การต่อเครือข่ายฐานข้อมูลซีดีรอม

CD-ROM NETWORKING

โดย

นางสาวอภิรมณ ดำรงศิริ

นายอภิสิทธิ์ สุขสาคร

ปริญญานิพนธ์ส่วนนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

การต่อเครือข่ายฐานข้อมูลซีดีรอม
CD-ROM NETWORKING

โดย

นางสาวอภิรมณ ดำรงศิริ รหัสประจำตัว 34109485

นายอภิสิทธิ์ สุขสาคร รหัสประจำตัว 34109487

อาจารย์ที่ปรึกษา

ดร. เอื้อน ปิ่นเงิน

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตเจ้าคุณทหาร ลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2537

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การต่อเครือข่ายฐานข้อมูลซีดีรอม

ผู้จัดทำ

1. นางสาวอภิรมณ ดำรงศิริ
2. นายอภิสิทธิ์ สุขสาคร

.....อาจารย์ที่ปรึกษา

(ดร. เอื้อน ปิ่นเงิน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การต่อเครือข่ายฐานข้อมูลซีดีรอม

อภิรมณ ดำรงศิริ
อภิสิทธิ์ สุขสาคร
ดร. เอื้อน ปิ่นเงิน อาจารย์ที่ปรึกษา
ปีการศึกษา 2537

บทคัดย่อ

ในปริญญานิพนธ์ฉบับนี้ เรียบเรียงขึ้นจากผลงานที่พัฒนาขึ้นเพื่อใช้งานกับระบบฐานข้อมูลภายในห้องสมุด โดยแบ่งการทำงานออกเป็นสองส่วน ส่วนแรกเป็นการพัฒนาโปรแกรมขึ้นเพื่อการใช้งานบนระบบเน็ตแวร์ โปรแกรมดังกล่าวเป็นโปรแกรมที่ใช้เพื่อเชื่อมโยงการทำงานกับผู้ใช้ (user interface) เพื่อให้มีการทำงานที่สะดวกขึ้น โดยโปรแกรมนี้จะมีความสามารถในการตรวจสอบและทำการแมปหมวด (volume) ได้โดยอัตโนมัติ ทำให้ผู้ใช้ไม่ต้องทำการแมปหมวดต่าง ๆ เอง รวมทั้งไม่ต้องทำการค้นหาหมวดจากการใช้คำสั่งของระบบเน็ตแวร์ (คำสั่ง volinfo) สำหรับส่วนที่สองจะเป็นการพัฒนาโปรแกรมเพื่อศึกษาการเบี่ยงเบนทิศทางข้อมูลเข้าและออก (redirection) จุดประสงค์หลักในการพัฒนาโปรแกรมส่วนนี้คือเพื่อใช้เป็นแนวทางในการพัฒนาระบบเครือข่ายฐานข้อมูลซีดีรอมในอนาคต ทำให้การทำงานไม่ผูกพันกับระบบเน็ตแวร์ นอกจากนี้ยังสามารถใช้โปรแกรมนี้ในการเชื่อมโยงการทำงานกับผู้ใช้ เพื่อลดการคีย์ข้อมูลที่ไม่จำเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CD-ROM NETWORKING

Apiramon Damrongsiri
Apisit Suksakorn
Dr. Ouen Pinngern Advisor
1994

Abstract

This thesis is a research of CD-ROM NETWORK. It is a case study of how to use CD-ROM in the library more efficiently. The developments are devided into two parts. First, to write a program to control NetWare system. In this part, the program is developed for mapping volume in a file server automatically. If the program is not used, it is necessary for users to learn how to map volume, and how to find the name of the volume they want. So, this program is a kind of user interface that can help user in NetWare environment. Second, is about redirection. Redirection is a method that uses to create CD-ROM NETWORK. In this thesis, the test program is installed on the NetWare system. So this program is only the beginning of how to create our own CD-ROM network.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทที่ 1	บทนำ	1
บทที่ 2	ระบบเครือข่ายซีดีรอม	3
บทที่ 3	ระบบเน็ตแวร์และหลักการในการเขียนโปรแกรมอรรถประโยชน์	16
บทที่ 4	ทฤษฎีในการเบี่ยงเบนทิศทางข้อมูล	26
บทที่ 5	ทฤษฎีในการเขียนโปรแกรม	43
บทที่ 6	ผลการทำงานของโปรแกรม	69
บทที่ 7	บทวิจารณ์และสรุป	70
	ภาคผนวก ก	71
	ภาคผนวก ข	81
	ภาคผนวก ค	132
	กิตติกรรมประกาศ	135
	หนังสืออ้างอิง	136

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่	รายละเอียด	หน้า
2.1	ภาพแสดงการติดต่อสื่อสารผ่านระบบเครือข่ายโดยใช้เครื่องบริการแฟ้ม	4
2.2	ภาพแสดงการติดต่อสื่อสารผ่านระบบเครือข่ายโดยใช้เครื่องบริการซีดีรอมภาพแสดง	5
2.3	การทำงานของเครื่องบริการแฟ้มและซีดีรอม (เช่น SCSI Express)	6
2.4	ภาพแสดงถึงผลิตภัณฑ์ที่ใช้วิธีการเปลี่ยนแปลงทิศทางข้อมูล.ในการสร้างเครื่องบริการซีดีรอม (เช่นออพติเน็ต)	7
2.5	ชนิดที่ 1 : เครื่องบริการซีดีรอม	12
2.6	ชนิดที่ 2 : เครื่องบริการซีดีรอมและเครื่องบริการแฟ้ม	12
2.7	ชนิดที่ 3a : การรวมกันระหว่างเครื่องบริการแฟ้มและเครื่องบริการซีดีรอม	13
2.8	ชนิดที่ 3b : การรวมกันระหว่างเครื่องบริการซีดีรอมและเครื่องบริการแฟ้ม	14
2.9	ชนิดที่ 4 : การทำงานแบบกระจายไม่มีเครื่องบริการ	15
3.1	เครื่องบริการกับสถานีงาน	18
3.2	แสดงความสัมพันธ์ระหว่างเซลล์ของเน็ตแวร์กับดอส	19
3.3	คำขอจะถูกส่งและตอบสนองผ่านระบบเครือข่าย	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่	รายละเอียด	หน้า
4.1	การแบ่งหน้าจอออกเป็นตาราง	29
4.2	ค่าลักษณะเฉพาะในการแสดงสี	30
4.3	ค่าลักษณะเฉพาะของโมโนโครม	31
4.4	โหมดแสดงผลและตำแหน่งที่ใช้	33
4.5	หน่วยความจำแสดงผลบนจอภาพในโหมดข้อความ	34
4.6	แสดงค่าสถานะของปุ่ม toggle และปุ่ม Shift	40
4.7	แสดงเนื้อที่ในหน่วยความจำที่ไบออสใช้ในการจัดการคีย์บอร์ด	41
4.8	บัฟเฟอร์ของแป้นพิมพ์	42
5.1	โครงสร้างของไดเรกทอรีเปรียบเทียบได้กับเอกสารที่เก็บอยู่ในตู้เอกสารซึ่งแบ่งการเก็บออกเป็นลำดับชั้นคือแฟ้มย่อย,แฟ้มใหญ่และลิ้นชักรวมกันอยู่ในตู้เอกสาร	44
5.2	ตารางไดเรกทอรี	46
5.3	แสดงการเรียกใช้แฟ้มในระบบดอสและเน็ตแวร์	47
5.4	ตัวอย่างการเขียนโปรแกรมควบคุมเน็ตแวร์โดยใช้การขัดจังหวะของดอส	55
5.5	แผนภูมิสายงานการเขียนโปรแกรมแมปหมวด	59
5.6	แผนภูมิสายงานแสดงการทำงานหลักของโปรแกรมเบี่ยงเบนทิศทาง	67
5.7	แผนภูมิสายงานแสดงการทำงานเมื่อมีอินเทอร์รัพต์ของสัญญาณนาฬิกา	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมา

ปริญญานิพนธ์ฉบับนี้เป็นงานวิจัยเพื่อการใช้งานในห้องสมุด โดยมีจุดมุ่งหมายเพื่อการพัฒนาการเก็บฐานข้อมูลบนซีดีรอม (CD-ROM) ให้สามารถใช้งานผ่านเครือข่าย (Network) ได้ เพื่อให้มีความสามารถในการรองรับความต้องการจากผู้ใช้ได้หลายคนในเวลาเดียวกัน

การนำเอาซีดีรอมมาใช้งานในระบบเครือข่ายจะทำให้สามารถรองรับการทำงานในด้านต่าง ๆ ได้อย่างมีประสิทธิภาพ โดยระบบเครือข่ายจะเป็นตัวจัดการให้มีการแชร์การใช้งานทรัพยากรต่าง ๆ ในขณะที่เทคโนโลยีของซีดีรอมจะเป็นอุปกรณ์ที่ใช้เก็บและค้นคืน (retrieve) ข้อมูล โดยข้อมูลที่มีการเก็บบนซีดีรอมจะมีขนาดใหญ่ ทั้งนี้เนื่องจากซีดีรอมจะเป็นอุปกรณ์ที่มีความจุสูง

1.2 ประโยชน์ที่ได้รับ

1.2.1 ในแง่ของงานวิจัย

1. เพื่อเป็นแนวทางในการพัฒนาระบบเครือข่ายซีดีรอมของตนเองในอนาคต
2. เพื่อเป็นแนวทางในการพัฒนาระบบฐานข้อมูลในห้องสมุดของตนเองขึ้นมา

1.2.2 ในแง่ประโยชน์ที่ได้รับทั่วไป

1. เพื่อการแชร์แฟ้มและโปรแกรมต่าง ๆ ภายในระบบร่วมกัน
2. เพื่อเพิ่มความเป็นหนึ่งเดียวของข้อมูล
3. ผู้ใช้ทุกคนมีความแน่ใจได้ว่าฐานข้อมูลที่ทุกคนได้รับเป็นอันหนึ่งอันเดียวกัน
4. เพิ่มความปลอดภัยในการทำงาน
5. ทำให้สามารถควบคุมการแชร์ทรัพยากรต่าง ๆ เป็นไปอย่างเข้มงวด
6. เพิ่มผลผลิตในการทำงาน

1.3 ขอบเขตของโครงงาน

การทำงานจะแบ่งออกเป็นสองส่วนคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ศึกษาการเขียนโปรแกรมควบคุมการทำงานของระบบเน็ตแวร์ (NetWare) ซึ่งเป็นโปรแกรมพื้นฐานที่ใช้ในการแชร์การทำงานของอุปกรณ์ต่าง ๆ
2. ศึกษาวิธีการเขียนโปรแกรม ที่ใช้ในการเบี่ยงเบนทิศทางอินพุตและเอาต์พุตของอุปกรณ์ (redirection) เพื่อเป็นพื้นฐานในการสร้างระบบเครือข่ายซีดีรอมของตนเองในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ระบบเครือข่ายซีดีรอม

2.1 หน้าที่พื้นฐานของระบบเครือข่ายซีดีรอม

หน้าที่พื้นฐานของซีดีรอมในระบบเครือข่ายคือการเก็บและทำการค้นคืนข้อมูลไปสู่ผู้ใช้ โดยผู้ใช้จะสามารถใช้งานโปรแกรมหรือค้นข้อมูลที่ต้องการได้อย่างถูกต้อง แต่การทำงานในลักษณะเช่นนี้ได้จะต้องมีการควบคุมระดับการเข้าถึงซีดีรอมในอัตราที่เหมาะสม มิฉะนั้นอาจทำให้ซีดีรอมต้องทำงานเกินกำลัง

2.2 ลักษณะโดยทั่วไป

เครือข่ายของซีดีรอมแบบพื้นฐานจะมีอยู่ 3 ลักษณะคือ

1 . การใช้งานในลักษณะแบบจุดต่อจุด (peer-to-peer)

2. การใช้งานร่วมกันระหว่างเครื่องบริการแฟ้ม (file server) และซีดีรอม (รูปที่ 2.1)

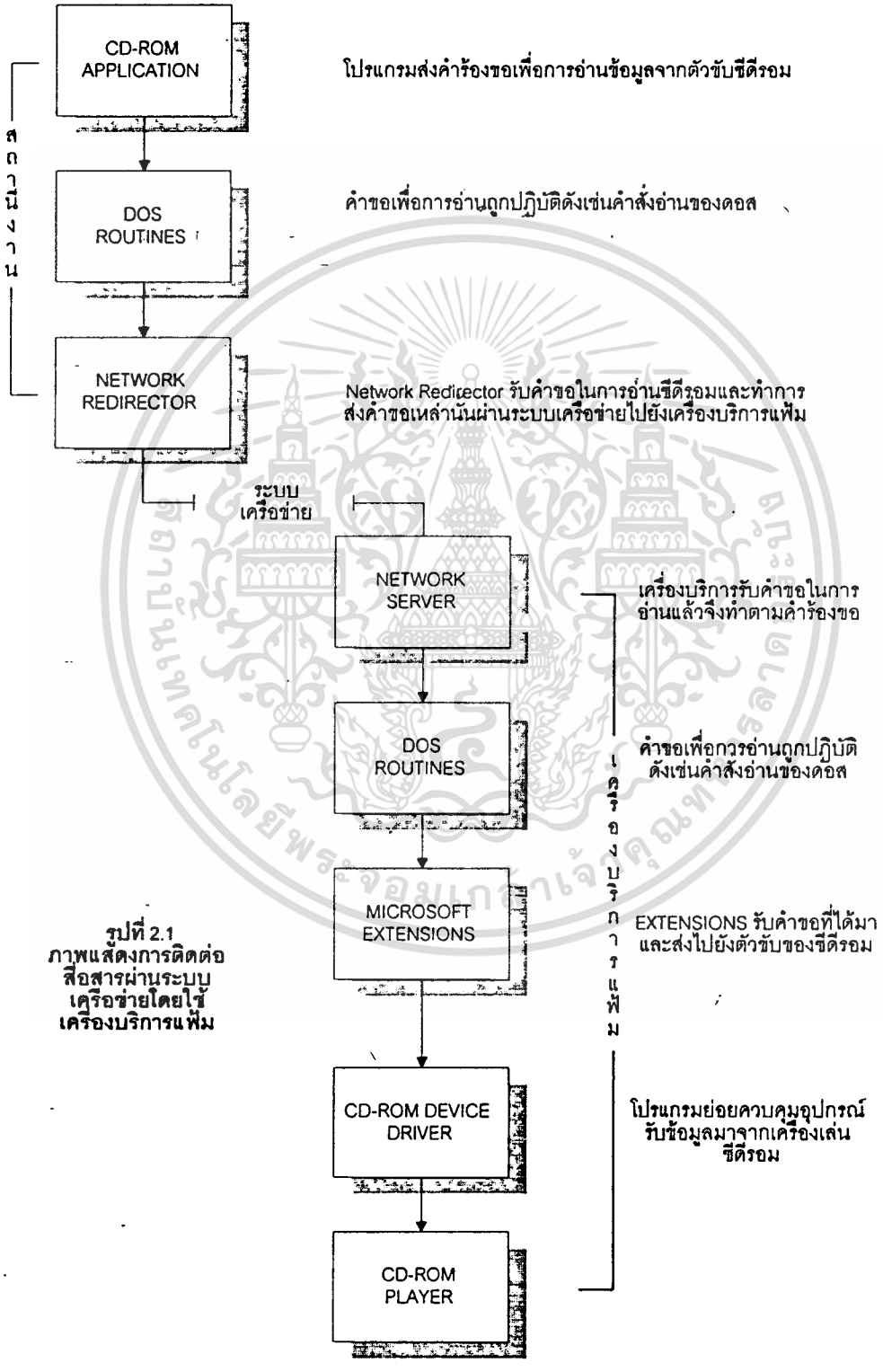
3 . การทำงานในลักษณะที่เป็นเครื่องบริการซีดีรอม (CD-ROM server) เพียงอย่างเดียว (รูปที่ 2.2)

การทำงานในลักษณะแบบจุดต่อจุด เช่นดิสก์พอร์ต (Discport) ของไมโครเทสต์ (Microtest) เหมาะกับการใช้งานบนระบบเครือข่ายที่มีความเรียบง่าย โดยไม่มีการเน้นความสำคัญในเรื่องของความปลอดภัย ใช้ในเครือข่ายขนาดเล็กที่ไม่มีการเคลื่อนย้ายหรือไม่มีการจราจรของข้อมูลสูงมากนัก

สำหรับการใช้เครื่องบริการแฟ้ม การทำงานในลักษณะนี้ เครื่องบริการแฟ้มจะเป็นศูนย์กลาง ทำให้การควบคุมต่าง ๆ ทำได้ดีขึ้น ประสิทธิภาพจึงสูงกว่าแบบแรก ถ้ามีการเคลื่อนย้ายข้อมูล (swap disk) บ่อยครั้งจะทำให้เกิดภาระในการทำงานด้านนี้มากขึ้น ซึ่งถ้าข้อมูลมีจำนวนมากก็จะทำให้ต้องการหน่วยความจำแคช (cache memory) สำหรับงานในด้านนี้ (งานด้านการเคลื่อนย้ายข้อมูล) และในบางครั้งก็อาจจะทำให้ต้องใช้เครื่องบริการแฟ้มทำการควบคุมการควบคุมการทำงานของซีดีรอมเพียงอย่างเดียว เพื่อให้ได้รับผลในการทำงานที่ได้รับมีประสิทธิภาพสูงสุด

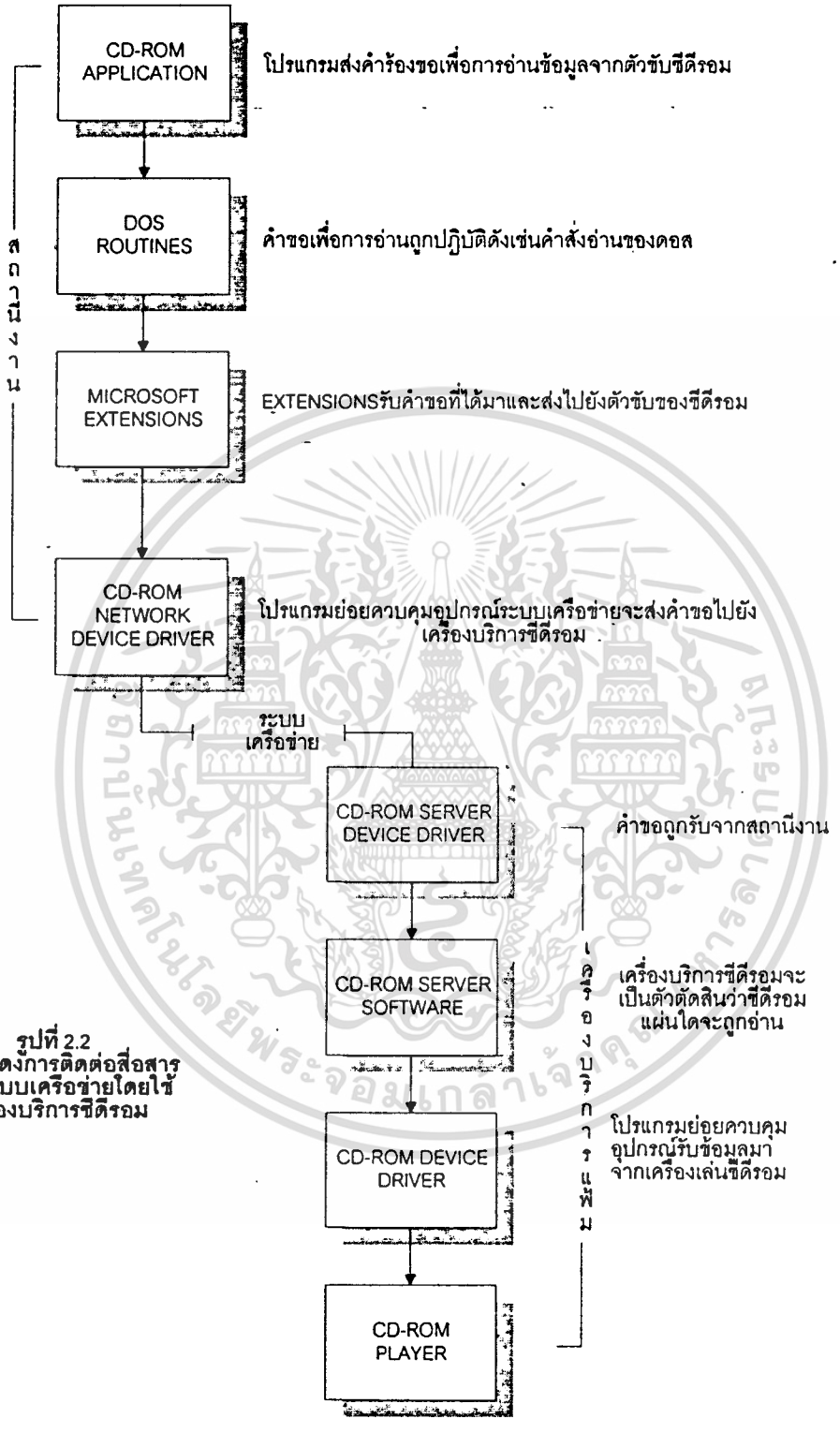
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลิตภัณฑ์บางชนิดเช่นสกาซี เอกซ์เพรส 600 ดีเอกซ์ (SCSI Express 600CDX : รูปที่ 2.3) จากไมโคร ดีไซน์ อินเทอร์เนชันแนล (Micro Design International) จะมีการทำงานในลักษณะที่



