายที่ผู้ใช้จะต้องกำหนด ซอกเกตจะต้องถูกเปิดโดยโปรแกรมที่ทำงานที่เครื่องที่จะส่งไป หมายเลขซอกเกตที่ใช้ได้จะอยู่ในช่วง 0x4000 ถึง 0x8000
Source Network	หมายเลขเครือข่ายของแพคเกจที่ส่งมา
Source Node	หมายเลขประจำวงจรเชื่อมต่อเครือข่ายที่ทำการส่งแพคเกจ
Source Socket	แพคเกจถูกส่งมาจากซอกเกตหมายเลขนี้
Data Portion	ส่วนของข้อมูลหรือข่าวสารจะอยู่ในบริเวณนี้ ข่าวสารมีความยาวได้ตั้งแต่ 0 ถึง 546 ไบต์

5.8 SPX (Sequenced Packet Exchange)

SPX เป็นการติดต่อสื่อสารแบบเซสชัน ก่อนที่ SPX จะทำการรับหรือส่งแพคเกจจะต้องเริ่มด้วยการสร้างการเชื่อมต่อระหว่างเครื่องทั้งสองที่ต้องการส่งข่าวสาร (SPX จะกำหนดหมายเลขการเชื่อมต่อให้ทั้งสองฝั่ง) ภายหลังจากการเชื่อมต่อแล้วข่าวสารจะได้รับการยืนยันการส่งทุกครั้ง และยังยืนยันว่าข่าวสารจะถึงตามลำดับก่อนหลังถูกต้อง (ในกรณีที่มีการส่งแพคเกจหลาย ๆ แพคเกจ) การทำงานของ SPX สามารถเทียบได้กับชั้นทรานสปอร์ต (Transport Layer) และมีลักษณะบางอย่างเหมือนกับคุณสมบัติของชั้นเซสชัน (Session Layer) ตามรูปแบบของ OSI

5.8.1 การเริ่มต้นและการสิ้นสุดการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องคอมพิวเตอร์แต่ละเครื่องจะเริ่มการติดต่อโดยใช้ SPX โดยการเรียกฟังก์ชัน SPXInitialize ถ้าฟังก์ชันดังกล่าวตรวจสอบพบว่า ไม่มีการติดตั้ง SPX โปรแกรมจะต้องหยุดการทำงาน หลังจากนั้นแต่ละเครื่องจะทำการเปิดซอกเกตอย่างน้อยหนึ่งซอกเกตโดยการเรียกฟังก์ชัน IPXOpenSocket ซอกเกตที่ทำการเปิดจะเป็นซอกเกตที่กำหนดโดยโปรแกรม การที่จะสร้างการเชื่อมต่อ เครื่องคอมพิวเตอร์ A จะเรียกฟังก์ชันการทำงาน SPXListenForConnection ในซอกเกตที่มีการเปิดไว้ เครื่องคอมพิวเตอร์ B จะเรียกฟังก์ชันการทำงาน SPXEstablishConnection โดยจะมีการระบุตำแหน่งที่อยู่ของเครื่อง A

ถ้าการเรียกใช้งานฟังก์ชัน SPXEstablishConnection มีการเรียกใช้งาน โดยที่การลองพยายามสร้างการเชื่อมต่อทุกครั้งไม่สำเร็จคือ ยังไม่มีการเรียกใช้งานฟังก์ชัน SPXListenForConnection การสร้างการเชื่อมต่อจะถูกยกเลิก