รูปที่ 2.1
ภาพแสดงการติดต่อสื่อสารผ่านระบบเครือข่ายโดยใช้เครื่องบริการแฟ้ม

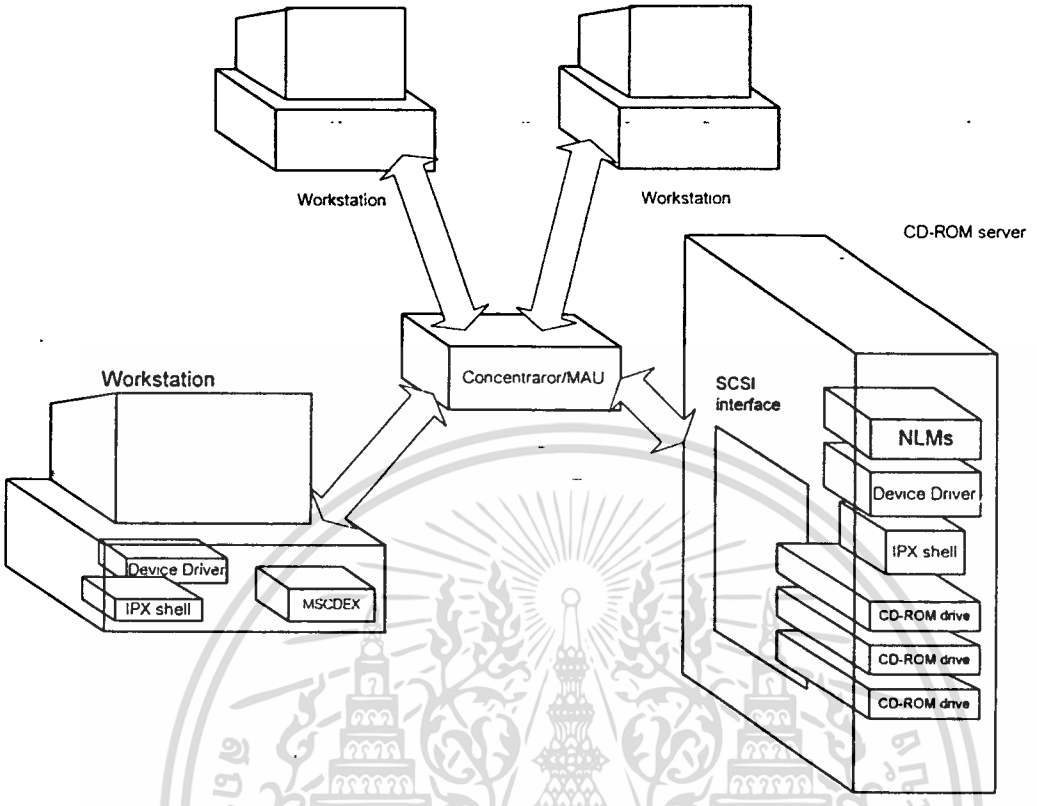
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2
ภาพแสดงการติดต่อสื่อสารผ่านระบบเครือข่ายโดยใช้เครื่องบริการซีดีรอม

ทำการแชร์เครื่องบริการแฟ้ม แต่ประสิทธิภาพที่ได้จากการทำงานของผลิตภัณฑ์นี้จะสูงกว่าการทำงานที่ได้จากเครื่องบริการซีดีรอมเพียงอย่างเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 **ภาพแสดงการทำงานของเครื่องบริการแฟ้มและซีดีรอม (เช่น** SCSI Express)

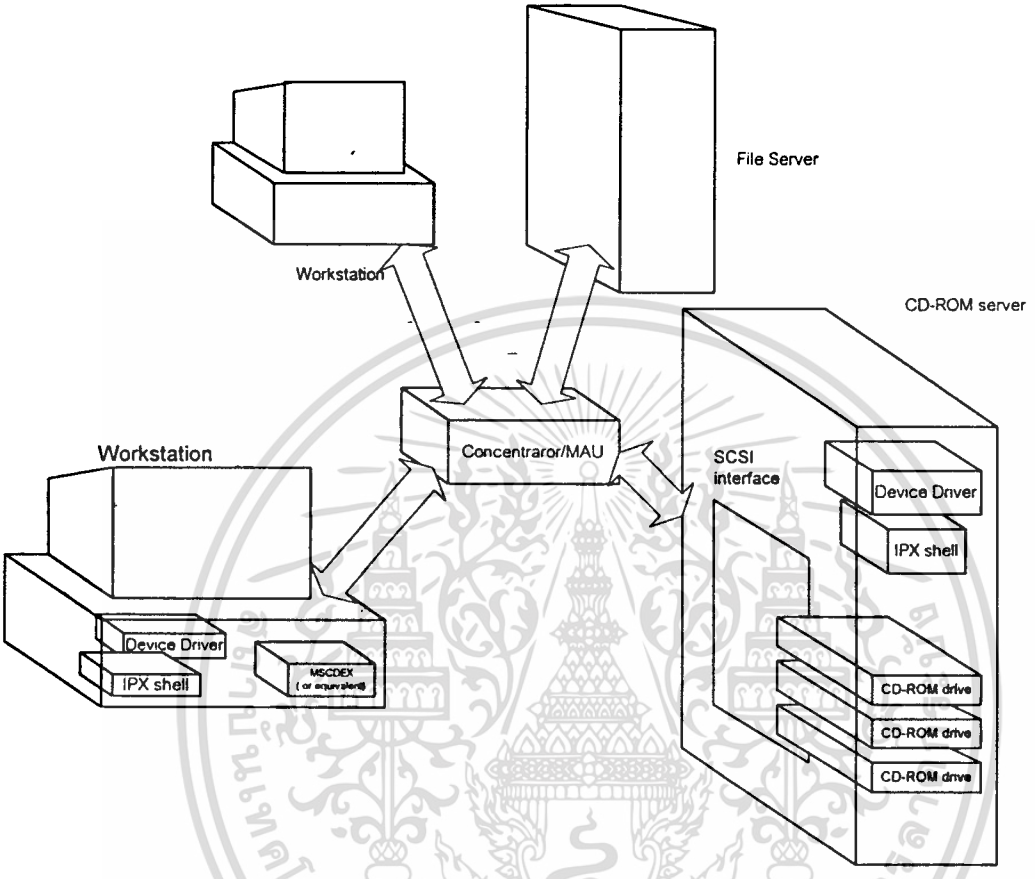
สำหรับผลิตภัณฑ์บางชนิดจะใช้วิธีการเปลี่ยนแปลงทิศทางของข้อมูล (redirection) เพื่อการเปลี่ยนแปลงเครื่องคอมพิวเตอร์ส่วนบุคคลให้กลายเห็นเครื่องบริการซีดีรอม เช่นออพติเน็ต (Opti-Net : รูปที่ 2.4) จากออนไลน์ คอมพิวเตอร์ ซิสเต็ม (Online Computer System) ระบบประเภทนี้ส่วนมากมักจะมีการจัดการที่ดีและมีระบบความปลอดภัยสูง

2.3 ข้อดีของซีดีรอม

1. ตัวขับของซีดีรอมสามารถใช้อ่านข้อมูลที่มีความจุสูงและสามารถทำการเปลี่ยนข้อมูลง่าย ซึ่งทำโดยการเปลี่ยนแผ่นที่ทำการใช้งาน
2. ในปัจจุบันรูปแบบของการเก็บข้อมูลบนซีดีรอมบางโปรแกรมจะมีลักษณะที่สามารถใช้งานกับเครือข่ายได้เลย ดังนั้นจึงเป็นข้อควรระวังในการซื้ออย่างหนึ่งที่ควรจะตรวจสอบให้เรียบร้อยว่าซีดีรอมที่ซื้อมาสำหรับผู้ใช้กี่คน
3. รูปแบบในการเก็บข้อมูลของซีดีรอมจะมีมาตรฐานในการเก็บขอมูลแบบเดียวกันคือ ISO 9660
4. ซีดีรอมมีความทนทานในการเก็บข้อมูล มีอายุในการใช้งานสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ราคาของซีดีรอมเมื่อเทียบกับคุณภาพจะถือได้ว่ามีราคาต่ำ



รูปที่ 2.4 ภาพแสดงถึงผลิตภัณฑ์ที่ใช้วิธีการเปลี่ยนแปลงทิศทางข้อมูลในการสร้างเครื่องบริการซีดีรอม (เช่นออพติเน็ต)

2.4 ซอฟต์แวร์สำหรับระบบเครือข่าย

ในกรณีที่โปรแกรมที่ใช้งานบนซีดีรอมไม่สามารถทำการแชร์ข้อมูลในระบบเครือข่ายได้ ทางแก้ปัญหาทางหนึ่งก็คือการใช้ซอฟต์แวร์ต่าง ๆ ที่จะกล่าวต่อไปนี้เพื่อใช้เพิ่มความสามารถในการแชร์ข้อมูลและอุปกรณ์ต่าง ๆ

2.4.1. โนเวล แอดวานซ์ เน็ตแวร์ (Novell Advanced NetWare)

แอดวานซ์ เน็ตแวร์เป็นซอฟต์แวร์ที่ใช้งานกับข่ายคอมพิวเตอร์เฉพาะบริเวณ (LAN) ที่ได้รับความนิยมมากที่สุด นอกจากนี้ในการทำงานของโนเวลจะต้องมีเครื่องบริการแฟ้ม ซึ่งถูกใช้ในควบคุมการทำงานด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับโนเวล 3.1x และ 4.x จะมีความสามารถพิเศษคือสามารถแชร์การทำงานของซีดีรอมได้อีกด้วย

2.4.2. ทรีคอม (3COM)

เป็นซอฟต์แวร์ที่มีความสามารถและได้รับความนิยมเช่นเดียวกับเน็ตแวร์

2.4.3. แลนทาสติก (LANtastic)

เป็นซอฟต์แวร์ที่มีราคาต่ำ สามารถใช้งานร่วมกับซีดีรอมได้ แลนทาสติกจะช่วยอำนวยความสะดวกในการติดต่อระหว่างโปรแกรมและซีดีรอมโดยการใช้งานชุดคำสั่งมาตรฐานของดอส ดังนั้นจะไม่สามารถใช้งานร่วมกับโปรแกรมที่เชื่อมการทำงานกับไมโครซอฟต์ เอกซ์เทนชันโดยตรงหรือโปรแกรมที่เชื่อมการทำงานกับโปรแกรมย่อยควบคุมอุปกรณ์ (device driver) ของซีดีรอมโดยตรง

2.5 การคำนวณหาแรม (RAM) ที่ต้องใช้ในสถานีงาน

ปัญหาหนึ่งที่มักจะพบในการใช้งานระบบเครือข่ายซีดีรอมก็คือการหาจำนวนแรมที่มากพอเพื่อใช้ในการทำงานของโปรแกรมประยุกต์ของซีดีรอม โดยปกติระบบเครือข่ายจะต้องการแรมเพื่อใช้เก็บโปรแกรมที่ฝังตัวอยู่ นอกเหนือไปจากดอสที่ต้องการแรมอยู่แล้ว 640K สำหรับสถานีงานของระบบเครือข่ายซีดีรอมจะต้องมีดอส ANSI.SYS โปรแกรมย่อยควบคุมระบบเครือข่าย โปรแกรมย่อยควบคุมระบบเครือข่ายของซีดีรอมและไมโครซอฟต์ เอกซ์เทนชัน ซึ่งโปรแกรมที่กล่าวมานี้จะถูกโหลดเข้าไปในหน่วยความจำก่อนที่โปรแกรมสำเร็จประยุกต์ของซีดีรอม (CD-ROM application) จะถูกโหลดเข้าไป

(แต่ในปัจจุบันนี้โปรแกรมฝังตัว (resident program) หรือโปรแกรมย่อยควบคุมบางตัวจะสามารถถูกโหลดขึ้นไปอยู่ในหน่วยความจำยืดขยาย (extended memory) หรือถูกโหลดให้เข้าไปอยู่ในหน่วยความจำขยาย (expanded memory) ได้)

ในการคำนวณหาแรมนั้น ยังมีปัจจัยที่สำคัญอื่น ๆ คือชนิดของสถานีงานและชนิดของระบบเครือข่าย ทั้งนี้เนื่องจากชนิดของอุปกรณ์ต่าง ๆ เหล่านี้จะมีผลต่อการโหลดโปรแกรมให้อยู่นอกเหนือจากหน่วยความจำฐาน (high memory)

ขั้นตอนในการคำนวณ

1. หาแรมที่โปรแกรมสำเร็จประยุกต์ซีดีรอม (CD-ROM application) ต้องการใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขั้นตอนนี้จะต้องมีการคำนวณหา

- จำนวณแรมที่น้อมที่สุดที่โปรแกรมสำเร็จประยุกต์ซีดีรอมต้องการในการทำงาน
- โปรแกรมสำเร็จประยุกต์นี้สามารถใช้งานได้กับไมโครซอฟต์ เอกซ์เทนชันหรือไม่ ?
- โปรแกรมสำเร็จประยุกต์ทำการติดต่อโดยตรงกับไมโครซอฟต์ เอกซ์เทนชันหรือว่าการทำงานจะทำการติดต่อผ่านโปรแกรมย่อยควบคุมอุปกรณ์ซีดีรอม

2. ตรวจหาคุณสมบัติของข่ายคอมพิวเตอร์เฉพาะบริเวณ

- ระบบเครือข่ายมีการทำงานติดต่อโดยตรงกับไมโครซอฟต์ เอกซ์เทนชันหรือว่าการทำงานจะทำการติดต่อผ่านโปรแกรมย่อยควบคุมอุปกรณ์ซีดีรอม
- ภายหลังจากการโหลดระบบต่าง ๆ แล้ว มีแรมเหลือมากน้อยเท่าไรเพื่อใช้ในการทำงานของโปรแกรมสำเร็จประยุกต์ซีดีรอม

ตัวอย่างในการคำนวณแรม

จำนวนแรมทั้งหมด (อย่างน้อยที่สุด)	655 000 ไบต์
ดอส	62 000 *
ANSI.SYS	2 000
โปรแกรมระบบเครือข่าย	56 000**
โปรแกรมย่อยควบคุมอุปกรณ์ซีดีรอม	6 000**
Microsoft Extensions	23 000***
จำนวนแรมที่เหลือ	506 000

* ขนาดของดอสจะขึ้นอยู่กับรุ่น จำนวนบัฟเฟอร์ (buffer) และค่าต่าง ๆ ที่ตั้งไว้ใน CONFIG.SYS

** ขนาดจะมีการแปรเปลี่ยนไปตามผลิตภัณฑ์ (ผลิตภัณฑ์ต่างชนิด ต่างรุ่น จะมีขนาดไม่เท่ากัน

*** รวมบัฟเฟอร์สำหรับตัวขับซีดีรอมเชิงตรรก (logical CD-ROM drive) ถ้าสถานีงานมีตัวขับซีดีรอมเชิงตรรกมากกว่า 1 หน่วย ให้เพิ่มค่า 8 192 ไบต์สำหรับแต่ละตัวขับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการ
034860

3. ทำการเปรียบเทียบผลที่ได้โดยข้อความถัดไปนี้จะต้องเป็นจริง

- แรมคงเหลือในสถานีงานจะต้องเท่ากับหรือมากกว่าความต้องการแรมของโปรแกรมสำเร็จประยุกต์ซีดีรอมที่มีขนาดใหญ่ที่สุด
- ถ้ามีโปรแกรมสำเร็จประยุกต์ที่ทำการติดต่อกับไมโครซอฟต์ เอกซ์เทนชันโดยตรง ระบบเครือข่ายจะต้องรองรับการทำงานของไมโครซอฟต์ เอกซ์เทนชันด้วย
- ถ้ามีโปรแกรมสำเร็จประยุกต์ที่ทำการติดต่อกับโปรแกรมย่อยควบคุมอุปกรณ์ซีดีรอมโดยตรง ระบบเครือข่ายจะต้องรองรับการทำงานของโปรแกรมย่อยควบคุมอุปกรณ์ซีดีรอมด้วย

4. ถ้ามีรูปแบบการทำงานต่าง ๆ เพิ่ม จะต้องมีการสำรองแรมเพิ่มเติมด้วย เช่น

- มีการใช้เมนูหรือไม่
- สถานีงานสามารถใช้ทรัพยากรต่าง ๆ ได้หรือไม่
- ผู้ใช้สามารถเปลี่ยนการทำงานไปดอส (DOS promp) ได้หรือไม่
- มีการจำกัดจำนวนผู้ที่เข้าถึงฐานข้อมูลหรือไม่

2.6 สถานีงาน

สถานีงานในข่ายคอมพิวเตอร์เฉพาะบริเวณส่วนใหญ่จะประกอบด้วยไมโครคอมพิวเตอร์ แรม จานแข็ง การ์ดที่ใช้เพื่อการเชื่อมโยงกับระบบเครือข่าย การเลือกใช้ไมโครคอมพิวเตอร์ในการทำสถานีงานก็มีเหตุอันเนื่องมาจากเป็นเครื่องที่มีความเชื่อถือได้ มีตัวแทนจำหน่าย นอกจากนี้ยังมีราคาต่ำ สำหรับจอภาพสีจะมีประโยชน์ในการทำงานทั้งนี้เนื่องจากโปรแกรมส่วนใหญ่จะใช้สีสันเพื่อสื่อความหมายกับผู้ใช้

สำหรับจานแข็งเป็นอุปกรณ์ที่มีความสำคัญเป็นอย่างยิ่งในสถานีงานของซีดีรอม ทั้งนี้เนื่องจากซอฟต์แวร์สำหรับการค้นหาข้อมูลของซีดีรอมโดยส่วนมากมักจะทำการโหลดจากตัวขับเฉพาะบริเวณ (local drive) มากกว่าที่จะทำการโหลดมาจากเครื่องบริการแฟ้ม สำหรับแรมนั้นจะต้องมีมากพอในการทำงานด้วย

2.7 เครื่องบริการแฟ้มและเครื่องบริการซีดีรอม

ในข่ายคอมพิวเตอร์เฉพาะบริเวณจะต้องมีการใช้ไมโครคอมพิวเตอร์เพื่อการควบคุมการทำงานในระบบเครือข่าย ซึ่งไมโครคอมพิวเตอร์ที่ใช้ในการควบคุมการทำงานต่าง ๆ เหล่านี้จะถูกเรียกว่าเครื่องบริการแฟ้ม เครื่องบริการแฟ้มเหล่านี้จะต้องมีไมโครโพรเซสเซอร์ที่มีความเร็วสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(เช่น 80386 หรือจะต้องมีความเร็วอย่างต่ำต้องสูงกว่าความเร็วของไมโครโพรเซสเซอร์ของสถานีงาน) นอกจากนี้ยังต้องมีจานแข็งอย่างต่ำ 30 ถึง 60 MB สำหรับหน่วยความจำจะต้องมีมากกว่าสถานีงาน สำหรับหน่วยความจำที่เหลือจะถูกใช้งานร่วมกับซีดีรอมเพื่อเพิ่มความเร็วในการทำงาน

เครื่องบริการเหล่านี้จะเป็นเครื่องแสดงถึงความเร็วของระบบเครือข่าย ทั้งนี้ขึ้นกับ

- ชนิดของไมโครโพรเซสเซอร์
- อัตราสัญญาณนาฬิกาของไมโครโพรเซสเซอร์
- สถาปัตยกรรมของอินพุต/เอาต์พุตบัสและบัสหน่วยความจำ (memory bus)
- อัตราเร็วในการเข้าถึงจานแข็ง

ดังจะเห็นได้จากไมโครคอมพิวเตอร์ขนาด 80386-25MHz ที่มีสถาปัตยกรรมบัสแบบไมโครชาแนล (MicroChanal bus architecture) จะสามารถเข้าถึงระบบเครือข่ายได้เร็วกว่าเครื่องพีซี (PC) หรือเอที (AT)

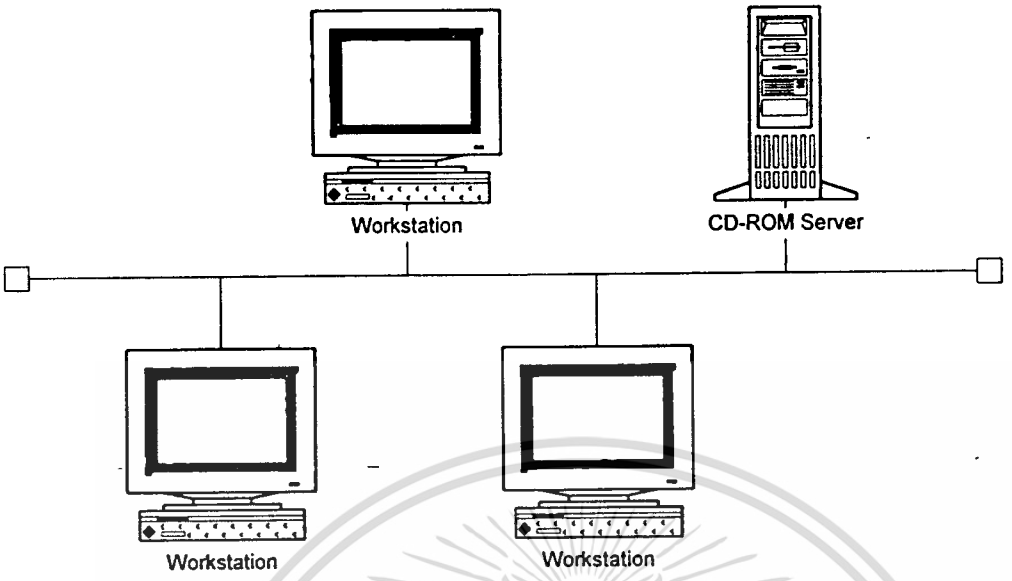
อย่างไรก็ตามในการทำงานของระบบเครือข่ายซีดีรอม อาจจะเป็นการทำงานที่อาศัยการทำงานของเครื่องบริการแฟ้ม แต่ในบางครั้งการทำงานอาจจะใช้เครื่องบริการซีดีรอมเพียงอย่างเดียว (ดังเช่นที่ได้กล่าวไว้ในหัวข้อ 2.2) โดยระบบเครือข่ายซีดีรอมสามารถแบ่งแยกย่อยออกเป็นรูปแบบต่าง ๆ ได้ 5 ชนิด โดยที่ชนิดที่ 1 2 3a และ 3b จะประกอบด้วยระบบเครือข่ายเป็นศูนย์กลางและมีเครื่องบริการควบคุมการทำงาน ส่วนชนิดที่ 4 จะการทำงานจะมีลักษณะที่กระจายออก (decentralized network) และไม่มีเครื่องบริการควบคุมการทำงาน

2.7.1 ชนิดที่ 1 : เครื่องบริการซีดีรอม

(Type 1 : CD-ROM Server Only)

เครื่องบริการซีดีรอมจะถูกติดตั้งเข้ากับระบบเครือข่าย ซึ่งอาจจะมีสถานีงานต่อเพียง 3 ถึง 5 สถานีงาน (รูปที่ 2.5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

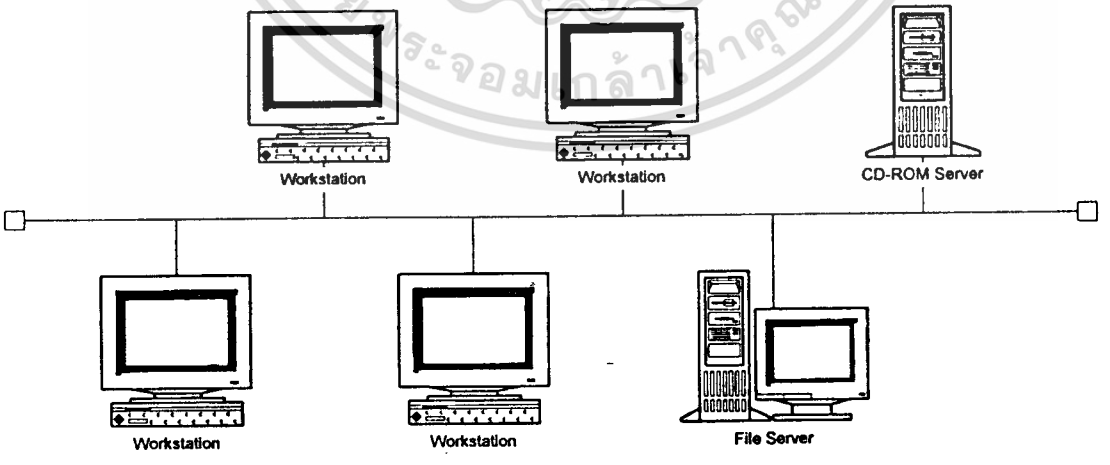


รูปที่ 2.5 ชนิดที่ 1 : เครื่องบริการซีดีรอม

2.7.2 ชนิดที่ 2 : เครื่องบริการซีดีรอมและเครื่องบริการแฟ้ม

(Type 2 : CD-ROM Server and File Server)

ระบบเครือข่ายชนิดนี้มีการพัฒนามาจากชนิดที่ 1 โดยมีการเพิ่มเครื่องบริการแฟ้มเข้าไปในระบบเพื่ออำนวยความสะดวกและช่วยจัดการงานด้านต่าง ๆ เช่นงานพิมพ์ โดยบางครั้งอาจจะนำเอาเครื่องบริการซีดีรอมมาต่อเพิ่มเติมจากเครื่องบริการแฟ้มเดิมที่มีอยู่ เครื่อง



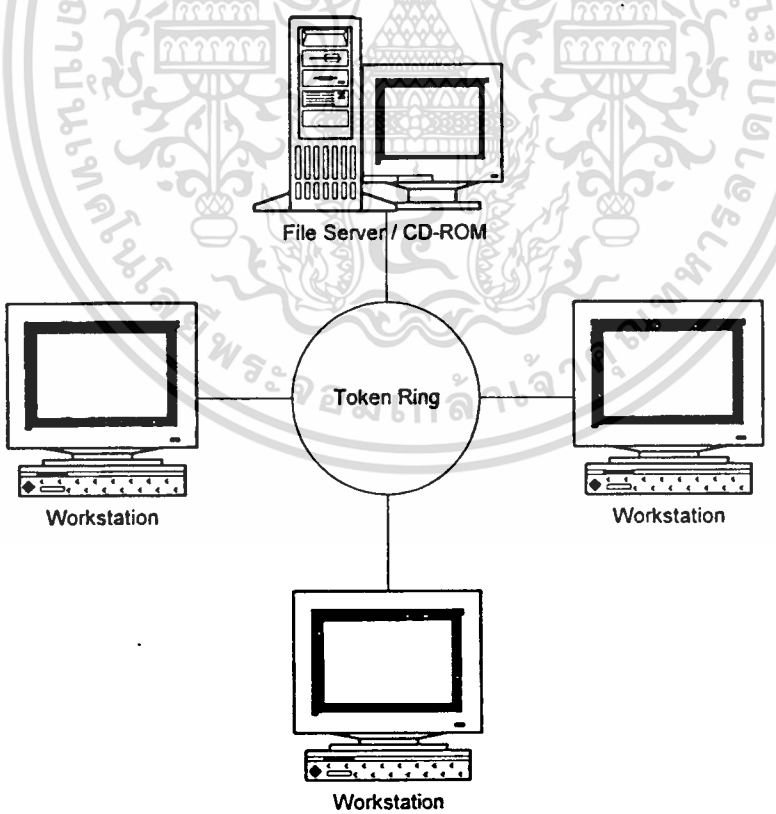
รูปที่ 2.6 ชนิดที่ 2 : เครื่องบริการซีดีรอมและเครื่องบริการแฟ้ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บริการแฟ้มจะทำหน้าที่ในการควบคุมการทำงานของระบบเครือข่าย ส่วนเครื่องบริการซีดีรอมจะทำหน้าที่ในการควบคุมการทำงานของซีดีรอม การติดตั้งการ์ดควบคุมการทำงานของระบบหลายตัวขับ (multidrive unit) ควรทำการติดตั้งในเครื่องไมโครคอมพิวเตอร์มากกว่าที่จะทำการติดตั้งบนเครื่องบริการแฟ้ม ระบบเครือข่ายชนิดที่ 2 จะสามารถรองรับการทำงานของสถานีงานได้ตั้งแต่ 4 ถึง 20 สถานีงาน

2.7.3 ชนิดที่ 3a : การรวมกันระหว่างเครื่องบริการแฟ้มและเครื่องบริการซีดีรอม (Type 3a : File Server / CD-ROM Server Combination)

เป็นการนำข่ายคอมพิวเตอร์เฉพาะบริเวณมาปรับปรุงโดยการนำตัวขับซีดีรอมชนิดหลายตัวขับ (multiple CD-ROM drives) มาต่อเข้ากับเครื่องบริการแฟ้มโดยใช้การ์ดควบคุมการทำงานของซีดีรอม สำหรับการติดตั้งโปรแกรมย่อยควบคุมอุปกรณ์จะทำการติดตั้งบนจานแข็ง เครื่องบริการแฟ้มจะทำหน้าที่ในการจัดการข่ายคอมพิวเตอร์เฉพาะบริเวณ รวมทั้งยังจัดการการทำงาน



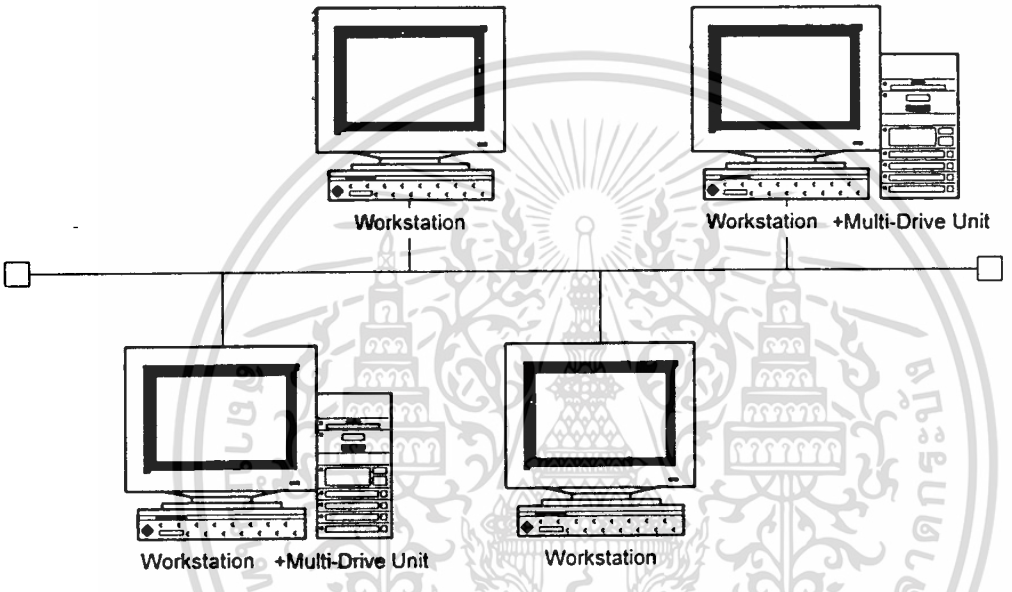
รูปที่ 2.7 ชนิดที่ 3a : การรวมกันระหว่างเครื่องบริการแฟ้มและเครื่องบริการซีดีรอม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.5 ชนิดที่ 4 : การทำงานแบบกระจายไม่มีเครื่องบริการ

(Type 4 : No Dedicated Server)

ระบบเครือข่ายแบบนี้จะทำโดยการเชื่อมสถานีงานซีดีรอม (CD-ROM workstation) เข้ากับสถานีงานต่าง ๆ สถานีงานซีดีรอมอาจจะมีมากกว่าหนึ่งสถานีงาน ดังนั้น ในการทำงานแบบนี้จะไม่มีการใช้เครื่องบริการแฟ้มหรือเครื่องบริการซีดีรอมแต่อย่างใด



รูปที่ 2.9 ชนิดที่ 4 : การทำงานแบบกระจายไม่มีเครื่องบริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ระบบเน็ตแวร์และหลักการในการเขียน โปรแกรมอรรถประโยชน์

ในบทนี้จะกล่าวถึงความเป็นมาของระบบเน็ตแวร์ (NetWare) และวิธีการเขียนโปรแกรมเพื่อทำการควบคุมระบบในแบบต่าง ๆ ข้อจำกัดในการเขียนโปรแกรม และรายละเอียดของโปรแกรมทดลอง

3.1 ประวัติในการพัฒนาระบบเน็ตแวร์

ในปลายปี 1970 บริษัทโนเวล ดาต้า ซิสเต็ม จำกัด (Novell Data Systems Inc.) ได้ทำการผลิตไมโครคอมพิวเตอร์ (microcomputer) z-80 เพื่อการใช้งานบน CP/M ต่อมาในราวต้นปี 1980 โนเวลก็ได้ทำการสร้างผลิตภัณฑ์ที่ใช้ในระบบการข่ายสื่อสาร (networking)

เครื่องบริการแฟ้ม (file server) รุ่นแรกของโนเวล ดาต้า ซิสเต็ม ถูกออกแบบและสร้างโดยใช้เครื่องประมวลผล (processor) Motorola 68000 ซึ่งมีลักษณะการทำงานรูปแบบเครือข่ายแบบดาว (star topology) และมีสายเคเบิ้ลแยกออกมาจากเครื่องบริการแฟ้มกลางต่อผ่าน RS-422 โดยจะมีความเร็วในการส่งข้อมูลสูงถึง 232 กิโลบิตต่อวินาที

สำหรับซอฟต์แวร์ที่ใช้กับเครื่องบริการ (server) นั้น ถูกสร้างโดยกลุ่มที่ปรึกษาซึ่งสำเร็จการศึกษามาจากมหาวิทยาลัยบริคัม ยัง (Brigham Young University) โดยมีการเรียกชื่อกลุ่มว่า ซูเปอร์เซ็ต (Superset) กลุ่มซูเปอร์เซ็ตนี้เป็นกลุ่มที่สร้างพื้นฐานของระบบปฏิบัติการ (operating system) เน็ตแวร์ขึ้นมา

ต่อมาไม่นานบริษัทโนเวล ดาต้าซิสเต็มได้ประสบภาวะล้มละลาย แต่ก็ได้มีผู้รวบรวมการก่อตั้งขึ้นมาใหม่ในนามของบริษัทโนเวล (Novell Inc) ซึ่งในขณะนั้นได้มีการแนะนำผลิตภัณฑ์คอมพิวเตอร์ส่วนบุคคลจากไอบีเอ็มสู่ตลาด สมาชิกของกลุ่มซูเปอร์เซ็ตได้สั่งซื้อคอมพิวเตอร์ส่วนบุคคลนี้มา และได้เริ่มเขียนเชลล์ (shells) เพื่อให้เครื่องสามารถทำการติดต่อกับเครื่องบริการแฟ้มที่สร้างจาก 68000 ได้

การพัฒนาระบบปฏิบัติการของเครื่องบริการแฟ้มได้มีการดำเนินงานต่อมาเรื่อย ๆ และมีการเปลี่ยนชื่อเป็น "แชร์เน็ต (ShareNet) " ซึ่งต่อมาชื่อนี้ได้กลายมาเป็นชื่อของระบบเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(network system) ที่สมบูรณ์ สำหรับการ์ดเชื่อมโยงเน็ตแวร์ (NetWare Interface cards) จะถูกติดตั้งไว้ในเครื่องเพื่ออำนวยความสะดวกในการเข้าถึงเครื่องบริการแฟ้ม

เมื่อไอบีเอ็มได้ทำการนำเครื่องคอมพิวเตอร์ส่วนบุคคลรุ่นเอ็กซ์ที (XT) สู่ตลาด แซร์เน็ต จึงถูกเปลี่ยนชื่อเป็น "เน็ตแวร์/เอส-เน็ต (NetWare/S-net)" โดย S จะมาจากคำว่า Star ซึ่งเป็นลักษณะการต่อสายของรูปแบบเครือข่ายแบบดาว

ในปี 1983 บริษัทโนเวลได้เริ่มวางโปรแกรมควบคุมระบบเครือข่ายโปรแกรมแรกออกสู่ตลาด โดยใช้ชื่อว่าเน็ตแวร์ 86 (NetWare 86) ซึ่งหมายถึงระบบปฏิบัติการเน็ตแวร์ที่ใช้ไมโครคอมพิวเตอร์ที่มีหน่วยประมวลผลกลาง 8086 เป็นเครื่องบริการ โดยเน็ตแวร์จะทำหน้าที่เป็นระบบปฏิบัติการของเครื่องบริการแทนดอส เพื่อเพิ่มความเร็วในการทำงานของเครือข่ายให้สูงขึ้น ส่วนปัญหาในการล็อคบันทึก (record) เน็ตแวร์ได้ออกโครงสร้างของจานแข็ง (harddisk) ขึ้นมาใหม่ ซึ่งต่างไปจากดอสเพื่อแก้ปัญหาเมื่อมีผู้ใช้หลาย ๆ คน ใช้แฟ้ม (file) ข้อมูลร่วมกัน ซึ่งต่อมาดอสเองก็ต้องเปลี่ยนแปลงโครงสร้างของจานแข็งใหม่เช่นกันเพื่อปรับปรุงขึ้นเป็นดอสรุ่น 3.0

ต่อมาเมื่อมีการนำหน่วยประมวลผลกลาง 80286 มาใช้งานกับไมโครคอมพิวเตอร์ โนเวลได้วางตลาดโปรแกรมแอดวานซ์เน็ตแวร์ 286 (Advanced NetWare 286) ในปี 1985 โดยใช้ร่วมกับหน่วยประมวลผลกลาง 80286 (ไมโครคอมพิวเตอร์ระดับ AT) โดยถูกออกแบบมาให้ใช้งานในโหมดคุ้มกัน (protect mode) ของ 80286 ทำให้หลุดพ้นจากข้อจำกัดของดอสได้อย่างสิ้นเชิง และทำให้สามารถใช้ประสิทธิภาพของหน่วยประมวลผลกลาง มีความเร็วในการทำงานที่แตกต่างจากระบบปฏิบัติการอื่นอย่างเห็นได้ชัด ทำให้แอดวานซ์เน็ตแวร์ 286 ได้รับความนิยมอย่างมาก สำหรับระบบข่ายคอมพิวเตอร์เฉพาะบริเวณ (LAN) ขนาดเล็ก โนเวลก็ได้ออก อีแอลเอสเน็ตแวร์ (Entry Level System NetWare) เป็นรุ่นสำหรับเครือข่ายขนาดเล็ก ที่มีผู้ใช้ 4-8 คน

เพื่อเน้นความถูกต้องในการทำงาน โนเวลได้พัฒนาโปรแกรมในตระกูลเน็ตแวร์ 286 ขึ้นอีกเรียกว่าเอสเอฟทีเน็ตแวร์ 286 (System Fault Tolerant NetWare) เป็นระบบที่มีลักษณะในการทำงานแบบความทนความผิดพร่อง (fault tolerant) ซึ่งจะทำงานโดยสำรองเนื้อที่ส่วนไดเรกทอรี (directory) ตารางการเก็บแฟ้ม (FAT - file allocation table) หรือสำรองจานแข็งไว้สองชุดตลอดเวลาที่ทำงาน ถ้าหากเกิดปัญหาขึ้นจะสามารถใช้ข้อมูลชุดสำรองทำงานแทนได้ทันที

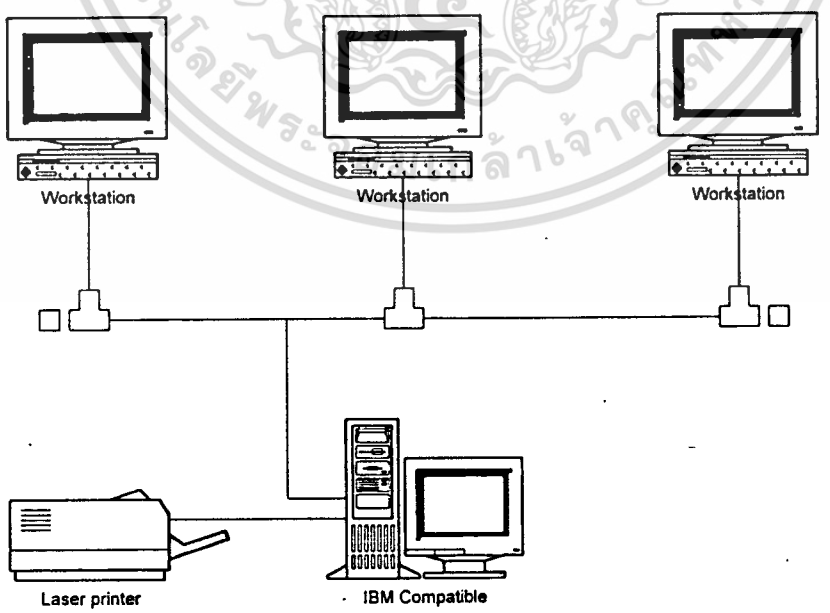
นอกจากเอสเอฟทีเน็ตแวร์ 286 แล้วโปรแกรมชุด เน็ตแวร์ 286 ยังมี แอดวานซ์เน็ตแวร์ 286 ในแบบที่ทำงานกับเครือข่ายแบบจุดต่อจุด (Peer-to-Peer) อีกด้วย ตามปกติการทำงานของเน็ตแวร์ ไมโครคอมพิวเตอร์ที่ทำหน้าที่เป็นเครื่องบริการ จะถูกกำหนดให้ทำหน้าที่ให้บริการกับคอมพิวเตอร์เครื่องอื่นเพียงอย่างเดียว ผู้ใช้ไม่สามารถใช้งานในลักษณะไมโครคอมพิวเตอร์ทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้อีก ซึ่งจะถูกเรียกว่าเครื่องบริการเฉพาะ (Dedicated Files Server) ส่วนเน็ตแวร์ที่ทำงานกับเครือข่ายแบบจุดต่อจุดจะมีชื่อว่าเน็ตแวร์ไลต์ (NetWare Lite) ซึ่งวางตลาดในปี 1991 โดยเครื่องที่ทำหน้าที่เป็นเครื่องบริการจะยอมให้ผู้ใช้ทำงานในลักษณะของสถานีงาน (workstation) ไปได้พร้อมกัน ซึ่งช่วยให้การใช้งานประหยัดขึ้น แต่ความเร็วในการทำงานจะต่ำกว่าในแบบเครื่องบริการเฉพาะ และเครื่องที่นำมาทำเป็นเครื่องบริการจะต้องมีหน่วยความจำและความเร็วสูงพอที่จะทำหน้าที่เป็นทั้งเครื่องบริการและสถานีงาน (workstation) ไปได้พร้อม ๆ กัน เน็ตแวร์ไลต์นี้ได้ถูกพัฒนาเป็นเน็ตแวร์ส่วนบุคคล (Personal NetWare) ในปัจจุบัน

ในเวลาต่อมาโนเวลได้รวมเอาเน็ตแวร์ในระดับของหน่วยประมวลผลกลาง 286 ทั้งหมดเข้าด้วยกันเป็นตัวเดียวเรียกว่าเน็ตแวร์ 2.2 (NetWare 2.2) ซึ่งเป็นรุ่นที่ยังใช้กันมาจนถึงปัจจุบัน แต่ก็ถูกแทนด้วยเน็ตแวร์รุ่นถัดไปคือเน็ตแวร์ 386 (NetWare 386) หรือ 3.xx ไปเรื่อย ๆ

หลังจากที่เน็ตแวร์ 286 ออกสู่ตลาดหลายปี โนเวลจึงได้พัฒนาโปรแกรมเน็ตแวร์ให้ทำงานได้โดยใช้ประสิทธิภาพของหน่วยประมวลผลกลาง 80386 ซึ่งทำงานแบบ 32 บิต โดยให้ชื่อว่าเน็ตแวร์ 386 เริ่มวางตลาดในปี 1990 โดยหวังว่าเน็ตแวร์ 386 จะเป็นมาตรฐานของระบบเครือข่ายในอนาคต เน็ตแวร์ในรุ่นนี้มีความเร็วในการทำงานที่เพิ่มขึ้น และสามารถรองรับปริมาณงานได้เพิ่มขึ้น แต่เน็ตแวร์ 386 หรือที่เรียกกันอีกชื่อหนึ่งว่าเน็ตแวร์ 3.xx ยังขาดความสามารถในการจัดการทรัพยากรของเครือข่ายขนาดใหญ่ที่มีเครื่องบริการเชื่อมต่อกันหลาย ๆ ตัว



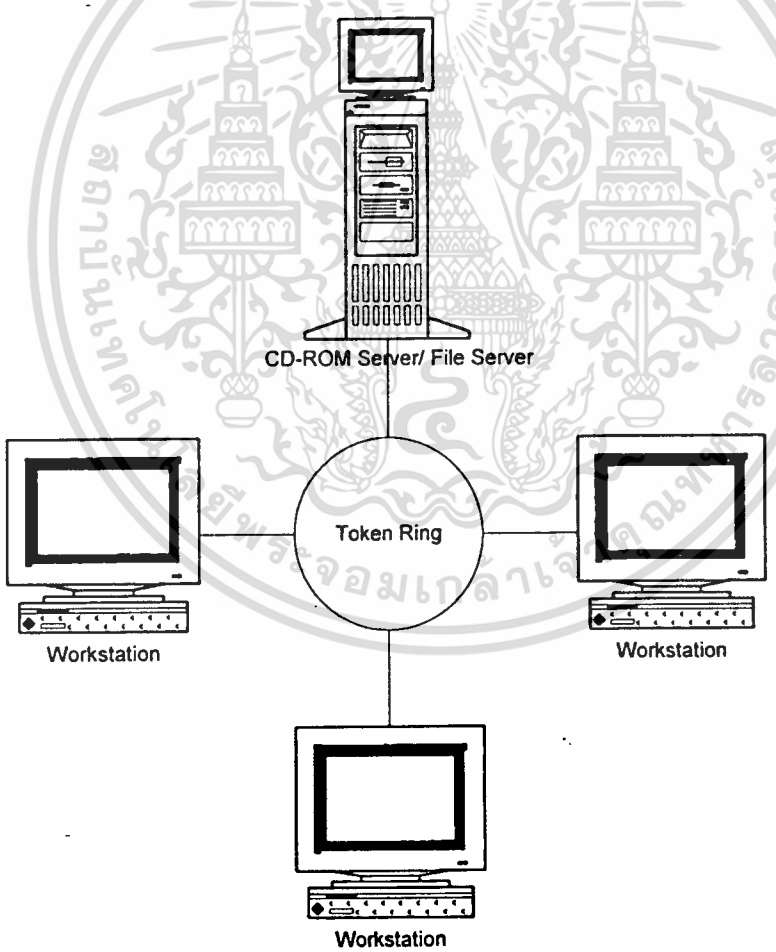
รูปที่ 3.1 เครื่องบริการกับสถานีงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่าง ๆ ในระบบเครือข่ายเช่นงานพิมพ์และการส่งข่าวสาร รวมทั้งยังมีหน้าที่ในการจัดการตัวขับซีดีรอมอีกด้วย ระบบเครือข่ายชนิด 3a จะสามารถรองรับการทำงานได้ถึง 10 สถานีงานต่อเครื่องบริการ