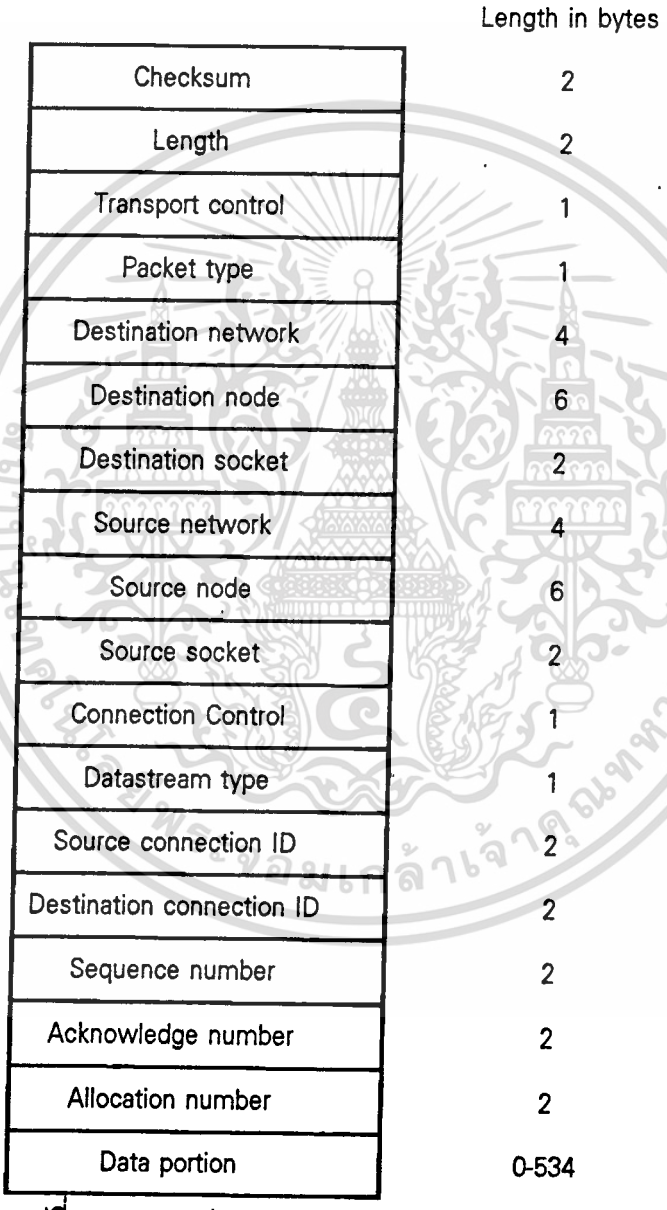
จากนั้นเครื่องคอมพิวเตอร์ทั้งสองเครื่องจะต้องสร้างการเชื่อมต่อโดยการเรียกใช้ฟังก์ชัน SPXListenForSequencedPacket มีเหตุผล 3 ประการที่จะต้องดำเนินขั้นตอนนี้

- SPX ใช้ผลที่ได้รับจากการเรียกใช้ฟังก์ชันดังกล่าวในการสร้างการเชื่อมต่อ
- จะมีการเตรียมหน่วยความจำสำหรับ SPX เพื่อให้ SPX สามารถรับและจัดลำดับคิวของข่าวสารได้
- SPX ในบางครั้งมีการยืมหน่วยความจำชั่วคราวสำหรับแพคเกจนั้นเพื่อใช้สำหรับการทำงานของตัวเอง

ขณะที่เริ่มสร้างการติดต่อ SPX จะมีการกำหนดหมายเลขการเชื่อมต่อให้สำหรับเครื่องแต่ละเครื่อง โปรแกรมของเครื่องจะใช้หมายเลขดังกล่าวเป็นหมายเลขอ้างอิงในการติดต่อข่าวสาร ในกรณีที่สิ้นสุดการติดต่อ เครื่องจะต้องเรียก SPXTerminateConnection SPX จะทำการส่งแพคเกจ เมื่อฝ่ายรับได้รับโปรแกรมจะต้องทำการยกเลิกการติดต่อ และ เครื่องคอมพิวเตอร์ทั้งสองจะต้องทำการปิดซอกเกตทุกซอกเกตที่ทำการเปิดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.8.2 รูปแบบแพคเกจของ SPX



รูปที่ 5.4 แสดงรูปแบบของ SPX แพคเกจ

ในส่วนของ Connection control เป็นฟิลล์ที่ใช้โดย SPX และโดยผู้ใช้งานมีรูปแบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

S A ! E X X X X

บิท S (System packet flag) เป็นบิทแสดงสถานะของแพคเกจ บิท A (Acknowledgement required bit) เป็นการแสดงว่าต้องการการตอบรับของแพคเกจ บิท ! (Attention bit) คือบิทแจ้งให้เตรียมพร้อม บิท E (End-of-message bit) เป็นบิทแสดงการสิ้นสุดของข่าวสาร บิทที่เหลือ 4 บิทไม่ได้มีการระบุการใช้งานไว้

Datastream type	ส่วนนี้จะมีการใช้ทั้งผู้ใช้และ SPX ผู้ใช้สามารถกำหนดค่าในส่วนนี้ได้ตั้งแต่ 0 ถึง 253 (0xFD) SPX จะใช้ค่า 254 และ 255 โดยที่เมื่อเครื่อง A ต้องการยกเลิกการติดต่อจะส่งค่า 254 ไปยังเครื่องปลายทาง และเมื่อเครื่องปลายทางรับทราบการยกเลิกการติดต่อจะทำการส่งค่านี้เป็น 255
Source connection ID	SPX จะเป็นผู้กำหนดค่าในส่วนนี้ จะเป็นส่วนบอกหมายเลขการเชื่อมต่อของเครื่องที่ทำการส่งแพคเกจ
Destination ID	SPX จะเป็นผู้กำหนดค่าในส่วนนี้ จะเป็นส่วนบอกหมายเลขการเชื่อมต่อของเครื่องที่จะได้รับแพคเกจ
Sequence number	SPX จะเป็นผู้กำหนดค่าในส่วนนี้ จะเป็นส่วนที่ใช้บอกหมายเลขของแพคเกจ และใช้ในการยกเลิกแพคเกจเมื่อมีการส่งแพคเกจซ้ำ
Acknowledge number	SPX จะเป็นผู้กำหนดค่าในส่วนนี้ สำหรับเก็บหมายเลขของแพคเกจที่คาดว่าจะได้รับแพคเกจต่อไป
Allocation number	SPX จะเป็นผู้กำหนดค่าในส่วนนี้ สำหรับเก็บจำนวนของแพคเกจที่ได้ทำการส่งไปแล้วแต่ยังมิได้รับการตอบรับ
Data portion	ใช้สำหรับเก็บข้อมูลที่ต้องการส่ง ขนาดสูงสุดคือ 534 ไบต์ ซึ่งจะน้อยกว่าของ IPX เนื่องจากถูกใช้เป็นส่วนหัวของ SPX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
วิจารณ์และสรุปผล