2.7.4 ชนิดที่ 3b : การรวมกันระหว่างเครื่องบริการซีดีรอมและเครื่องบริการแฟ้ม (Type 3b : CD-ROM Server / File Server Combination)

ระบบเครือข่ายแบบนี้มีการพัฒนามาจากชนิด 1 โดยการเพิ่มจานแข็ง จอภาพแป้นพิมพ์เข้ากับซีดี เน็ต เทาเวอร์ (CD Net Tower) เครื่องบริการชนิดนี้จะสามารถจัดการระบบเครือข่ายและซีดีรอมได้อย่างมีประสิทธิภาพ



รูปที่ 2.8 ชนิดที่ 3b : การรวมกันระหว่างเครื่องบริการซีดีรอมและเครื่องบริการแฟ้ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในปี 1993 โนเวลได้วางตลาดเน็ตแวร์ 4.xx (NetWare 4.xx) ซึ่งใช้เครื่องบริการระดับ 80386 แต่สามารถจัดการบริหารเครือข่ายขนาดใหญ่ได้ ซึ่งช่วยให้ผู้ใช้เน็ตแวร์ทำการเชื่อมต่อหลาย ๆ ระบบได้โดยสะดวกและมีประสิทธิภาพมากขึ้น

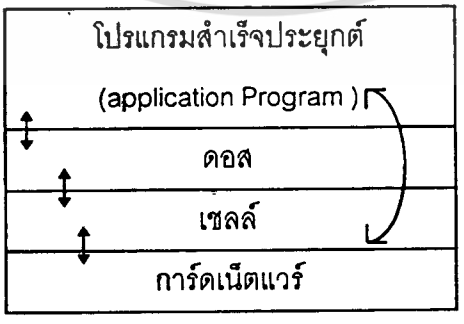
3.2 การเขียนโปรแกรมควบคุมการทำงานของระบบเน็ตแวร์

การเขียนโปรแกรมควบคุมการทำงานของระบบเน็ตแวร์มีจุดประสงค์เพื่อการประยุกต์การใช้งานของเน็ตแวร์ การสร้างโปรแกรมอรรถประโยชน์ (utilities program) ต่าง ๆ โดยสามารถทำได้สองวิธีคือ

3.2.1 การเขียนโปรแกรมโดยการขัดจังหวะ (interrupt)

การเขียนโปรแกรมในลักษณะนี้จะทำโดยผ่านเชลล์ของสถานีงาน (workstation shell) เชลล์ของสถานีงานคือโปรแกรมที่ถูกโหลดเข้าไปในหน่วยความจำก่อนที่ผู้ใช้จะลงบันทึกเข้า (login) ไปในเครือข่าย เชลล์ของเน็ตแวร์ 2.1 หรือเน็ตแวร์ในรุ่นถัดมาจะประกอบด้วยโปรแกรม IPX และ NET3 ซึ่ง IPX จะถูกใช้เพื่อการติดต่อสื่อสารกับเครื่องบริการแฟ้มและสถานีงานต่าง ๆ ส่วน NET3 จะใช้เป็นตัวเชื่อโยงระหว่างเครือข่ายกับดอส

สำหรับผู้ที่ใช้เน็ตแวร์ในรุ่นก่อนหน้า 2.1 จะพบว่ามีการรวม IPX และ NET3 เข้าไว้ในแฟ้มเดียวกัน ซึ่งถูกเรียกว่า ANET3.COM สำหรับในเน็ตแวร์ 2.1x IPX จะมีลักษณะเป็นเอกเทศ (standalone) ซึ่งจะมีการเปลี่ยนแปลงไปตามชนิดของการ์ดแลน



รูปที่ 3.2 แสดงความสัมพันธ์ระหว่างเชลล์ของเน็ตแวร์กับดอส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเซลล์ของเน็ตแวร์ถูกโหลด เซลล์จะเลือกใช้การขัดจังหวะของดอสในส่วนที่สงวนไว้ ซึ่งจะมีการขัดจังหวะที่ถูกใช้งานโดย IPX.COM และการขัดจังหวะที่ถูกใช้โดย NETX.COM

3.2.1.1 การขัดจังหวะโดย IPX.COM

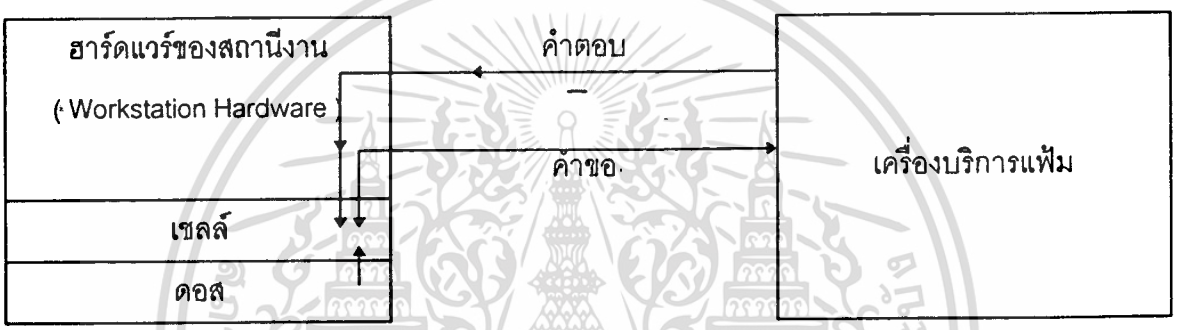
08h	IPX จะรับข้อมูลเกี่ยวกับเวลาจากการขัดจังหวะนี้
2Fh	ดอสมัลติเพลกซ์ - เมื่อมีการเรียกใช้การขัดจังหวะนี้ เซลล์จะทำการตรวจสอบเพื่อดูว่าใช่การขัดจังหวะหมายเลข 7Ah หรือไม่ ถ้าใช่ตำแหน่งเซกเมนต์:ออฟเซท (segment:offset) ของการขัดจังหวะหมายเลข 7Ah จะถูกส่งกลับ ซึ่งจะถูกใช้งานในการเขียนโปรแกรมภายหลัง
64h	เป็นการขัดจังหวะภายในของ IPX (จะชี้ไปยังตำแหน่งเดียวกับการขัดจังหวะหมายเลข 7Ah)
7Ah	เป็นการขัดจังหวะของ IPX (ใช้สำหรับอรรถประโยชน์โดยทั่วไป)
F4h & F6h	เป็นการขัดจังหวะภายในของ IPX
Fxh	ใช้ในงานอื่น ๆ

3.2.1.2 การขัดจังหวะโดย NETX.COM (Net2 Net3 Net4)

10h	ไบออสของวีดีโอ (ใช้โดยกระบวนการส่งข่าวสาร)
17h	ไบออสของเครื่องพิมพ์ (สำหรับเปลี่ยนทิศทางของเครื่องพิมพ์แบบขนาน)
1Bh	ปุ่มคอนโทรลเบรคหรือคอนโทรลซี (สงวนการใช้งานของปุ่มทั้งสองในการทำงานของเซลล์)
20h	เป็นการขัดจังหวะเมื่อมีการสิ้นสุดโปรแกรมเดิม
21h	เป็นการขัดจังหวะของดอส (โปรแกรมโดยส่วนใหญ่มักใช้การขัดจังหวะหมายเลขนี้)
24h	แสดงความผิดพลาดที่เกิดขึ้นในเครือข่าย
27h	การขัดจังหวะของโปรแกรมที่ฝังตัวอยู่ในหน่วยความจำ
F5h	ใช้ในการทำงานภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เชลล์จะทำการขัดจังหวะระหว่างสถานีงานและเครื่องบริการแฟ้ม เช่นถ้าผู้ใช้ต้องการเปิดแฟ้มภายในเครือข่าย เชลล์จะทำการแทรป (trap) คำขอในการเปิดไฟล์ และทำการส่งกลุ่มข้อมูลข่าวสาร (packet) ไปยังเครื่องบริการแฟ้ม หลังจากนั้นเครื่องบริการจะทำการส่งกลุ่มข้อมูลข่าวสารกลับไปยังเชลล์เพื่อแจ้งว่าคำขอเหล่านั้นประสบความสำเร็จหรือไม่ เชลล์จึงจะกลับจากการขัดจังหวะหมายเลข 21h ไปสู่โปรแกรมหลักดังรูป



รูป 3.3 คำขอจะถูกส่งและตอบสนองผ่านระบบเครือข่าย

นอกจากนี้ภายในเชลล์ยังประกอบด้วยตารางต่าง ๆ ที่ทำการเก็บข้อมูลที่เกี่ยวข้องกับตัวขับ เครื่องบริการที่เกี่ยวข้อง รวมทั้งรายละเอียดเกี่ยวกับสถานีงานและสภาพแวดล้อม (environment) ต่าง ๆ

สำหรับการเขียนโปรแกรมเหล่านี้สามารถใช้ภาษาได้หลายรูปแบบ ทั้งภาษาแอสเซมบลี ภาษาซี ภาษาปาสคาล ภาษาฟอแทรน คลิปเปอร์ และภาษาอื่น ๆ อีกมากมาย

3.2.2 การเขียนโปรแกรมโดยใช้โปรแกรมสำเร็จรูปจากโนเวล

โนเวลได้ทำการออกโปรแกรมอรรถประโยชน์เพื่อช่วยใช้ในการประสานการทำงานระหว่างผู้ใช้กับระบบเน็ตแวร์ ทำให้ผู้ใช้สามารถเขียนโปรแกรมได้ง่ายขึ้น ไม่ต้องยุ่งเกี่ยวกับการขัดจังหวะโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมของโนเวลจะประกอบด้วยไลบรารีที่ทำการประสานงานกับ Watcom C Borland Turbo C++ Turbo C Microsoft C และ Lattice C ซึ่งผู้ใช้สามารถเลือกใช้โปรแกรมเหล่านี้ได้ตามต้องการ

อย่างไรก็ตามการเขียนโปรแกรมทั้งสองแบบนี้ยังมีข้อจำกัดว่าผู้เขียนมีสิทธิในการเขียนโปรแกรมได้มากแค่ไหน ซึ่งบางคำสั่งผู้ที่จะใช้งานได้ต้องมีระดับเป็นผู้ดูแล (Supervisor) หรือบางคำสั่งผู้ที่จะสามารถใช้งานได้ต้องสามารถใช้คำสั่ง FCONSOLE ได้เป็นต้น

3.3 ระดับผู้ใช้ต่าง ๆ ในระบบเน็ตแวร์ที่มีผลกระทบต่อการเขียนโปรแกรม

เน็ตแวร์มีการแบ่งประเภทผู้ใช้ระบบข่ายคอมพิวเตอร์เฉพาะบริเวณออกเป็น 5 ประเภทใหญ่ ๆ ด้วยกันคือ

3.3.1 ผู้ใช้ทั่วไป

ผู้ใช้ประเภทนี้จะมีสิทธิและขอบเขตการทำงานตามที่กำหนดไว้ล่วงหน้า และผู้ใช้ประเภทนี้จะไม่มีสิทธิพิเศษอื่น ๆ

3.3.2 ผู้ควบคุมเครื่อง

ผู้ใช้ประเภทนี้จะทำหน้าที่ควบคุมการทำงานของอุปกรณ์ต่าง ๆ ภายในเน็ตแวร์ ซึ่งผู้ควบคุมเครื่องของเน็ตแวร์ประกอบด้วย

3.3.2.1 ผู้ควบคุมเครื่องบริการแฟ้ม (File Server Console Operator)

จะทำหน้าที่เปิด/ปิดเครื่องบริการ โดยใช้คำสั่ง FCONSOLE

3.3.2.2 ผู้ควบคุมลำดับการพิมพ์ (Print Queue Operator)

จะทำหน้าที่ควบคุมและสั่งงานลำดับการพิมพ์ซึ่งจะกำหนดได้ในคำสั่ง PCONSOLE

3.3.2.3 ผู้ควบคุมเครื่องบริการการพิมพ์ (Print Server Operator)

จะทำหน้าที่ควบคุมเครื่องบริการการพิมพ์โดยใช้คำสั่ง PCONSOLE หรือ PSC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 เมื่อมีการใช้คำสั่ง map ดูหมวดใน MCLAB3 จะได้ผลดังนี้

```
Drive A:  maps to a local disk.
Drive B:  maps to a local disk.
Drive C:  maps to a local disk.
Drive D:  maps to a local disk.
Drive E:  maps to a local disk.
Drive F: = MCLAB3\SYS:  \
Drive G: = MCLAB3\SYS:HOME\CLIB1  \
-----
SEARCH1:  = M:. [MCLAB3\SYS:  \PUBLIC]
SEARCH2:  = L:. [MCLAB3\SYS:  \]
SEARCH3:  = Z:. [MCLAB3\SYS:  \PUBLIC]
SEARCH4:  = C:\DOS
SEARCH5:  = Y:. [MCLAB3\SYS:  \]
SEARCH6:  = X:. [MCLAB3\SYS:  \WINDOWS]
SEARCH7:  = W:. [MCLAB3\SYS:  \WINDOWS\MSOFFICE\WORKDIR]
SEARCH8:  = U:. [MCLAB3\SYS:  \CAD\ACAD12]
SEARCH9:  = T:. [MCLAB3\SYS:  \UTIL\NC]
SEARCH10:  = S:. [MCLAB3\SYS:  \UTIL\NU]
SEARCH11:  = R:. [MCLAB3\SYS:  \PKG\FOXPRO25]
SEARCH12:  = Q:. [MCLAB3\SYS:  \WINDOWS\AMIPRO\STYLES\TSM]
SEARCH13:  = P:. [MCLAB3\SYS:  \WINDOWS\WINPROJ]
SEARCH14:  = O:. [MCLAB3\SYS:  \UTIL\TOOLS]
SEARCH15:  = N:. [MCLAB3\SYS:  \WINDOWS\LD\LDLIB]
SEARCH16:  = V:. [MCLAB3\SYS:  \WINDOWS\MSAPPS\PROOF]
```

แต่เมื่อมีการใช้คำสั่ง VOLINFO ดู จะพบว่ามีหมวด SYS และ FULCRUM เมื่อตรวจดูจากคำสั่ง map จะพบว่าหมวด FULCRUM ยังไม่ได้ทำการแมป ดังนั้นถ้าหากเป็นการทำงานในรูปแบบปกติ จะต้องใช้คคำสั่ง map next เพื่อทำการแมปหมวด FULCRUM เพิ่มเติม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 เมื่อมีการใช้โปรแกรมนี้ในการทำงาน

ไม่จำเป็นที่จะต้องใช้คำสั่ง VOLINFO เพื่อดูชื่อหมวด โดยโปรแกรมนี้จะมีความสามารถในการค้นหาชื่อหมวดที่ยังไม่ได้ทำการแมป แล้วแมปตัวขับถัดไปให้ ซึ่งมีผลคล้ายคำสั่ง map next นอกจากนี้โปแกรมนี้ยังสามารถแสดงชื่อของเครื่องบริการแฟ้มและหมวดทั้งหมดที่มีอยู่ได้อีกด้วย แต่โปรแกรมนี้มีขนาดค่อนข้างใหญ่ ทั้งนี้เนื่องจากผู้เขียนมีระดับการใช้งานระดับผู้ใช้ทั่วไป ทำให้ไม่สามารถใช้งานในบางคำสั่งได้โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ทฤษฎีในการเบี่ยงเบนทิศทางทิศทางข้อมูล

การเบี่ยงเบนทิศทาง(Redirection) คือ การเปลี่ยนแฟ้มข้อมูล (file) ตัวเดิมที่เคยเป็นทางเข้าหรือทางออกของกระแสข้อมูล (stream) หนึ่งไปเป็นแฟ้มข้อมูลใหม่ ซึ่งปรกติโปรแกรมที่ทำงานอยู่ภายใต้ระบบปฏิบัติการจะมีเส้นทางเข้าออกของข้อมูลหลักอยู่สองเส้นทาง คือ เส้นทางจากแป้นพิมพ์ และเส้นทางไปยังจอภาพ ซึ่งเส้นทางเหล่านี้จะตรงกับสิ่งที่เราเรียกว่า กระแสข้อมูล (stream) ส่วนแป้นพิมพ์และจอภาพนั่นก็คือแฟ้มข้อมูลนั่นเอง กระแสข้อมูลเหล่านี้สามารถปรับเปลี่ยนทิศทางให้รับข้อมูลจากที่เดิมคือแป้นพิมพ์ ไปยังแหล่งรับข้อมูลจากแหล่งอื่นได้ ในทำนองเดียวกันนั้น การส่งข้อมูลไปยังจอภาพก็อาจปรับเปลี่ยนทิศทางให้ส่งออกไปยังแหล่งอื่นได้เช่นเดียวกัน

สำหรับในระบบปฏิบัติการดอส จะใช้เครื่องหมาย > บอกการ เบี่ยงเบนทิศทางส่งข้อมูลออกและเครื่องหมาย < บอกการเบี่ยงเบน ทิศทางส่งข้อมูลเข้า โดยจะเขียนเครื่องหมายดังกล่าวร่วมกับชื่อแฟ้มข้อมูลใหม่ที่ต้องการเปลี่ยนทิศทางไป ในรูปของพารามิเตอร์ที่ตามหลังชื่อโปรแกรม ซึ่งจะมีการทำงานที่ดอสพรอมต์ สำหรับในวิทยานิพนธ์ฉบับนี้จะเป็นการศึกษาถึงการเปลี่ยนทิศทางจากแป้นพิมพ์ไปเป็นแฟ้มข้อมูลเข้า และการเปลี่ยนทิศทางจากจอภาพไปเป็นแฟ้มข้อมูลออก ซึ่งจะต้องศึกษาถึงรายละเอียดการทำงานของแป้นพิมพ์ และ จอภาพอย่างละเอียด ดังนี้

4.1 หลักการทำงานของจอภาพ

4.1.1 วิธีการทำงานของจอภาพ

จอภาพมีหน้าที่แสดงข้อมูลต่างๆ และข้อมูลนั้นจะต้องถูกบันทึกไว้ที่ใดที่หนึ่ง เครื่องพีซีจะเก็บข้อมูลเหล่านั้นไว้ภายในเครื่องคอมพิวเตอร์แทนการเก็บข้อมูลเหล่านั้นไว้ที่จอภาพ ทั้งนี้เนื่องจากต้องการให้เกิดความยืดหยุ่นและมีความเร็วในการทำงานมากที่สุดนั่นเอง การทำเช่นนี้ได้จะต้องมีการกันเนื้อที่ของหน่วยความจำส่วนหนึ่ง เพื่อใช้สำหรับเก็บข้อมูลที่ต้องการแสดงผล โดยจะเรียกหน่วยความจำส่วนนี้ว่า "หน่วยความจำแสดงผล" (display memory)

หน่วยความจำแสดงผลมีแตกต่างจากหน่วยความจำส่วนอื่น ๆ ตรงที่หน่วยความจำส่วนนี้จะใช้งานได้จาก 2 ทาง ทางแรกคือ โปรเซสเซอร์และโปรแกรมต่าง ๆ เรียกใช้งาน การทำเช่นนี้จะมีลักษณะเช่นเดียวกับการใช้งานในหน่วยความจำส่วนอื่น ๆ อีกทางหนึ่งคือการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจอภาพ ใช้สำหรับทำหน้าที่ในการตรวจสอบข้อมูลที่ต้องการไปแสดงผล การเข้าถึงหน่วยความจำสำหรับแสดงผลทั้งสองแบบนั้นจะทำได้โดยการเรียกใช้งานจากโปรเซสเซอร์ ซึ่งจะใช้เพื่อการเก็บข้อมูลที่ต้องการแสดงผล จอภาพจะทำการตรวจสอบหน่วยความจำแสดงผลเพื่อดูว่ามีข้อมูลส่วนใดที่ต้องการนำไปแสดงผลบ้าง โดยทั้งโปรเซสเซอร์และจอภาพล้วนแล้วแต่ใช้หน่วยความจำแสดงผลผ่านเกตของตัวเอง ไม่มีความเกี่ยวข้องใด ๆ ทั้งสิ้น ลักษณะของหน่วยความจำแบบนี้มีชื่อเรียกว่า หน่วยความจำแบบ 2 พอร์ต (dual-port memory)

เมื่อโปรแกรมที่กำลังทำงานอยู่ต้องการแสดงผลออกหน้าจอ จะทำได้โดยการใส่ข้อมูลเหล่านั้น ๆ ลงในหน่วยความจำแสดงผล จอภาพจะอ่านข้อมูลจากหน่วยความจำแสดงผลเพื่อตรวจสอบว่ามีข้อมูลอะไรที่จะต้องทำการแสดงผล การแสดงผลจะทำโดยการสร้างภาพของข้อมูลนั้น ๆ ออกสู่หน้าจอ การอ่านและการสร้างภาพดังกล่าวจะมีการซ้ำกันหลาย ๆ ครั้งราว 70 ครั้งต่อวินาที ส่วนที่ทำงานดังกล่าวจะเป็นอุปกรณ์ทางอิเล็กทรอนิกส์ที่อยู่ในจอภาพ จากการที่มีการสร้างภาพอยู่บนจอภาพบ่อยครั้งนี่เอง ทำให้ข้อมูลที่เก็บบนหน่วยความจำแสดงผลจะถูกแสดงออกบนจอโดยทันที ดังนั้นถ้าโปรแกรมใดมีความต้องการที่จะแสดงผล จะสามารถทำได้โดยการใส่ข้อมูลเหล่านั้นลงในหน่วยความจำแสดงผลตามที่ต้องการ

4.1.2 หลักการพื้นฐานของโหมดแสดงผล

ในการแบ่งโหมดพื้นฐานของโหมดแสดงผลนั้น จะแบ่งได้ในหลายลักษณะ การแบ่งโหมดการแสดงผลในแบบแรกคือการแบ่งออกเป็นโหมดข้อความและโหมดกราฟิก ซึ่งในส่วนของโหมดข้อความเองก็แบ่งได้เป็นหลายประเภท

จอภาพจะสามารถแสดงผลได้เพียงชุดตัวอักษรพื้นฐานของเครื่องตระกูลพีซีเท่านั้น ตัวอักษรทั้ง 256 ตัวดังกล่าวเป็นอักขระที่มีใช้กันมากในการเขียนหนังสือ รวมทั้งมีอักขระที่ใช้ในการวาดรูปกรอบต่าง ๆ ซึ่งทำให้สามารถสร้างภาพอื่น ๆ ขึ้นมาได้โดยอาศัยตัวอักษรชุดดังกล่าวประกอบกัน จุดอ่อนของโหมดข้อความคือ สิ่งที่จะนำมาแสดงบนจอภาพจะจำกัดอยู่เพียง 256 ตัวอักษรในชุดตัวอักษรพื้นฐานเท่านั้น ไม่มีความสามารถพิเศษอื่นใด และการแสดงผลจะสามารถแสดงได้ทั้งหมด 25 แถว แต่ละแถวจะประกอบด้วยคอลัมน์ 80 คอลัมน์

วิธีการแสดงผลอีกโหมดคือการแสดงผลในโหมดกราฟิก จอภาพที่แสดงในโหมดกราฟิกจะเปรียบเทียบได้กับตารางที่บรรจุจุดเล็ก ๆ ซึ่งเรียกว่า จุดภาพ (pixel ย่อมาจาก picture elements) โดยภาพต่าง ๆ ที่ปรากฏบนจอจะเกิดจากการนำจุดเหล่านี้มาประกอบเข้าด้วยกัน สำหรับโหมดกราฟิกนั้นจะมีอยู่หลายแบบ ทั้งนี้ขึ้นอยู่กับความหนาแน่นของจุดบนจอภาพ ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรียกว่าความละเอียด (resolution) ในโหมดที่มีความละเอียดสูง (high resolution) จะสามารถแสดงจุดได้มากกว่า 200 จุดในแนวตั้งและมากกว่า 640 จุดในแนวนอน หรือมีความละเอียดของจอภาพมากกว่า 640 x 200 นั่นเอง ในการวาดรูปต่าง ๆ บนจอภาพจะอาศัยจุดเหล่านี้มาประกอบกันเป็นรูปที่ต้องการ ซึ่งรวมถึงตัวอักษรต่าง ๆ สำหรับในเครื่องพีซีจะมีโปรแกรมพิเศษที่ถูกติดตั้งมาอยู่ในรอมไบออส เพื่อใช้สำหรับทำหน้าที่ในการวาดภาพตัวอักษรดังกล่าวแบบจุดต่อจุด โปรแกรมต่าง ๆ ที่ทำงานอยู่ในโหมดกราฟิกสามารถเรียกใช้งานโปรแกรมพิเศษนี้ได้ ทำให้สามารถทำงานอย่างอื่นได้โดยไม่ต้องเสียเวลากับวาดภาพอีก

นอกจากนี้ โหมดแสดงผลยังสามารถแบ่งออกได้ตามความสามารถในการให้สี ซึ่งบางโหมดก็สามารถให้สีได้ 2 สี คือ สีขาวกับดำ บางโหมดจะสามารถให้สีได้มากกว่านี้ เช่น 4 สี หรืออาจจะมากกว่า 256 สี ท้ายสุดคือ โหมดโมโนโครม (monochrome mode) คือโหมดที่สามารถแสดงผลได้สีเดียว แต่มี "ลักษณะเฉพาะในการแสดงผล (display attribute)" อันประกอบด้วยตัวอักษรปรกติ ตัวเข้ม ตัวกระพริบ ตัวกลับขาวดำ และตัวขีดเส้นใต้

จากการแบ่งโหมดการแสดงผลใน 2 มุมมองดังกล่าว ทำให้เกิดการรวมโหมดต่าง ๆ เข้าด้วยกัน ซึ่งแบ่งออกเป็นประเภทใหญ่ ๆ ได้ 4 ประเภทคือ โหมดข้อความหรือโหมดกราฟิก และมีสีหรือไม่มีสี นอกจากนี้ยังอาจนำเอาคุณสมบัติอื่น ๆ มาร่วมพิจาราณาด้วยเช่นความละเอียดของจอภาพ จำนวนสีที่ใช้ และ อื่น ๆ

ในวิทยานิพนธ์ฉบับนี้จะกล่าวถึงการทำงานในโหมดข้อความเท่านั้น ซึ่งจะกล่าวถึงรายละเอียดต่าง ๆ ในหัวข้อถัดไป

4.1.3 โครงสร้างของโหมดข้อความ

4.1.3.1 จอภาพ

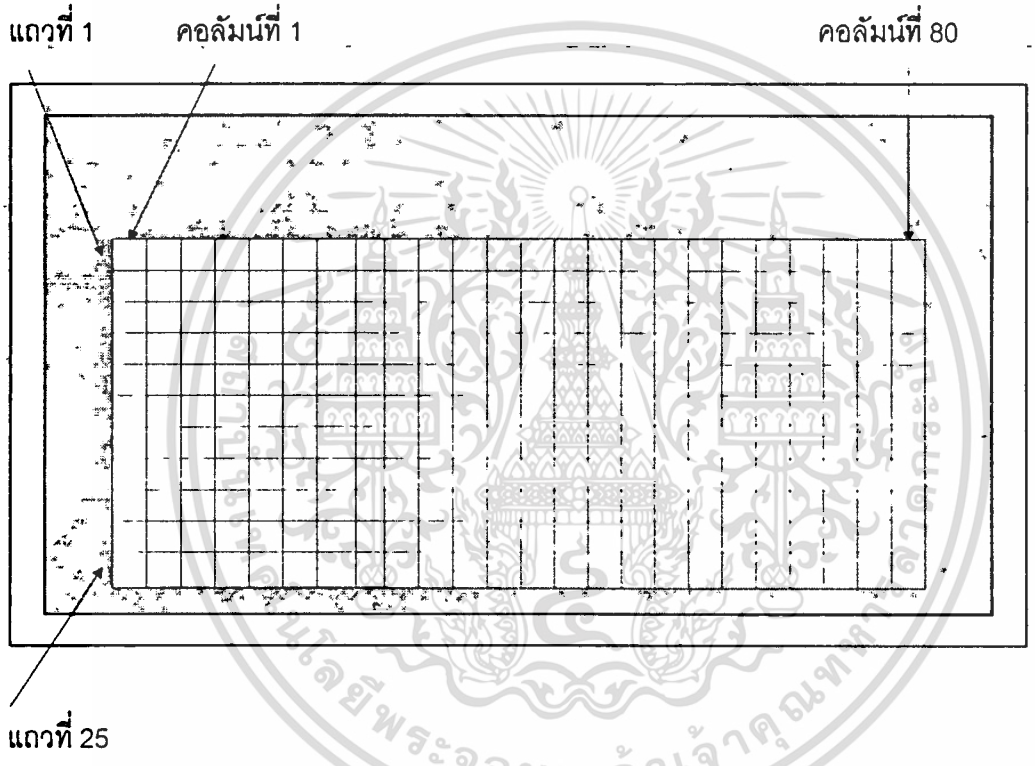
จอภาพของพีซีในโหมดข้อความจะมีการแบ่งหน้าจอออกเป็นตำแหน่งของตัวอักขระต่าง ๆ กันในรูปแบบของตาราง แต่ละตำแหน่งของตัวอักษรนั้นจะประกอบด้วยสองส่วนคือส่วนของ "ข้อมูล" ซึ่งใช้ในการพิจารณาดูความเป็นไปได้ที่จะนำตัวอักษรไปแสดงผล ส่วนที่สองคือส่วนของ "ลักษณะเฉพาะ" (attribute) ซึ่งจะเป็นตัวบ่งบอกว่าจะแสดงตัวอักษรออกมาในลักษณะใด

(เช่น สีอะไร ตัวกระพริบ ฯลฯ)

โปรแกรมที่เขียนให้ทำงานในโหมดข้อความสามารถควบคุมทั้งส่วนของข้อมูล และ ลักษณะเฉพาะได้อย่างเต็มที่ ดังนั้นจึงสามารถระบุได้ว่าจะนำตัวอักขระใดไปแสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผล และจะแสดงผลของบนหน้าจอที่ใด ด้วยคุณลักษณะแบบใด อย่างไรก็ตามโปรแกรมที่เขียนขึ้นนี้จะไม่สามารถควบคุมในรายละเอียดอื่น ๆ ได้ทั้งหมดเช่นวิธีในการวาดภาพตัวอักษรและการควบคุมตำแหน่งของอักขระบนจอภาพ สิ่งต่าง ๆ เหล่านี้จะถูกกำหนดโดยความสามารถของอะแดปเตอร์แสดงผลและจอภาพ (ทำได้โดยการเขียนโปรแกรมควบคุมความสามารถส่วนนี้ในโหมดกราฟิก)



รูปที่ 4.1 การแบ่งหน้าจอออกเป็นตาราง

โดยสรุปแล้วการทำงานในโหมดข้อความ จะทำได้โดยอาศัยโครงสร้างของจอซึ่งแบ่งเป็นออกเป็นช่อง ๆ คล้ายตารางที่มีลักษณะที่แน่นอน การกำหนดโครงร่างของจอแบบนี้ มีประโยชน์ต่อผู้เขียนโปรแกรมในแง่ที่สามารถลดความยุ่งยากของรายละเอียดเหล่านี้ลงไปได้

จากรูปที่ 4.1 แสดงให้เห็นถึงการจัดโครงสร้างโดยแบ่งเป็นตาราง ตำแหน่งตัวอักษรบนหน้าจอจะถูกแบ่งออกเป็น 25 แถว ซึ่งโดยปรกติจะมีจำนวนคอลัมน์ 80 คอลัมน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้ทั่วไปจะมีการมองจอภาพในลักษณะที่แบ่งออกเป็นบรรทัดหรือในลักษณะที่แสดงเป็นชุดของตัวอักษรที่ยาวติดต่อกันถึง 2000 ตัว (25 แถว แถวละ 80 ตัว) ก็ได้ โดยเครื่องพีซีจะถูกออกแบบมาให้ใช้งานได้ทั้งสองวิธี กล่าวคือในขณะที่เขียนโปรแกรม จะสามารถเขียนโปรแกรมได้ในลักษณะที่ทำการควบคุมการแสดงผลของโปรแกรมให้มีแสดงข้อมูลในแถวและคอลัมน์ที่ระบุ หรือสามารถเขียนโปรแกรมในลักษณะที่ทำให้เกิดการแสดงผลของข้อมูลที่ยาวติดต่อกันออกทางหน้าจอได้ ซึ่งในกรณีหลัง เมื่อมีการแสดงผลไปถึงปลายสุดบรรทัด เครื่องจะทำการขึ้นบรรทัดใหม่ให้โดยอัตโนมัติ สำหรับจอภาพของเครื่องพีซีจะมีความสามารถในการทำงานได้ทั้งสองวิธีดีพอ ๆ กัน เพื่อให้เกิดความยืดหยุ่นสูงที่สุดในการทำงาน

รหัส	ลักษณะที่ปรากฏ
0	สีดำ (ทำให้มองไม่เห็นเพราะเป็นสีเดียวกับสีพื้น)
1	สีน้ำเงิน
2	สีเขียว
3	สีฟ้าอมเขียว (น้ำเงิน + เขียว)
4	สีแดง
5	สีม่วง (น้ำเงิน + แดง)
6	สีเหลืองอ่อน (เขียว + แดง)
7	สีขาว (น้ำเงิน + เขียว + แดง)
8	สีเทา (ลำแสงสว่าง)
9	สีน้ำเงินเข้ม (น้ำเงิน + ลำแสงสว่าง)
10	สีเขียวเข้ม (เขียว + ลำแสงสว่าง)
11	สีฟ้าอมเขียวเข้ม (น้ำเงิน + เขียว + ลำแสงสว่าง)
12	สีแดงเข้ม(แดง + ลำแสงสว่าง)
13	สีม่วงเข้ม (น้ำเงิน + แดง + ลำแสงสว่าง)
14	สีเหลืองเข้ม (เขียว + แดง + ลำแสงสว่าง)
15	สีขาวเข้ม (น้ำเงิน + เขียว + แดง + ลำแสงสว่าง)

รูปที่ 4.2 ค่าลักษณะเฉพาะในการแสดงสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จอภาพสามารถแสดงตัวอักษรทุกตัวในชุดตัวอักษรของเครื่องพีซี แต่การแสดงผลตัวอักขระบางตัว อาจต้องใช้เทคนิคพิเศษบางอย่าง ทั้งนี้เนื่องจากตัวอักขระบางตัวเป็นตัวอักษรควบคุม ซึ่งอาจจะมีความหมายพิเศษต่าง ๆ รวมทั้งยังส่งผลต่อการแสดงผลออกทางหน้าจอเช่นส่งผลถึงการขึ้นบรรทัดใหม่เมื่อมีการกดปุ่มเหล่านี้เป็นต้น ตัวอักขระควบคุมนี้มีรหัสแอสกี 0 ถึง 31 ถ้าตัวอักษรเหล่านี้ถูกเขียนออกหน้าจอโดยใช้คำสั่ง "PRINT " จะมีผลในการทำงานเหมือนกับการใช้ตัวอักขระควบคุม ส่งผลให้เกิดการทำงานตามความหมายของอักษรควบคุมเหล่านั้น แต่ถ้าไม่นับอักขระควบคุมเหล่านี้ ในการทำงานจะสามารถนำตัวอักษรอื่น ๆ ในเครื่องตระกูลพีซีมาแสดงผลบนหน้าจอได้โดยง่าย และยังส่งผลให้สามารถทำการแสดงผลที่ตำแหน่งใดของหน้าจอก็ได้

ตัวอักษรที่ปรากฏออกมาบนหน้าจอคือส่วนของข้อมูล ส่วนค่าของลักษณะเฉพาะนั้นจะเป็นส่วนที่ใช้ในการควบคุมการแสดงผลของตัวอักษร ว่าโดยจะควบคุมรูปแบบในการแสดงผลเช่น ควบคุมการแสดงสีของตัวอักษรเป็นต้น

รหัส	ลักษณะที่ปรากฏ
0	มองไม่เห็น
1	ขีดเส้นใต้
7	ตัวปรกติ
009 (8 + 1)	ตัวสว่างขีดเส้นใต้
015 (8 + 7)	ตัวสว่าง
112	ตัวกลับดำ-ขาว
129 (128 + 7)	ตัวกะพริบขีดเส้นใต้
135 (128 + 7)	ตัวกะพริบ
137 (128 + 8 + 1)	ตัวกะพริบสว่างและขีดเส้นใต้
143 (128 + 8 + 7)	ตัวกะพริบสว่าง
240 (128 + 112)	ตัวกะพริบกลับขาวดำ

รูปที่ 4.3 ค่าลักษณะเฉพาะของโมโนโครม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะเฉพาะ หรือรหัสควบคุมจะใช้ในการพิจารณาว่าตัวอักขระจะถูกมาแสดงบนจอภาพอย่างไร ซึ่งข้อมูลที่จะนำมาแสดงในแต่ละตำแหน่งจะมีลักษณะเฉพาะของตัวเองไม่ขึ้นกับค่าลักษณะเฉพาะในตำแหน่งอื่น โดยทั่วไปแล้วรหัสควบคุมจะมี 2 ชุด คือ ชุดหนึ่งจะถูกออกแบบมาสำหรับโมโนโครม และอีกชุดหนึ่งจะถูกออกแบบมาสำหรับการแสดงผลแบบสี แต่รหัสควบคุมทั้งสองชุดนี้ได้ถูกออกแบบมาให้โครงสร้างทั้งสองมีความเข้ากันได้

สำหรับการแสดงสีในแต่ละตำแหน่งนั้น จะควบคุมโดยลักษณะเฉพาะซึ่งจะประกอบด้วย 3 ส่วน ส่วนแรกจะเป็นส่วนที่ใช้ระบุสีที่ใช้ในการแสดงตัวอักขระ (forground color) ส่วนที่สองจะควบคุมสีพื้น (background color) ส่วนสุดท้ายจะใช้ควบคุมการกระพริบของตัวอักษร สำหรับสีของอักขระจะมี 16 สี คือ เบอร์ 0 ถึง เบอร์ 15 ดังแสดงในรูปที่ 4.2 สีต่าง ๆ จะเกิดจากการรวมกันของสี 3 สีคือสีแดง สีเขียว และสีน้ำเงิน จากการรวมกันของสีทั้งสามจะทำให้เกิดสีหลัก ๆ 8 สี และเมื่อมีการรวมวิธีการแสดงผลแบบตัวปรกติ หรือ ตัวสว่างของแต่ละสีแล้ว ทำให้เกิดสีรวมทั้งหมด 16 สี ส่วนสีพื้นนั้นจะมีอยู่ทั้งหมด 8 สีเช่นเดียวกับสีของอักขระแต่ไม่รวมการทำตัวอักษรตัวสว่าง สำหรับ ส่วนสุดท้ายของลักษณะเฉพาะในการแสดงสีคือความสามารถในการควบคุมการกระพริบของตัวอักษร

ในการแสดงผลแบบโมโนโครมจะใช้ลักษณะเฉพาะควบคุมลักษณะการปรากฏของตัวอักษรด้วย แต่แตกต่างจากที่กล่าวมาแล้ว ทั้งนี้เพราะโมโนโครมมีการใช้สีเพียงสีเดียว ส่วนลักษณะการแสดงผลของตัวอักษรแบบต่าง ๆ เช่น ตัวปรกติ ตัวเข้ม ตัวกระพริบ ตัวขีดเส้นใต้ หรือ ตัวกลับขาวดำ (ตัวอักษรดำบนพื้นขาว) ยังมีผลต่อการแสดงผลของตัวอักษร โดยสามารถนำเอาลักษณะต่าง ๆ เหล่านี้มาใช้ร่วมกันได้ ในรูปที่ 4.3 จะแสดงลักษณะเฉพาะในโหมดของโมโนโครมที่เป็นไปได้ทั้งหมด แต่ก็ใช่ว่าจะสามารถรวมคุณสมบัติทุกตัวเข้าด้วยกันได้ เช่นจะไม่สามารถแสดงตัวกลับขาวดำแบบขีดเส้นใต้ได้ เป็นต้น

สิ่งที่กล่าวมาทั้งหมดคือลักษณะที่สำคัญในการแสดงผลแบบตัวอักษร ส่วนรายละเอียดทางเทคนิคที่เกี่ยวกับการจัดรูปแบบโครงสร้างของข้อมูลที่อยู่ในหน่วยความจำแสดงผล รวมถึงการเข้ารหัสต่าง ๆ ของลักษณะเฉพาะว่ามีการเข้ารหัสกันอย่างไร และรายละเอียดที่น่าสนใจอื่น ๆ จะมีการ กล่าวถึงภายหลัง

ตัวควบคุมวีดีโอและโปรเซสเซอร์จะใช้ข้อมูลของหน่วยความจำแสดงผลร่วมกัน หน่วยความจำที่กล่าวนี้ไม่ใช่ส่วนที่อยู่ในหน่วยความจำหลัก แต่เป็นส่วนที่มาพร้อมกับชิปวีดีโอ ในกรณีของวีดีโอรุ่นใหม่ที่เป็นแถวลำดับ (array) แต่ถ้าเป็นรุ่นเก่าที่เป็นอะแดปเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำจะอยู่บนบอร์ดอะแดปเตอร์นั่นเอง อย่างไรก็ตาม การอ้างอิงถึงข้อมูลที่อยู่ในหน่วยความจำแสดงผลสามารถทำได้โดยการใช้ตำแหน่ง (address) ที่ถือเป็นส่วนหนึ่งของพื้นที่ใช้งานของดอสและ 8086

(อันได้แก่พื้นที่ขนาด 1 เมกะไบต์) โดยตำแหน่งที่ว่านี้คือ ส่วนของบล็อก A และ B (บล็อก A เริ่มตั้งแต่ A000 - AFFF ส่วนบล็อก B จะอยู่ตั้งแต่ B000 - BFFF)

ตารางในรูปที่ 4.4 แสดงถึงโหมดการแสดงผลแต่ละโหมดพร้อมด้วยตำแหน่งพื้นที่ที่ใช้ สิ่งที่น่าสังเกตคือโหมดที่ใช้กับวีดีโอรุ่นเก่า (CGA และ MDA) จะใช้บล็อก B โดย CGA จะเริ่มใช้ข้อมูลตั้งแต่ B000 และใช้พื้นที่ครึ่งแรกของบล็อก ในขณะที่ MDA จะใช้พื้นที่ส่วนหลังของบล็อก B คือ เริ่มตั้งแต่ B800 ส่วนบล็อก A จะถูกใช้กับโหมดใหม่

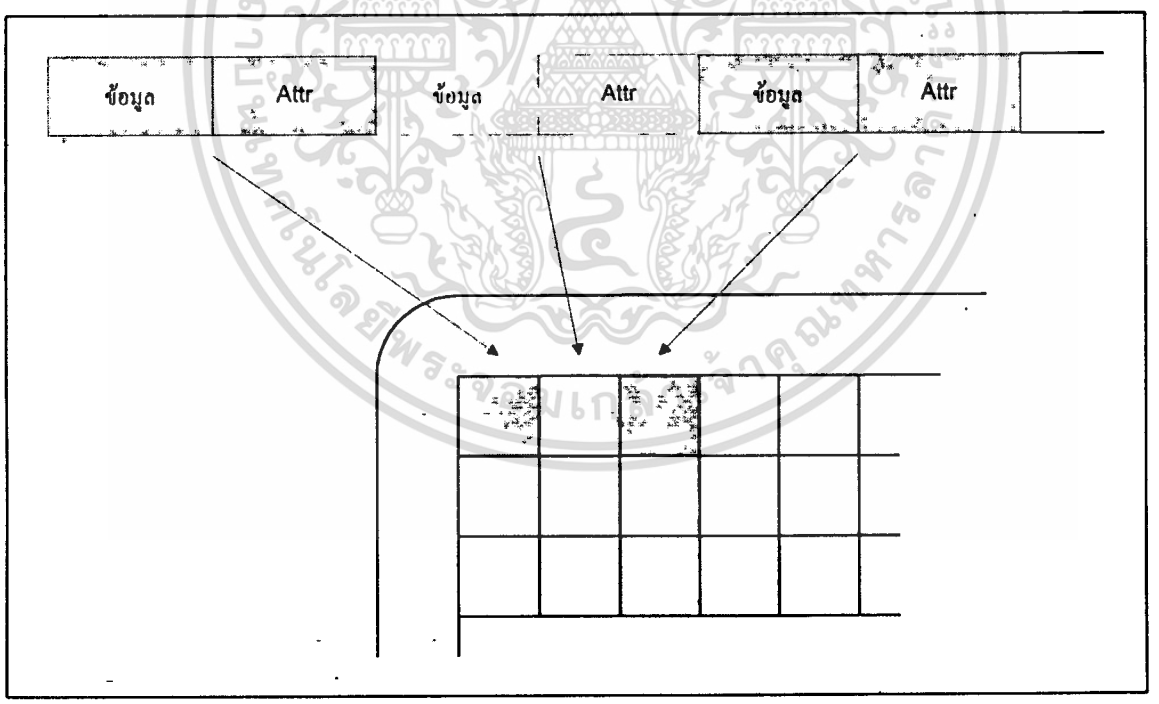
โหมด	ตำแหน่ง	จำนวนไบต์	วีดีโอมาตราฐาน
0 1	B800	2000	VGA MCGA EGA CGA
2 3	B800	4000	VGA MCGA EGA CGA
4 5	B800	16000	VGA MCGA EGA CGA
6	B800	16000	VGA MCGA EGA CGA
7	B000	4000	VGA EGA MDA
D	A000	32000	VGA EGA
E	A000	64000	VGA EGA
F	A000	56000	VGA EGA
10	A000	112000	VGA EGA
11	A000	38000	VGA MCGA
12	A000	153000	VGA
13	A000	64000	VGA MCGA