6.1 ปัญหาในการทำงาน

6.1.1 เรื่องการจองหน่วยความจำ

เนื่องจากต้องใช้หน่วยความจำในการเก็บคำศัพท์ในภาษาอังกฤษจำนวนมากทำให้ต้องนำหน่วยความจำเสริมส่วน XMS มาใช้ แต่หน่วยความจำส่วนดังกล่าวเป็นส่วนที่ดอสไม่ได้สนับสนุนการใช้งานเป็นพิเศษเพราะฉะนั้นการอ้างถึงจึงไม่สะดวกมากนัก ความสามารถในการจัดการหน่วยความจำส่วน XMS จึงเพียง

- การสั่งจองหน่วยความจำ โดยระบุหน่วยความจำที่จะจองมีหน่วยเป็นกิโลไบต์ ซึ่งไม่สามารถจะจองในปริมาณที่ไม่เต็มหน่วยไม่ได้
- การอ้างถึงหน่วยความจำ สำหรับการเขียน ต้องเขียนตั้งแต่เริ่มต้นขอบเขตที่จอง ไม่สามารถระบุให้เขียนในตำแหน่งส่วนหนึ่งส่วนใดๆ ของขอบเขตที่จองได้
- การอ้างถึงหน่วยความจำ สำหรับการอ่าน ต้องอ่านมาทั้งบล็อคของหน่วยความจำที่จองไว้ ไม่สามารถระบุให้อ่านในตำแหน่งส่วนหนึ่งส่วนใดๆ ของขอบเขตที่จองได้
- การตรวจการทำงานของโปรแกรม (Debug) ไม่สะดวกเพราะคำสั่งที่ใช้ในการจัดการไม่สามารถสั่งทำงานแล้วดูผลการทำงานที่ละคำสั่งได้
- การถามหน่วยความที่เหลืออยู่

สำหรับการจัดการกับส่วนหน่วยความจำหลัก

- บางกรณีไม่สามารถใช้หน่วยความจำเสริมได้ต้องใช้หน่วยความจำหลัก
- หน่วยความจำหลักมีขนาดจำกัด
- การจองหน่วยความจำหลักมาก ๆ ทำให้ไม่สามารถทดสอบผลการทำงาน บนภาษาที่เขียนที่ใช้ในการพัฒนาได้ (บนภาษา C)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.2 ส่วนของการติดต่อสื่อสาร

- สิทธิบนระบบสื่อสารที่ได้มีขีดจำกัด ทำให้ไม่สามารถทดสอบการทำงานของโปรแกรมได้
- ระบบสื่อสาร มีโปรแกรมระบบควบคุมต่างๆ กันไป ซึ่งมาตรฐานข้อกำหนดบางอย่างที่ต่างกันทำให้ไม่สามารถรับรองว่า โปรแกรมที่เขียนจะใช้ได้กับทุกระบบสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. ธันวา ศรีประโมง, "การโปรแกรมภาษาซี สำหรับวิศวกรรม",วิทยาลัยมหานคร, 292 หน้า, 2536
2. Kent Porter, "Stretching Turbo C", Dr. Dobbs'Journal, 431 p., 1988
3. Ray Duncan, Charles Petzold, Andrew Schulman, M.Steven Baker, Ross P.Nelson, Stephen R.Davis, Robert Moote, "Extending DOS ", Addison-Wesley Publishing Company, Inc., 538 p., 1992
4. Robert P.Davidson, Nathan J.Muller, "Internetworking LANs: Operation, Design, and Management", Artech House, Inc., 550 p.,1992
5. Barry Nance, "Network Programming in C", QUE corporation, 648 p.,1990
6. Charles G.Rose, "Programmer's Guide to NetWare", McGraw-Hill, Inc., 933 p.,1990
7. Ben Ezzell, "Graphic Programming in Turbo C 2.0", McGraw-Hill, 568 p., 1988
8. Steve Rimmer, "Graphical User Interface Programming", McGraw-Hill, 423 p.,1992

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้