รูปที่ 4.4 โหมดแสดงผลและตำแหน่งที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปคือแต่ละโหมดจะใช้เนื้อที่ในหน่วยความจำแต่ละส่วน แต่รูปแบบการใช้หน่วยความจำจะเหมือนกันหมด กล่าวคือ จะใช้เนื้อที่ 2 ไบต์ สำหรับการแสดงตัวอักขระในแต่ละตำแหน่ง โดยเริ่มจากไบต์แรกจะบรรจุอักขระที่ต้องการแสดงออกหน้าจอ ณ ตำแหน่งมุมบนซ้าย ส่วนในไบต์ถัดมาก็จะเก็บค่าลักษณะเฉพาะของอักขระตำแหน่งนี้ ข้อมูลในไบต์คู่ถัดมาจะใช้เก็บตัวอักขระและลักษณะเฉพาะที่จะแสดงผลในจอคอลัมน์ที่ 2 ของบรรทัดแรก ซึ่งจะมีลักษณะเรียงกันไปเรื่อยๆ จนกว่าจะครบถึงตำแหน่งสุดท้าย คือมุมล่างด้านขวาดังเช่นในรูป 4.5

ข้อมูลที่เก็บในหน่วยความจำแสดงผลจะถูกมองเสมือนหนึ่งว่าเป็นชุดของไบต์คู่ติดกัน 2000 คู่ โดยไม่มีการแบ่งเป็นบรรทัด ๆ ดังนั้นการแสดงผลบนหน้าจอจึงต้องมีการปัดขึ้นบรรทัดใหม่ให้โดยอัตโนมัติเมื่อถึงจุดสิ้นสุดบรรทัด ซึ่งข้อมูลในหน่วยความจำแสดงผลจะถูกเก็บเรียงแบบไบต์ต่อไบต์ ดังนั้นถ้าหากต้องการทราบว่าตำแหน่งของการแสดงผลว่าจะอยู่ที่บรรทัดใด คอลัมน์อะไร ก็สามารถคำนวณตำแหน่งที่ตรงกับตำแหน่งบนหน้าจอได้



รูปที่ 4.5 หน่วยความจำแสดงผลบนจอภาพในโหมดข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการคำนวณหาตำแหน่งที่อยู่ในหน่วยความจำ (นับจากตำแหน่งที่เริ่มต้นของหน่วยความจำแสดงผล) สำหรับตำแหน่งบนหน้าจอสามารถกระทำด้วยสูตรง่าย ๆ โดยเริ่มจากการใส่ค่าบรรทัดและ คอลัมน์ที่ต้องการลงไปในสูตร และกำหนดให้ทั้งบรรทัดและคอลัมน์แรกเริ่มต้นที่ 0 (แทนที่จะเป็น 1)

```
LOCATION = (ROW * 80 + CLOUMN) * 2
```

ตำแหน่งที่ได้นี้จะเป็นตำแหน่งที่เทียบกับจุดเริ่มต้นของหน่วยความจำแสดงผล และสูตรนี้จะใช้ได้เฉพาะกับโหมดการแสดงผลแบบ 80 คอลัมน์ ซึ่งถ้าเป็นแบบ 40 คอลัมน์ จะต้องคูณจำนวนแถวด้วย 40 แทน 80

```
LOCATION = (ROW * 40 + CLOUMN) * 2
```

ส่วนตำแหน่งที่เก็บค่าลักษณะเฉพาะของข้อมูลแต่ละตำแหน่งจะอยู่ในไบต์ถัดไปจึงต้องบวกเข้าไปอีก 1 ถ้าต้องการค่าลักษณะเฉพาะของการที่จะให้หน้าจอแสดงข้อมูลนั้น จริง ๆ แล้วมีเพียงวิธีเดียวที่ทำได้ คือการวางข้อมูลเหล่านั้นไว้ในหน่วยความจำแสดงผล ณ ตำแหน่งที่เหมาะสม ซึ่งสามารถทำได้โดยการเขียนโปรแกรมใส่ข้อมูลเข้าไปโดยตรง หรือสามารถเรียกใช้บริการของไบออสที่ติดมากับเครื่องก็ได้ ส่วนการเลือกใช้วิธีไหนนั้นก็ขึ้นอยู่กับเหตุผลและความต้องการของผู้เขียนโปรแกรม ซึ่งถ้าผู้ใช้เขียนโปรแกรมเองก็จะได้การแสดงผลออกหน้าจอที่ความรวดเร็วและมีประสิทธิภาพสูงมาก โปรแกรมส่วนใหญ่ที่มีการแสดงผลหน้าจอเร็ว ๆ มักจะใช้วิธีนี้ แต่การที่จะเขียนโปรแกรมให้มีความสามารถเช่นนี้ได้ จำเป็นที่จะต้องรู้โครงสร้างของหน่วยความจำแสดงผล และต้องทราบถึงข้อมูลที่เกี่ยวกับการทำงานของอะแดปเตอร์แสดงผลด้วย อย่างไรก็ตามโปรแกรมเหล่านี้จะไม่สามารถทำงานได้กับอะแดปเตอร์แสดงผลที่มีการใช้หน่วยความจำแสดงผลที่มีตำแหน่งแตกต่างไป ในทางตรงข้ามถ้าเขียนโปรแกรมให้ทำงานโดยผ่านทางการบริการของรอมไบออส (ใช้จัดการการเก็บข้อมูลลงในหน่วยความจำแสดงผล) ก็สามารถที่จะเปลี่ยนจอภาพที่ใช้งานได้โดยง่าย เพราะรอมไบออสจะทำหน้าที่เป็นสื่อกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่างโปรแกรมที่ทำงานกับชนิดของจอภาพและหน่วยความจำแสดงผลแบบต่าง ๆ จึงทำให้โปรแกรมของสามารถปรับตัวให้เข้ากับเครื่องคอมพิวเตอร์ที่กำลังทำงานอยู่ได้

จากที่กล่าวมา ดูเหมือนว่าผู้เขียนโปรแกรมควรจะใช้บริการของรอมไบออสเมื่อต้องการแสดงผลออกทางจอภาพ ทั้งนี้เพื่อให้เกิดความยืดหยุ่นในการทำงาน แต่กระนั้นการใช้บริการผ่านรอมไบออสก็มีข้อเสีย นั่นคือการทำงานจะมีความช้ามาก เนื่องจากเหตุผลดังกล่าวจึงเห็นได้ว่าโปรแกรมที่ต้องทำงานร่วมกับหน้าจอแบบต่าง ๆ ส่วนใหญ่จะเขียนข้อมูลลงใส่หน่วยความจำแสดงผลโดยตรง

สำหรับบุคคลทั่ว ๆ ไปที่ใช้เครื่องคอมพิวเตอร์ มักมีความเชื่อที่ผิดพลาดเกี่ยวกับความสัมพันธ์ระหว่างเคอร์เซอร์กับข้อมูลที่แสดงออกทางหน้าจอ โดยเชื่อว่าข้อมูลที่แสดงออกสู่หน้าจอจะสามารถแสดงได้เฉพาะตรงตำแหน่งที่มีเคอร์เซอร์อยู่เท่านั้น ในความเป็นจริงมิได้เป็นเช่นนั้น ข้อมูลที่แสดงผลออกทางหน้าจอจะได้มาจากการวางข้อมูลเหล่านั้นลงในหน่วยความจำแสดงผล ซึ่งไม่มีความเกี่ยวข้องกับเคอร์เซอร์แต่อย่างใด ตัวเคอร์เซอร์ถูกใช้เพียงเพื่อเป็นตัวแสดงตำแหน่งของหน้าจอที่กำลังถูกใช้งานอยู่ ซึ่งจะมีประโยชน์ในการปรับปรุงการมองเห็นบนจอภาพ

เพื่อช่วยให้ผู้ใช้มีความสะดวกในการทำงาน งานบริการของรอมไบออสจึงทำการแสดงผลออกสู่หน้าจอที่ตำแหน่งเคอร์เซอร์ สำหรับรอมไบออสแล้วเคอร์เซอร์ไม่ใช่ตัวชี้ที่ปรากฏอยู่บนหน้าจอเพื่อช่วยในการมองเห็นเท่านั้น แต่ยังหมายถึงตำแหน่งบนหน้าจออีกด้วย โดยการใช้เคอร์เซอร์จะเป็นวิธีเดียวที่ทำให้รอมไบออสและโปรแกรมอื่น ๆ สามารถแสดงผลที่ตำแหน่งเดียวกันบนหน้าจอ

4.1.3.2 การเข้ารหัส

สำหรับเรื่องของการการเข้ารหัสของค่าลักษณะเฉพาะที่ใช้ในการควบคุมการแสดงผลตัวอักษรบนหน้าจอจะมีวิธีการทำได้ดังที่จะกล่าวต่อไปนี้ โดยค่าลักษณะเฉพาะที่ใช้ในงานของจอสีและโมโนโครมจะมีความแตกต่างกัน แต่พื้นฐานการออกแบบจะมีลักษณะร่วมกันอยู่

ข้อมูลในแต่ละค่าลักษณะเฉพาะทั้ง 8 บิตจะถูกแบ่งออกเป็น 4 ส่วน ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7	6	5	4	3	2	1	0	
B	.	.	.	.	.	.	.	กะพริบ (แสดงผล)
.	R	G	B	.	.	.	.	สีพื้น
.	.	.	.	I	.	.	.	สีเข้ม
.	.	.	.	.	R	G	B	สีตัวแสดงผล

ดังภาพจะเห็นได้ว่าบิต-4 ตัวที่อยู่ขวาสุดจะควบคุมสีของอักขระโดยใช้ 3 บิตเพื่อระบุถึงส่วนประกอบของสี (สีแดง สีเขียว สีน้ำเงิน) และใช้บิตที่ 4 สำหรับควบคุมความเข้มของสี ส่วน 4 บิตที่เหลือจะใช้ในการควบคุมสีพื้น โดยบิตที่ 4 (บิตซ้ายสุด) จะถูกใช้ในการควบคุมการกะพริบของตัวแสดงผล นอกจากนี้เรายังสามารถนำผลที่ได้จากการเอาทุก ๆ บิตเหล่านี้มาผสมกันได้อีกด้วยดังเช่น ถ้าหากต้องการแสดงตัวอักษรสีฟ้าเข้มและให้กะพริบบนพื้นสีแดง ก็จะต้องเซ็ตค่าลักษณะเฉพาะ เป็น 11001001 หรือ C9

ในการแสดงสีตามที่ต้องการ สามารถกระทำได้ด้วยการเซ็ตค่าตามที่ระบุในตารางดังกล่าวส่วนการทำงานในโหมดโมโนโครมก็อาศัยหลักการเดียวกัน เช่น การแสดงผลในโหมดปรกติ (ตัวหนังสือขาวบนพื้นดำ) จะเซ็ตค่าลักษณะเฉพาะเท่ากับ 07 (ฐาน 16) หรือ 00000111 (ฐานสอง) ซึ่งเทียบเท่ากับการแสดงสีตัวขาวบนพื้นดำ สำหรับการแสดงผลในลักษณะกลับกัน (reverse video) ก็จะตรงข้ามกันคือ 70 ซึ่งก็เหมือนกับตัวดำบนพื้นขาว สำหรับรหัสของการขีดเส้นใต้จะใช้ค่า 01 ซึ่งเทียบเท่ากับจอสีที่แสดงตัวอักษรสีน้ำเงิน ถ้าเราต้องการเซ็ตไม่ให้มองเห็นการทำงานในโหมดโมโนโครม ก็จะทำได้โดยการเซ็ตให้เป็นตัวอักษรสีดำและพื้นสีดำด้วย ซึ่งก็คือ รหัส 00 ซึ่งถ้าเป็นตัวอักษรขาวบนพื้นขาวจะมีรหัส 77 ทำให้ผลที่ได้จะอยู่ในโหมดที่มองไม่เห็นเช่นกันเพราะหน้าจอสว่างหมด แต่ในความเป็นจริง การทำงานในโหมดโมโนโครมจะสามารถผสมลักษณะเฉพาะได้ดังแสดงในรูปที่ 4.3 เท่านั้น โดยจะไม่สามารถผสมทุกบิตได้ดังที่ต้องการ

แม้ว่าการทำงานแบบโมโนโครมจะมีวิธีการผสมลักษณะเฉพาะที่ใช้ได้ในจำนวนจำกัด แต่การทำงานยังสามารถทำได้ไม่ว่าจะตั้งค่าลักษณะเฉพาะเป็นอะไรก็ตาม นั่นคือถ้าเกิดมีการสแกนพบค่าที่ไม่รู้จัก ก็จะแสดงผลด้วยรูปแบบมาตราฐานเอาไว้ก่อน ซึ่งโดยส่วนใหญ่ก็ได้แก่ การแสดงตัวอักษรในวิธีปรกติ คือ เหมือนกับได้รับค่าลักษณะเฉพาะรหัส 07

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทำงานพื้นฐานของแป้นพิมพ์

ก่อนที่จะศึกษาการทำงานของแป้นพิมพ์ จะต้องคำนึงถึงหลักการในการทำงานของแป้นพิมพ์ก่อน โดยเมื่อมีการกดปุ่มใด ๆ บนแป้นพิมพ์ การส่งตัวอักษรบนปุ่มนั้นเข้าสู่คอมพิวเตอร์มิได้กระทำโดยทันที เช่นเมื่อกดปุ่ม A ก็มิได้หมายความว่ามีการส่งตัวอักษร "A" ในทันที แต่การทำงานจะผ่านกลไกหลายขั้นตอน ทั้งนี้เนื่องจาก

1. ถึงแม้ว่าฮาร์ดแวร์ของคอมพิวเตอร์จะมีประสิทธิภาพสูงมาก แต่ก็ไม่อยู่ในสถานะที่จะทำงานต่าง ๆ เองได้ ฮาร์ดแวร์ไม่สามารถให้ความหมายหรือแปลความในสิ่งต่าง ๆ ที่ทำไว้ ในทางกลับกัน ซอฟแวร์จะเป็นส่วนที่ทำให้ฮาร์ดแวร์ทำงาน รวมทั้งสามารถให้ความหมายต่าง ๆ หรือแปลความต่าง ๆ ได้ ดังนั้นเมื่อกดปุ่ม A จึงไม่ได้หมายถึงตัวอักษร "A" ทั้งนี้เนื่องจากแป้นพิมพ์เป็นฮาร์ดแวร์ย่อมไม่สามารถทำงานในเรื่องการแปลความซึ่งเป็นหน้าที่ของซอฟแวร์ได้

2. หากการทำงานของฮาร์ดแวร์มีการแปลความหมายได้ ดังเช่นถ้ากดปุ่ม A แล้วมีความหมายว่าเป็นตัวอักษร "A" แล้ว จะทำให้การทำงานไม่มีความยืดหยุ่น เนื่องจากสิ่งที่สำคัญในการทำงานของคอมพิวเตอร์คือความยืดหยุ่น สามารถเปลี่ยนแปลงไปตามสถานะได้ ดังนั้นถ้าฮาร์ดแวร์ของคอมพิวเตอร์มีความยืดหยุ่น จะทำให้ประสิทธิภาพในการทำงานสูง

เมื่อมีการกดปุ่มทำงานบนแป้นพิมพ์ จะเกิดขั้นตอนการทำงานดังต่อไปนี้คือ

- แป้นพิมพ์รับรู้ถึงความเปลี่ยนแปลงที่เกิดขึ้น
- เริ่มทำการบันทึกรหัสสแกนเอาไว้ (ปุ่มแต่ละปุ่มจะมีรหัสเฉพาะเรียกว่ารหัสสแกน)

หลังจากนั้นแป้นพิมพ์จะส่งการขัดจังหวะ (interrupt) หมายเลข 9 ให้แก่โปรเซสเซอร์เพื่อแสดงว่าขณะนี้ได้มีการเปลี่ยนแปลงสถานะของแป้นพิมพ์เกิดขึ้น ซึ่งจะทำให้โปรเซสเซอร์หยุดการทำงานใด ๆ ที่กำลังทำอยู่ชั่วคราว เพื่อกระโดดการทำงานไปยังโปรแกรมที่จัดการการขัดจังหวะก่อนอันได้แก่การทำงานในส่วนของรอมไบออสนั่นเอง

การทำงานส่วนนี้ รอมไบออสจะทำหน้าที่ดูแลการขัดจังหวะของแป้นพิมพ์ โดยการเข้ามาควบคุมการทำงานด้วยการตรวจสอบดูสถานะของแป้นพิมพ์ และส่งคำสั่งไปยังแป้นพิมพ์เพื่อตรวจสอบหาสถานะที่เกิดขึ้นจากแป้นพิมพ์ แป้นพิมพ์จะส่งคำตอบกลับสู่ไบออสเพื่อแสดงว่าปุ่มใดถูกกด การทำงานเหล่านี้จะดำเนินงานผ่านพอร์ตที่แป้นพิมพ์ใช้อยู่ จากนั้นแป้นพิมพ์จะส่งรหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สแกนของปุ่มที่ถูกกดไปยังตำแหน่งพอร์ตอีกตำแหน่งหนึ่ง ซึ่งเป็นตำแหน่งที่รอมไบออสใช้ในการอ่าน

ดังที่กล่าวมาแล้ว แป้นพิมพ์จะทำการเก็บค่ารหัสสแกนของปุ่มที่ถูกกดและรอจนกระทั่งไบออสเรียกใช้ ดังนั้นแป้นพิมพ์จึงมีหน่วยความจำขนาดเล็กอยู่ภายในแป้น เพื่อใช้ในการบันทึกค่ารหัสสแกนที่ยังไม่ถูกส่งไปให้กับไบออส หน่วยความจำส่วนนี้บางครั้งจะถูกเรียกว่า "บัฟเฟอร์ของแป้นพิมพ์" โดยทั่วไปหน่วยความจำส่วนนี้จะมีขนาดพอที่จะเก็บบันทึกรหัสสแกนไว้ได้จำนวนหนึ่ง ซึ่งส่วนสำคัญที่น่าสนใจในระบบการทำงานของแป้นพิมพ์คือ

1. แป้นพิมพ์จะเก็บรหัสสแกนเมื่อมีการกดปุ่มและตอนที่ปล่อยปุ่ม โดยค่ารหัสสแกนของการกดและ การปล่อยปุ่มนี้จะมีค่าแตกต่างกัน (กล่าวคือรหัสของการปล่อยปุ่มพิมพ์จะมีค่าเท่ากับรหัสการกดปุ่มบวกด้วย 128 หรือ 80 ในฐาน 16) นั่นคือเมื่อมีการกดปุ่มหนึ่งครั้งที่แป้นพิมพ์ (ซึ่งหมายถึงต้องมีการกดและปล่อยปุ่มนั่นเอง) จะทำให้ไบออสถูกขัดจังหวะ 2 ครั้ง ดังนั้นไบออสจึงทราบสถานะการทำงานของปุ่มว่าปุ่มกำลังถูกกดอยู่หรือปล่อยแล้ว การทำเช่นนี้จะทำให้ไบออสจะสามารถรับรู้ได้ว่ามีการกดปุ่มค้างหรือไม่ เช่นสามารถรับรู้ได้ว่าขณะนี้มีการกดปุ่มอักษรตัวใหญ่เพราะปุ่ม Shift ถูกกดค้างไว้ไม่ปล่อย

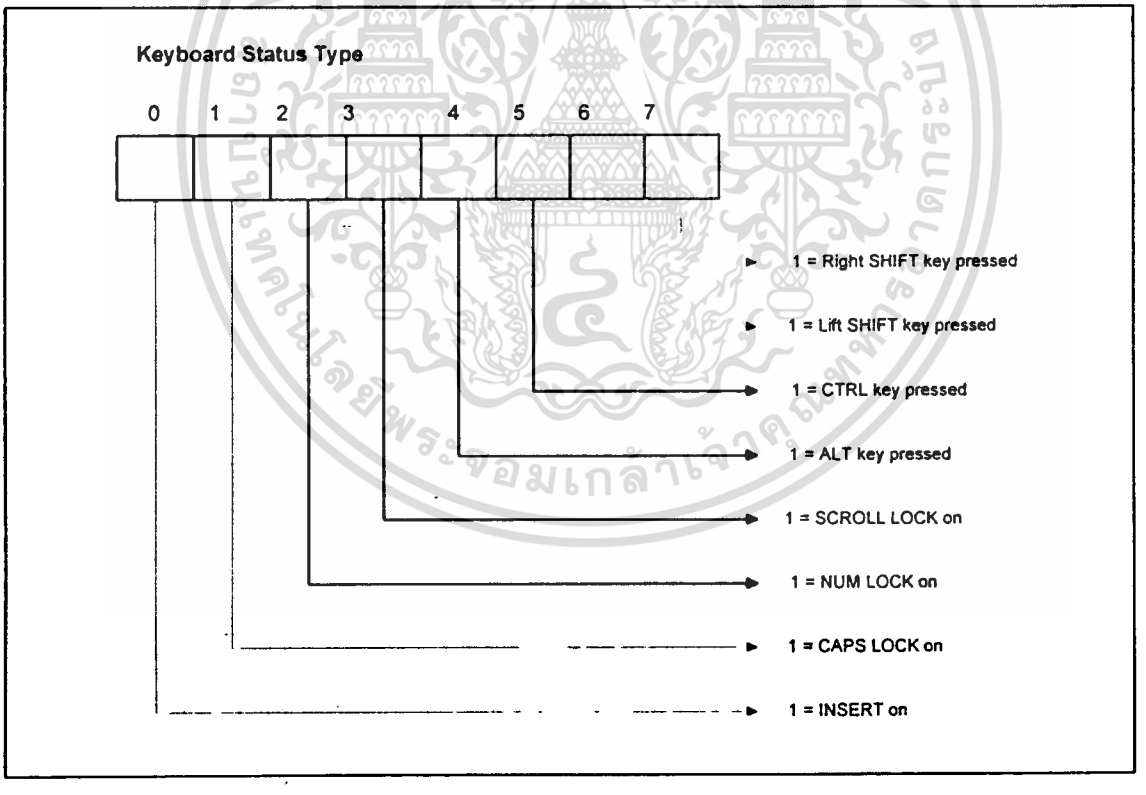
2. ในกรณีที่มีการกดปุ่มค้างไว้ไม่ปล่อย ฮาร์ดแวร์จะทำหน้าที่ตรวจสอบสถานะที่เกิดขึ้น โดยจะสามารถรับรู้ได้เพราะช่วงเวลาในการกดจะนานเกินกว่าที่กำหนดไว้ และแป้นพิมพ์ก็จะทำการส่งรหัสสแกนพิเศษ อันเป็นรหัสที่หมายถึงการกดปุ่มค้าง (Repeat Key) แทน

เมื่อตัวจัดการการขัดจังหวะที่ดูแลแป้นพิมพ์ของไบออสเริ่มทำงาน การขัดจังหวะจะรับค่ารหัสสแกนจากแป้นพิมพ์ (ในคีย์บอร์ดต่างชนิดจะมีรหัสสแกนไม่เหมือนกัน) และจะทำการแปลความหมายที่เกิดขึ้นโดยผ่านการวิเคราะห์หลายขั้นตอน

การทำงานจะเริ่มจากการตรวจสอบว่ามีการกดปุ่มประเภทพิเศษหรือไม่ ปุ่มพิเศษที่กำหนดไว้ได้แก่ ปุ่ม Shift ทางด้านซ้ายและขวา ปุ่ม Alt และปุ่ม Ctrl ซึ่งถ้ามีการเปลี่ยนแปลงสถานะของปุ่มเหล่านี้ จะมีการบันทึกว่าสถานะที่เกิดขึ้นอยู่ในสถานะ Shift ต่อจากนั้นไบออสจะตรวจสอบว่ามีการกดปุ่มประเภทท็อกเกิล (toggle) หรือไม่ ปุ่มประเภทนี้ได้แก่ ปุ่ม Capslock Numlock Scrolllock และ Ins

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความแตกต่างระหว่างปุ่มท็อกเกิลและ Shift คือลักษณะการใช้งานของปุ่มประเภท Shift นั้นจะต้องมีการกดค้างไว้และกดปุ่มอื่นตาม ซึ่งจะให้ความหมายที่แตกต่างจากการกดปุ่มอักขระดังกล่าวเพียงอย่างเดียว ส่วนการกดปุ่มประเภทท็อกเกิลจะกดเพื่อเปลี่ยนสถานะเช่นการกด Capslock จะเป็นการเปลี่ยนสถานะที่ส่งผลให้การกดปุ่มอักขระในเวลาต่อมากลายเป็นอักขระตัวใหญ่หมด โดยสรุปแล้วปุ่มทั้งสองประเภทนี้จะสามารถส่งผลต่อการแปลงรหัสสแกนได้ โดยปุ่มประเภท Shift จะส่งผลขณะที่มีการถูกกดค้างไว้เท่านั้น ในขณะที่ปุ่มประเภทท็อกเกิลจะส่งผลเมื่ออยู่ในสถานะ "On" สถานะทั้งสองประเภทนี้จะถูกเก็บในหน่วยความจำส่วนล่าง อันได้แก่ ตำแหน่ง 417 และ 418 โดยแต่ละบิตในแต่ละไบต์จะเก็บสถานะของปุ่มท็อกเกิลต่าง ๆ ว่าเป็น On หรือ Off นอกจากนั้นยังมีข้อมูลที่เก็บ ณ ตำแหน่ง 496 และ 497 เพื่อบอกสถานะว่าปุ่ม Alt หรือปุ่ม Ctrl ด้านขวามือถูกกดค้างไว้หรือไม่



รูปที่ 4.6 แสดงค่าสถานะของปุ่ม toggle และปุ่ม Shift

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่รอมไบออสได้ตรวจสอบว่ามีการเปลี่ยนสถานะของปุ่มประเภท Shift และท็อกเกิลหรือไม่ และอยู่ในสถาวะอะไรเป็นที่เรียบร้อยแล้ว จะมีการตรวจสอบต่อไปว่าได้มีการกดปุ่มประเภทที่จะทำให้เกิดความหมายพิเศษขึ้นหรือไม่ เช่นในการกดปุ่ม Ctrl-Alt-Del พร้อมกันจะทำให้เกิดการบูตดอสขึ้นมาใหม่ หรือการกดปุ่ม Pause ก็จะทำให้เกิดการหยุดการทำงานชั่วคราว

ถ้าภายหลังจากตรวจสอบพบว่ามีการกดปุ่ม 2 ประเภทนี้พร้อมกับการกดปุ่มอื่น ๆ อีกปุ่มหนึ่ง ไบออสจะจัดทำการแปลความหมาย เช่นเมื่อทำการตรวจสอบดู แล้วพบว่ามีการกดปุ่ม Shift และปุ่ม A พร้อมกัน ไบออสจะให้ความหมายว่าหมายถึง "A"

ความหมายเหล่านี้จะแบ่งได้เป็น 2 ประเภทคือ ประเภทแรกเป็นรหัสแอสกีที่เกิดจากการกดปุ่มแอสกีใด ๆ ส่วนประเภทที่สองได้แก่รหัสพิเศษที่เกิดจากการกดปุ่มพิเศษ เช่นปุ่มฟังก์ชันต่างๆหรือปุ่มที่ใช้เลื่อนเคอร์เซอร์เป็นต้น ในการเก็บบันทึกรหัสแอสกีและรหัสพิเศษของพีซีนั้น ไบออสจะใช้เนื้อที่ 2 ไบต์ในการเก็บ โดยถ้าเป็นตัวอักษรแอสกีก็จะเก็บรหัสแอสกีไว้ที่ไบต์แรก (และเก็บรหัสสแกนของปุ่มนั้นไว้ที่ไบต์สอง) แต่ถ้าเป็นอักขระพิเศษที่ไม่มีรหัสแอสกีก็จะเก็บ 0 ไว้ที่ไบต์แรกและเก็บรหัสของปุ่มคีย์พิเศษนี้ไว้ที่ไบต์ที่สอง

BIOS Variables For Keyboard Management

Offset	Meaning	Type
17H	Keyboard Status	1 byte
18H	Extended Keyboard Status	1 byte
19H	Code for ASCII input	1 byte
1AH	Next character in keyboard buffer	1 word
1CH	Last character in keyboard buffer	1 word
1EH	Keyboard buffer	16 word
80H	Start Address of keyboard buffer	1 word
82H	End Address of keyboard buffer	1 word

รูปที่ 4.7 แสดงเนื้อที่ในหน่วยความจำที่ไบออสใช้ในการจัดการคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการจัดการเกี่ยวกับแป้นพิมพ์ ไบออสจะใช้เนื้อที่ 8 ไบต์ในหน่วยความจำซึ่งอยู่ในเซกเมนต์ที่ 40 ในการจัดการแป้นพิมพ์ รวมทั้งใช้การติดต่อระหว่างการขัดจังหวะแฮนเดิลของแป้นพิมพ์ (09 H) และฟังก์ชันไบออสของแป้นพิมพ์ (16 H) ร่วมด้วย ซึ่งแสดงได้ดังตารางในรูปที่ 4.7

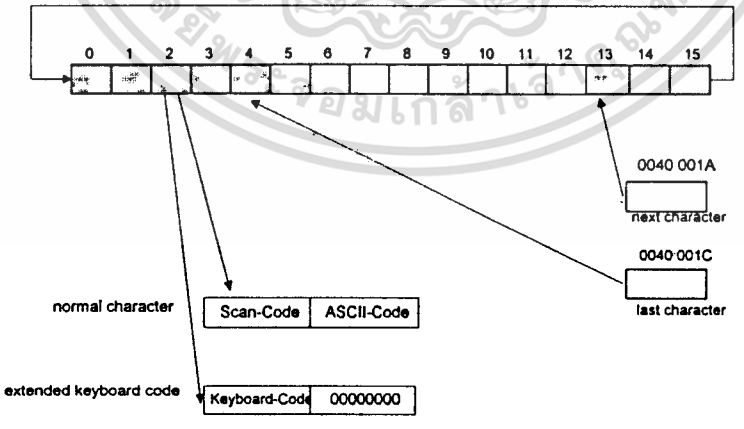
หน่วยความจำเหล่านี้จะใช้ประโยชน์สำหรับโปรแกรมแบบฝังตัวที่ใช้ในการเปลี่ยนการทำงานของแป้นพิมพ์ได้ จากตาราง ตำแหน่งที่ 19H จะใช้เพื่อป้อนรหัสแอสกี พร้อมกับปุ่ม Alt และปุ่มตัวเลข เมื่อปุ่มตัวเลขถูกกด ค่าของปุ่มนั้นจะถูกเก็บไว้ในไบต์นี้

ในการจัดการบัฟเฟอร์ของแป้นพิมพ์ จะใช้ตำแหน่งที่ 1AH 1CH 1EH เป็นตัวจัดการ โดยที่ตำแหน่งที่ 1EH จะเป็นบัฟเฟอร์ของแป้นพิมพ์ ซึ่งโครงสร้างจะมีลักษณะเป็นบัฟเฟอร์แบบวงแหวน บัฟเฟอร์นี้จะใช้เมื่อต้องการเขียนข้อมูลลงบัฟเฟอร์และอ่านข้อมูลจากบัฟเฟอร์เพื่อนำไปใช้ ซึ่งบัฟเฟอร์วงแหวนนี้จะใช้ตัวชี้สองตัว ตัวแรกสำหรับบอกถึงตัวอักษรตัวถัดไปที่จะอ่านได้จากบัฟเฟอร์ ส่วนตัวที่สองจะบอกถึงตำแหน่งที่จะเขียนตัวอักษรลงบัฟเฟอร์ครั้งต่อไป

จากหลักการนี้ การทำงานจะเริ่มจากการกำหนดให้ตัวชี้ทั้งสองตัวชี้ไปที่ตำแหน่งเริ่มต้นของบัฟเฟอร์ ตัวชี้จะมีการเคลื่อนไปเรื่อย ๆ ทุกครั้งที่มีการอ่านหรือเขียน เมื่อตัวชี้เลื่อนมาถึงตำแหน่งสุดท้ายของบัฟเฟอร์ก็จะเลื่อนกลับไปที่จุดเริ่มต้นของบัฟเฟอร์ใหม่

สำหรับความสัมพันธ์ของตัวชี้ทั้งสองนี้จะขึ้นอยู่กับเงื่อนไขที่สำคัญสองข้อ คือ

1. ถ้าตัวชี้ทั้งสองตัวมีค่าเหมือนกัน (ชี้ไปที่ตำแหน่งเดียวกัน) แสดงว่าบัฟเฟอร์ว่าง
2. ถ้าเกิดว่าตัวชี้ที่ชี้ไปที่ตำแหน่งสุดท้ายมีค่าน้อยกว่าตัวชี้ที่ชี้ไปที่ตำแหน่งแรกอยู่ 1 แสดงว่าบัฟเฟอร์เต็ม



รูป 4.8 บัฟเฟอร์ของแป้นพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
ทฤษฎีในการเขียนโปรแกรม

5.1 การเขียนโปรแกรมควบคุมการแมปหมวดบนเน็ตแวร์

5.1.1 ขอบเขตของโปรแกรม

โปรแกรมที่ทำการทดลองนี้จะมีความสามารถในการแมปหมวด (volume) ของเครื่องบริการที่ยังไม่ได้ทำการแมปให้กลายเป็นตัวขับ (drive) ถัดไป และในการแมปหมวดนั้นจะสามารถทำการแมปหมวดจากเครื่องบริการที่ติดต่อเข้ามายังเครื่องบริการแฟ้มหลัก (attach server)

5.1.2 ความรู้พื้นฐานในการเขียนโปรแกรม

ในการทำงานของสถานีงานจะมีการเก็บตารางเพื่อการทำงานต่าง ๆ ดังนี้

- ตารางที่ใช้เก็บชื่อของเครื่องบริการแฟ้ม (File Server Name Table)
- ตารางที่ใช้เก็บหมายเลขการเชื่อมต่อ (Connection ID Table)
- ตารางที่ใช้เก็แฮนเดิลของตัวขับ (Drive Handle Table)
- ตารางที่ใช้เก็บแฟล็กของตัวขับ (Drive Flag Table)
- ตารางที่ใช้เก็บคอนเน็คชันไอดีของตัวขับ (Drive Connection ID Table)

สำหรับโครงสร้างที่สำคัญต่าง ๆ ที่ใช้เขียนโปรแกรมควบคุมการแม็ปหมวดบนเน็ตแวร์จะประกอบด้วย

5.1.2.1 ลักษณะโครงสร้างไดเรกทอรีในเน็ตแวร์

ในระบบเน็ตแวร์มีการเก็บแฟ้ม (file) ต่าง ๆ เอาไว้เป็นลำดับชั้นเหมือนกับโครงสร้างไดเรกทอรีของดอส แต่มีบางส่วนเพิ่มเติมขึ้นมาเพื่อช่วยให้ผู้ใช้สามารถใช้งานระบบเครือข่ายได้สะดวกขึ้น โครงสร้างไดเรกทอรีของเน็ตแวร์จะเริ่มจากส่วนแรกซึ่งเป็นจานแข็งของเครื่องบริการแบ่งแยกเป็นลำดับชั้นย่อยลงไปทีละส่วน จนไปถึงแฟ้มที่เราใช้เก็บข้อมูล เมื่อมองดูจากแผนผังโครงสร้างจะมีลักษณะคล้ายกับรากของต้นไม้ โดยมีการแบ่งโครงสร้างแต่ละลำดับชั้นตามนี้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. จานแข็งของเครื่องบริการ

จะแบ่งออกเป็นส่วน ๆ แต่ละส่วนเรียกว่าหมวดซึ่งอาจจะมีหมวดเดียวหรือมีหลาย ๆ หมวดอยู่บนจานแข็งของเครื่องบริการก็ได้

2. หมวด

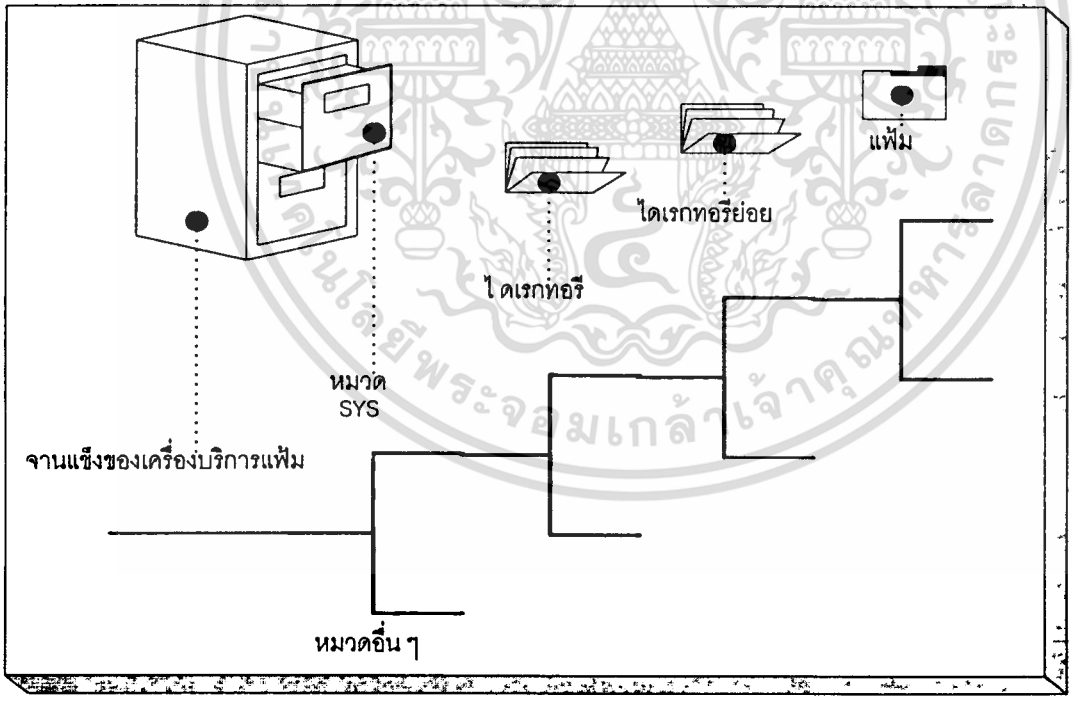
จะแบ่งเป็นส่วนย่อย ๆ เรียกว่าไดเรกทอรี ตามปกติในหนึ่งหมวดจะมีไดเรกทอรีรวมกันอยู่หลาย ๆ ไดเรกทอรี เทียบได้กับตัวขับของดอสนั่นเอง

3. ไดเรกทอรี

จะแบ่งออกเป็นไดเรกทอรีย่อย (subdirectory) หลาย ๆ ไดเรคทอรีย่อยอีกชั้นหนึ่ง

4. ไดเรกทอรีย่อย

จะเป็นที่เก็บแฟ้มต่าง ๆ หรืออาจแบ่งออกเป็นไดเรกทอรีย่อยลงไปอีกหลาย ๆ ชั้นก็ได้



รูปที่ 5.1 โครงสร้างของไดเรกทอรีเปรียบเทียบได้กับเอกสารที่เก็บอยู่ในตู้เอกสาร ซึ่งแบ่งการเก็บออกเป็นลำดับชั้นคือ แฟ้มย่อย, แฟ้มใหญ่ และลิ้นชัก รวมกันอยู่ในตู้เอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2.2 ตารางไดเรกทอรี (Directory Table)

เครื่องบริการแฟ้มจะทำหน้าที่ในการเก็บรักษาตารางไดเรกทอรี ซึ่งใช้ในการเก็บโครงสร้างไดเรกทอรีของหมวดต่าง ๆ อีกทั้งยังถูกใช้ในการเรียกใช้บริการต่าง ๆ จากไดเรกทอรี ข้อมูลที่เก็บในตารางไดเรกทอรี (รูปที่ 5.3) ได้แก่

1. รายละเอียดของไดเรกทอรี
2. ไฟล์ภายในไดเรกทอรี
3. ทรัสตีของไดเรกทอรี (Trustee)

ข้อมูลของไดเรกทอรีจะประกอบด้วยชื่อของไดเรกทอรี, คุณสมบัติของไดเรกทอรี,สิทธิในการทำงาน, วันที่สร้างไดเรกทอรี, ผู้สร้างไดเรกทอรี, ตัวชี้ (pointer) ที่ชี้ไปยังไดเรกทอรีพ่อ, ตัวชี้ไปยังทรัสตี

สำหรับข้อมูลของแฟ้มจะประกอบด้วยชื่อแฟ้ม, คุณสมบัติต่าง ๆ , ขนาดของแฟ้ม, วันที่ทำการสร้างแฟ้ม, ผู้สร้างแฟ้ม, วันที่ใช้แฟ้มครั้งล่าสุด, วันและเวลาที่ทำการเปลี่ยนแปลงข้อมูลในแฟ้ม และตัวชี้ไปยังไดเรกทอรีพ่อ

ทรัสตีจะประกอบด้วยการให้สิทธิ์ของทรัสตีต่าง ๆ, ตัวชี้ไปยังไดเรกทอรีพ่อ, และตัวชี้ไปยังทรัสตีถัดไป

5.1.2.3 ตารางไดเรกทอรี แฮนเดิล (Directory Handle Table)

เครื่องบริการแฟ้มจะทำการเก็บรักษาตารางไดเรกทอรี แฮนเดิลของแต่ละสถานีงานที่ทำการลงบนทึกเข้า (logged in) ไปในระบบ ตารางนี้จะถูกใช้งานเมื่อมีการจองไดเรกทอรี แฮนเดิล เมื่อมีการจองแฮนเดิล, จะมีการเก็บตัวชี้สองตัวไว้ในตาราง ตัวชี้ตัวแรกจะทำการชี้ไปยังตารางเก็บหมวด ส่วนตัวชี้อีกตัวจะชี้ไปยังไดเรกทอรีในตารางไดเรกทอรี

5.1.2.4 ตารางตัวขับ (Drive Tables)

เซลล์จะทำการเก็บรักษาตารางสามตารางเพื่อใช้จัดการการทำงานของตัวขับ แต่ละตารางจะประกอบด้วยข้อมูล 32 ตัว ซึ่งได้แก่ตัวขับถาวร A...Z และตัวขับชั่วคราว "[", "\", "]", "^", "_", และ "`" ตารางทั้งสามตารางได้แก่

1. ตารางเก็บแฮนเดิลของตัวขับ (Drive Handle Table)

ตารางนี้จะชี้ไปยังตารางไดเรกทอรีแฮนเดิล เพื่อแสดงว่าไดเรกทอรีถูกแมปไปยังตัวขับใด

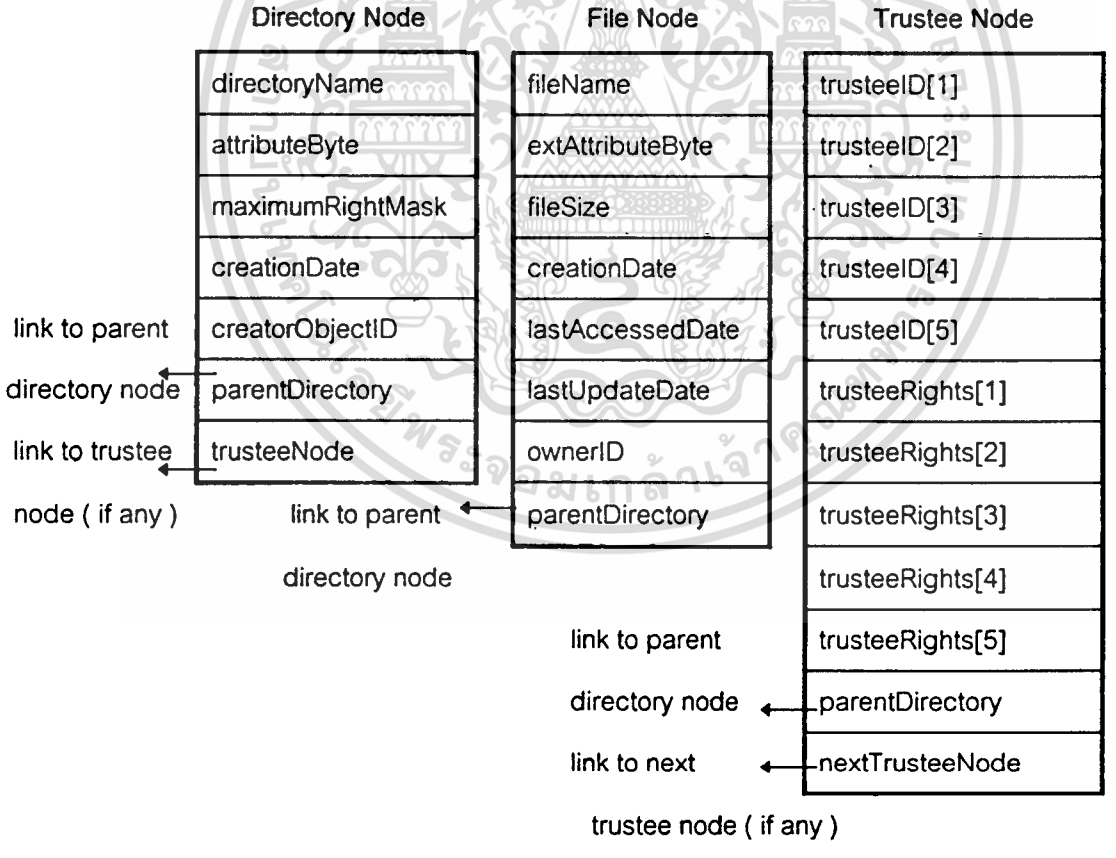
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ตารางเก็บแฟล็กของตัวขับ

ตารางนี้จะเก็บบิตที่แสดงถึงการจัดสรรการใช้งานของตัวขับ ว่าเป็นตัวขับเฉพาะบริเวณ (local drive), ตัวขับในระบบเครือข่าย, ตัวขับถาวรหรือตัวขับชั่วคราว

3. ตารางเก็บคอนเน็คชันไอดีของตัวขับ (Drive Connection ID Table)

ตารางนี้จะชี้ไปยังตารางที่เก็บชื่อเครื่องบริการ (Server Name Table) และตารางที่เก็บข้อมูลเกี่ยวกับการเชื่อมต่อ (Connection Information Table) เพื่อแสดงว่าตัวขับแมปไปยังเครื่องบริการเครื่องใด ถ้าตัวขับเป็นตัวขับโดยปริยาย (default drive) ตัวขับจะชี้ไปยังเครื่องบริการโดยปริยาย (default server)



รูปที่ 5.2 ตารางไดเรกทอรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2.5 ตารางหมวด (Volume Table)

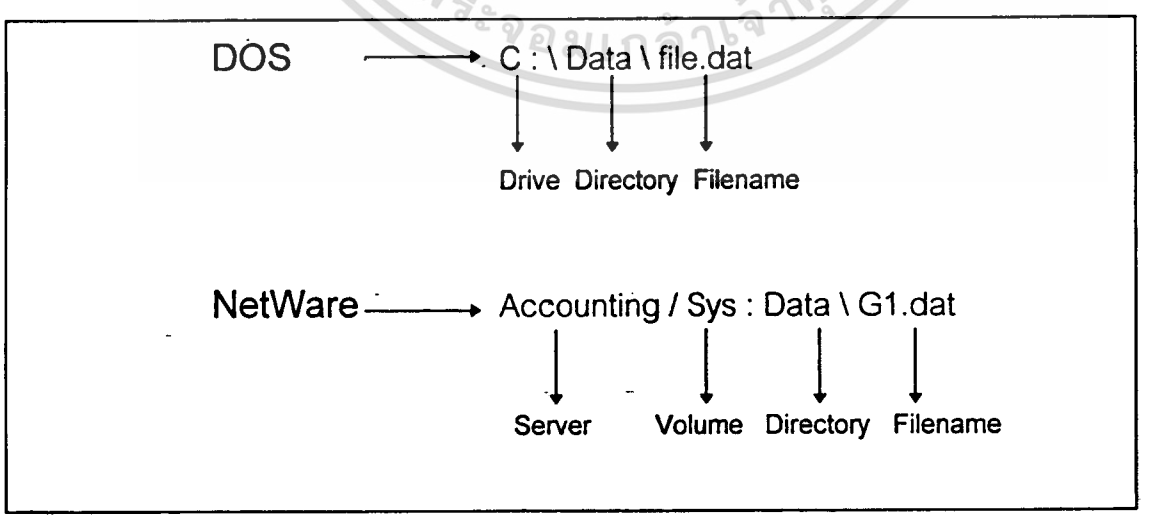
เครื่องบริการแฟ้มจะทำการเก็บหมวดไว้ภายในตารางหมวด ภายในตารางจะประกอบด้วยชื่อหมวดแต่ละหมวด, จำนวนบล็อกที่มากที่สุดและบล็อกที่ใช้งานได้ภายในจานบันทึก (disk), เซกเตอร์ต่อบล็อก (sector per block), จำนวนไดเรกทอรีสล็อต (directory slot) ทั้งหมดและไดเรกทอรีสล็อตที่ใช้งานได้ทั้งหมด, และแฟล็กที่ใช้แสดงว่าหมวดสามารถถอด (removable) ได้หรือไม่

5.1.2.6 การจัดการหมวด

หมวดคือการแบ่งเนื้อที่บนจานบันทึกให้แบ่งออกเป็นส่วน ๆ ผู้ใช้สามารถเรียกใช้หมวดโดยการแมปเข้ากับตัวขับโดยใช้คำสั่ง MAP เช่น MAP F:=SYS:APPS\WP คือการแมปหมวด SYS ไดเรกทอรี APPS\WP เข้ากับตัวขับ F:

หมวดแรกสุด (หมวด 0) บนเครื่องบริการแฟ้มของเน็ตแวร์จะถูกเรียกว่า SYS จะเป็นหมวดที่ใช้ในการบูตระบบ เมื่อต้องการใช้งาน เมื่อมีการแมปหมวดเข้ากับตัวขับ เชลล์จะทำการจองไดเรกทอรี แฮนเดิล จากนั้นจึงทำการเก็บไดเรกทอรี แฮนเดิลไว้ภายในตารางเก็บแฮนเดิลของตัวขับตามออฟเซ็ตหมายเลขตัวขับ

ตารางเก็บแฟล็กของตัวขับปรับทันกาล (update) ตามการจองแฮนเดิล ตัวขับถาวรในระบบเครือข่าย, และตารางเก็บคอนเน็คชันไอดีของตัวขับจะชี้ไปยังเครื่องบริการของหมวดนั้น



รูปที่ 5.3 แสดงการเรียกใช้แฟ้มในระบบดอสและเน็ตแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2.7 การอ้างถึงแฟ้มในเน็ตแวร์

การเรียกใช้แฟ้มในระบบเน็ตแวร์จะต้องมีการอ้างถึงชื่อแฟ้มหรือไดเรกทอรีที่ใช้เก็บพร้อมกับชื่อเต็มของแฟ้มเหล่านั้น

ในระบบการทำงานแบบดอส ตัวขับจานบันทึก (disk drives) จะมีความสำคัญสูงสุดในบรรดาอุปกรณ์หน่วยเก็บข้อมูล (storage device) แต่ในเน็ตแวร์ ตัวขับจะมีลักษณะแบบพลวัต (dynamic) ซึ่งอาจจะแมปไปยังเครื่องบริการแฟ้ม, หมวด, หรือไดเรกทอรีใดก็ได้

ในระบบเน็ตแวร์ เครื่องบริการแฟ้มจะประกอบด้วยหมวดซึ่งจะประกอบด้วยหมวด ภายในหมวดจะประกอบด้วยไดเรกทอรี ตัวขับจะแทนไดเรกทอรีของเครื่องบริการแฟ้มในเชิงตรรก (logical) แทนที่จะแสดงการแบ่งเนื้อที่บนจานบันทึกทางกายภาพ (physical) ดังเช่นการทำงานในดอส การทำเช่นนี้ทำให้สามารถใช้งานเครื่องบริการแฟ้มได้ โดยจะสามารถทำการเปลี่ยนหรือลบหมวดออกจากตัวขับได้ตามที่ต้องการ

ความแตกต่างของการเรียกใช้แฟ้มในระบบดอสและเน็ตแวร์จะแสดงดังรูปที่ 5.3 ซึ่งในดอสจะมีการเรียกใช้ file.dat จากไดเรกทอรี Data จากตัวขับ C: ส่วนในเน็ตแวร์ เครื่องบริการแฟ้ม Accounting จะเรียกใช้แฟ้ม G1.dat จากหมวด SYS ไดเรกทอรี Data

5.1.2.8 การเขียนโปรแกรมเมื่อมีเครื่องบริการมากกว่าหนึ่งเครื่อง

1. ถ้าฟังก์ชันที่ใช้งานมีหมายเลขคอนเน็คชันไอดี ฟังก์ชันก์ถูกส่งไปทำงานที่เครื่อง บริการตามหมายเลขคอนเน็คชันไอดีนั้น
2. ถ้ามีการใช้ SetPreferredConnectionID ฟังก์ชันจะถูกส่งไปทำงานที่เครื่องบริการที่ถูกเซ็ต
3. ถ้าไม่มีการเซ็ตหมายเลขคอนเน็คชันไอดี ฟังก์ชันจะถูกส่งไปทำงานที่เครื่องบริการแฟ้มที่ตัวขับติดตั้งอยู่
4. ถ้าตัวขับไม่ได้ติดต่อกับระบบเครือข่าย ฟังก์ชันจะถูกส่งไปทำงานที่เครื่องบริการแฟ้มที่ทำการลงบันทึกเข้า (หรือเปลี่ยนเครื่องบริการแฟ้มโดยใช้คำสั่ง SetPrimaryServer)
5. ถ้าสถานีงานขาดการติดต่อกับเครื่องบริการที่ทำการลงบันทึกเข้า ฟังก์ชันจะถูกส่งไปทำงานที่เครื่องบริการเครื่องแรกที่อยู่ในตารางที่ใช้เก็บชื่อเครื่องบริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2.9 คอนเน็คชันไอดี

คอนเน็คชันไอดีเป็นหมายเลขที่แสดงถึงสถานีงานที่ทำการติดต่อ (attach) เข้ากับระบบ เครือข่าย โดยจำนวนเครื่องบริการที่ทำการติดต่อกับระบบเครือข่ายจะจำกัดสูงสุดถึง 8 สถานีงาน

ส่วนหมายเลขคอนเน็คชัน (connection number) จะมีความแตกต่างจากคอนเน็คชันไอดีคือเป็นหมายเลขเฉพาะที่แสดงถึงเครื่องบริการแฟ้มแต่ละเครื่อง

ดังนั้นในการเขียนโปรแกรมจึงต้องระวังความสับสนในตัวเลขทั้งสองมิฉนั้นโปรแกรมที่ได้อาจมีความผิดพลาดเกิดขึ้น

5.1.3 ชนิดของตัวแปรที่ใช้งานในการเขียนโปรแกรมแบบการขัดจังหวะ

- BYTE unsigned char
- WORD unsigned int
- LONG unsigned long

5.1.4 คำสั่งที่ใช้ในการเขียนโปรแกรม

1. GetDefaultConnectionID

เป็นฟังก์ชันที่ใช้ในการหาคอนเน็คชันไอดีของเครื่องบริการแฟ้ม

ใช้ NWAPI LIBRARY	ใช้การขัดจังหวะของดอส (21h)
#include <nit.h>	WORD GetDefaultConnectionID(void)
WORD GetDefaultConnectionID(void)	รีจิสเตอร์เข้า :
เอาต์พุต : หมายเลขคอนเน็คชันไอดี	AH F0h
	AL 02h
	รีจิสเตอร์ออก :
	AL คอนเน็คชันไอดี (1 ถึง 8)

2. GetFileServerName

เป็นฟังก์ชันที่ใช้ในการหาชื่อเครื่องบริการแฟ้ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ NWAPI LIBRARY	ใช้การขัดจังหวะของดอส (21h)
#include <nit.h> void GetFileServerName (WORD connectionID, char *fileServerName) อินพุต :คอนเน็คชันไอดี เอาต์พุต : ชื่อเครื่องบริการแฟ้ม	void GetFileServerName (WORD connectionID, char *fileServerName) รีจิสเตอร์เข้า : AH EFh AL 04h รีจิสเตอร์ออก : ES:SI ชี้ไปยังเซลล์ของตารางที่ เก็บชื่อเครื่องบริการแฟ้ม

3. GetNumberOfLocalDrives

ใช้หาจำนวนตัวขับเชิงตรรก (local drive) ของสถานีงาน

ใช้ NWAPI LIBRARY	ใช้การขัดจังหวะของดอส (21h)
#include <nit.h> int GetNumberOfLocalDrive(void) เอาต์พุต :จำนวนตัวขับของเครื่อง (local drive)	int GetNumberOfLocalDrive(void) รีจิสเตอร์เข้า : AL DBh รีจิสเตอร์ออก : AL จำนวนตัวขับของเครื่อง (local drive)

4. IsSearchDrive

ใช้เพื่อหาว่าตัวขับอยู่ในเส้นทางที่ค้นหา (search path) หรือไม่

ใช้ NWAPI LIBRARY
#include <nit.h> int IsSearchDrive (char driveletter) เอาต์พุต : 1 : เมื่ออยู่ในเส้นทางที่ค้นหา 0 : เมื่อไม่อยู่ในเส้นทางที่ค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. IsConnectionIDInUse

ใช้ในการตรวจสอบเครื่องบริการมีการติดต่อ (attach) กับเครื่องบริการแฟ้มเครื่องอื่นหรือไม่

ใช้ NWAPI LIBRARY
#include <nit.h> int IsConnectionIDInUse (WORD connectionID) อินพุต :คอนเน็คชันไอดี เอาต์พุต : 0 : เครื่องบริการไม่มีการติดต่อถึงกัน 1 : มีการติดต่อ (attach) ถึงกัน

6. SetPrefferedConnectionID

ใช้เพื่อการเซ็ตคอนเน็คชันไอดีไปยังเครื่องบริการที่ต้องการ

ใช้ NWAPI LIBRARY	ใช้การขัดจังหวะของดอส (21h)
#include <nit.h> void SetPrefferedConnectionID (WORD connectionID) อินพุต :คอนเน็คชันไอดี	void SetPrefferedConnectionID(BYTE connectionID) รีจิสเตอร์เข้า : AH F0h AL 00h DL คอนเน็คชันไอดีที่ต้องการ(1 ถึง 8) หรือ 0 ถ้าไม่ต้องการเซ็ต รีเทิร์น : AL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. GetVolumeName

หาชื่อหมวดที่ต้องการ

ใช้ NWAPI LIBRARY	ใช้การขัดจังหวะของดอส (21h)
#include <nit.h> void GetVolumeName (int volumeNumber, char *volumeName) **อินพุต** : หมายเลขของสล็อตหมวด (volume slot) **เอาต์พุต** : ชื่อหมวด **ค่าที่รีเทิร์น** 0 : สำเร็จ 152 : ไม่มีชื่อหมวดที่สล็อตนั้น	int GetVolumeName(int volumeNumber, char *volumeName) **รีจิสเตอร์เข้า :** AH E2h DS:SI Request Buffer Address ES:DI Reply Buffer Address **รีจิสเตอร์ออก :** AL 0 : สำเร็จ 152 : ไม่มีชื่อหมวดที่สล็อตนั้น

8. GetDirectoryHandle

หาแฮนเดิลของตัวขับ

ใช้ NWAPI LIBRARY	ใช้การขัดจังหวะของดอส (21h)
#include <nit.h> BYTE GetDirectoryHandle (BYTE driveNumber) **อินพุต** : หมายเลขของตัวขับ (A = 0, B = 1,...) **เอาต์พุต** : ไดเรกทอรีแฮนเดิล	BYTE GetDirectoryHandle (BYTE driveNumber) **รีจิสเตอร์เข้า :** AH E9h AL 00h DX หมายเลขตัวขับ **รีจิสเตอร์ออก :** AL ไดเรกทอรีแฮนเดิล AH แฟล็กสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. GetDirInfo

แสดงขอมูลของไดเรกทอรี

ใช้ NWAPI LIBRARY
#include <ntt.h> WORD GetDirInfo (WORD serverConnID, BYTE dirHandle, NWDIR_INFO *dirInfo) **อินพุต** : คอนเน็คชันไอดี ของเครื่องบริการ dirHandle แสดงดรรชนี (index) ของ หมวดหรือไดเรกทอรี **เอาต์พุต** : dirInfo ประกอบด้วยข้อมูลต่าง ๆ ของ ไดเรกทอรี

10. GetFullPath

แสดงเส้นทางทั้งหมดของตัวขับ

ใช้ NWAPI LIBRARY
#include <nit.h> int GetFullPath (char drive, char *path) **อินพุต** : หมายเลขตัวขับ **เอาต์พุต** : เส้นทาง (path) ของตัวขับ **ค่าที่รีเทิร์น** 0 : สำเร็จ 255 : มีความผิดพลาดอันเนื่องมา จากแฮนเดิล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. MapDrive

คำสั่งใช้ในการแมปตัวขับ

ใช้ NWAPI LIBRARY
#include <nit.h> int MapDrive (WORD connectionID, Byte baseDriveNumber, char *directoryPath, BYTE searchFlag, int searchOrder, char *driveletter) **อินพุต** :คอนเน็คชันไอดี baseDriveNumber directoryPath searchFlag [DRIVE_ADD, DRIVE_ DELETE,DRIVE_INSERT] searchOrder [0 : เมื่อไม่ใช่ search drive, 1-16 ลำดับของการ ค้นหา] driveletter [A-Z] **เอาต์พุต** : 0 : สำเร็จ เลขอื่น : แสดงความผิดพลาดที่เกิดขึ้น

การที่การขัดจังหวะของดอสไม่มีคำสั่งบางคำสั่ง เช่น MapDrive ไม่ได้หมายความว่าไม่สามารถทำการแมปตัวขับได้ แต่สามารถทำได้โดยใช้คำสั่งอื่น เช่นโปรแกรม MapIt ดังแสดงในรูป 5.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (( argc < 2 ) || ( argc > 3 ))
    {
        printf("Usage : MapIt drive [-d%\]\n");
        printf("Maps SYS:PUBLIC to drive specified.  Use -d to remove mapping.\n");
        exit(1);
    }
    if (strcmp ("-d",argv[2]))
    {
        /* Map the drive */
        result = AllocatePermanentDirHandle( argv[1] );
        if ( result )
        {
            printf("Error trying to allocate diretory, Error #%X\n",result );
            exit(1);
        }
        else
            printf("Drive %s has been mapped to SYS:PUBLIC\n",argv[1]);
    }
    else
    {
        /* Remove the mapping */
        dirHandle = GetDirectoryHandle((WORD) (toupper(*argv[1])-'A'));
        result = DeallocatePermanentDirHandle( dirHandle );
        if ( result )
        {
            printf("Error trying to de-allocate diretory, Error #%X\n",result );
            exit(1);
        }
        else
            printf("Drive %s has had its drive mapping deleted\n",argv[1]);
    }
}
```

รูปที่ 5.4 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************************************************************

        MapIt : Maps a user-specified drive to the SYS:PUBLIC directory

        BY      : CHARLES ROSE

*****************************************************************************/


#include <dos.h>

#include <stdio.h>


typedef unsign char BYTE;

typedef unsign int WORD;

typedef unsign long LONG;


/* Reply Structure */

struct allocPermDirHandleReq {

        WORD    Length;

        BYTE    Function;

        BYTE    DirHandle;

        BYTE    Drive;

        BYTE    DirPathLength;

        BYTE    DirPath[255];

} Request;


int     AllocatePermanentDirHandle( char * Drive );

int     DeallocatePermanentDirHandle( BYTE DirHandle );

BYTE    GetDirectoryHandle( WORD driveNmber );


main( int argc, char *argv[] )

{

        int     result;

        BYTE    dirHandle;
```

รูปที่ 5.4 ตัวอย่างการเขียนโปรแกรมควบคุมเน็ตแวร์โดยใช้การขัดจังหวะของดอส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (( argc < 2 ) || ( argc > 3 ))
    {
        printf("Usage : MapIt drive [-d%\]\n");
        printf("Maps SYS:PUBLIC to drive specified.  Use -d to remove mapping.\n");
        exit(1);
    }
    if (strcmp ("-d",argv[2]))
    {
        /* Map the drive */
        result = AllocatePermanentDirHandle( argv[1] );
        if ( result )
        {
            printf("Error trying to allocate diretory, Error #%X\n",result );
            exit(1);
        }
        else
            printf("Drive %s has been mapped to SYS:PUBLIC\n",argv[1]);
    }
    else
    {
        /* Remove the mapping */
        dirHandle = GetDirectoryHandle((WORD) (toupper(*argv[1])-'A'));
        result = DeallocatePermanentDirHandle( dirHandle );
        if ( result )
        {
            printf("Error trying to de-allocate diretory, Error #%X\n",result );
            exit(1);
        }
        else
            printf("Drive %s has had its drive mapping deleted\n",argv[1]);
    }
}
```

รูปที่ 5.4 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int AllocatePermanentDirHandle( char *Drive )
{
        BYTE    Reply[4] = {2,0,0,0};

        strcpy( Request.DirPath, "SYS:PUBLIC");

        Request.Length = sizeof( struct allocPermDirHandleReq ) - 2;
        Request.Function = 0x12;
        Request.DirHandle = 0;
        Request.Drive = toupper( *Drive ); /* Make it uppercase */
        Request.DirPathLength = strlen( Request.DirPath );

        _AH = 0xE2;
        _SI = ( unsigned )&Request;
        _DI = ( unsigned )Reply;
        _ES = _DS;

        getinterrupt( 0x21 );
        return _AL;
}

int DeallocatePermanentDirHandle( BYTE dirHandle )
{
        BYTE    Reply[2] = {0,0};

        strcpy( Request.DirPath, "SYS:PUBLIC");

        Request.Length = sizeof( struct allocPermDirHandleReq ) - 2;
        Request.Function = 0x14;
        Request.DirHandle = dirHandle;

        _AH = 0xE2;
        _SI = ( unsigned )&Request;
        _DI = ( unsigned )Reply;
        _ES = _DS;

        getinterrupt( 0x21 );
        return _AL;
}
```

รูปที่ 5.4 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BYTE GetDirectoryHandle( WORD driveNumber )
{
        _AH = 0xE9;
        _AL = 0;
        _DX = driveNumber;

        getinterrupt( 0x21 );
        return _AL;
}
```

รูปที่ 5.4 (ต่อ)

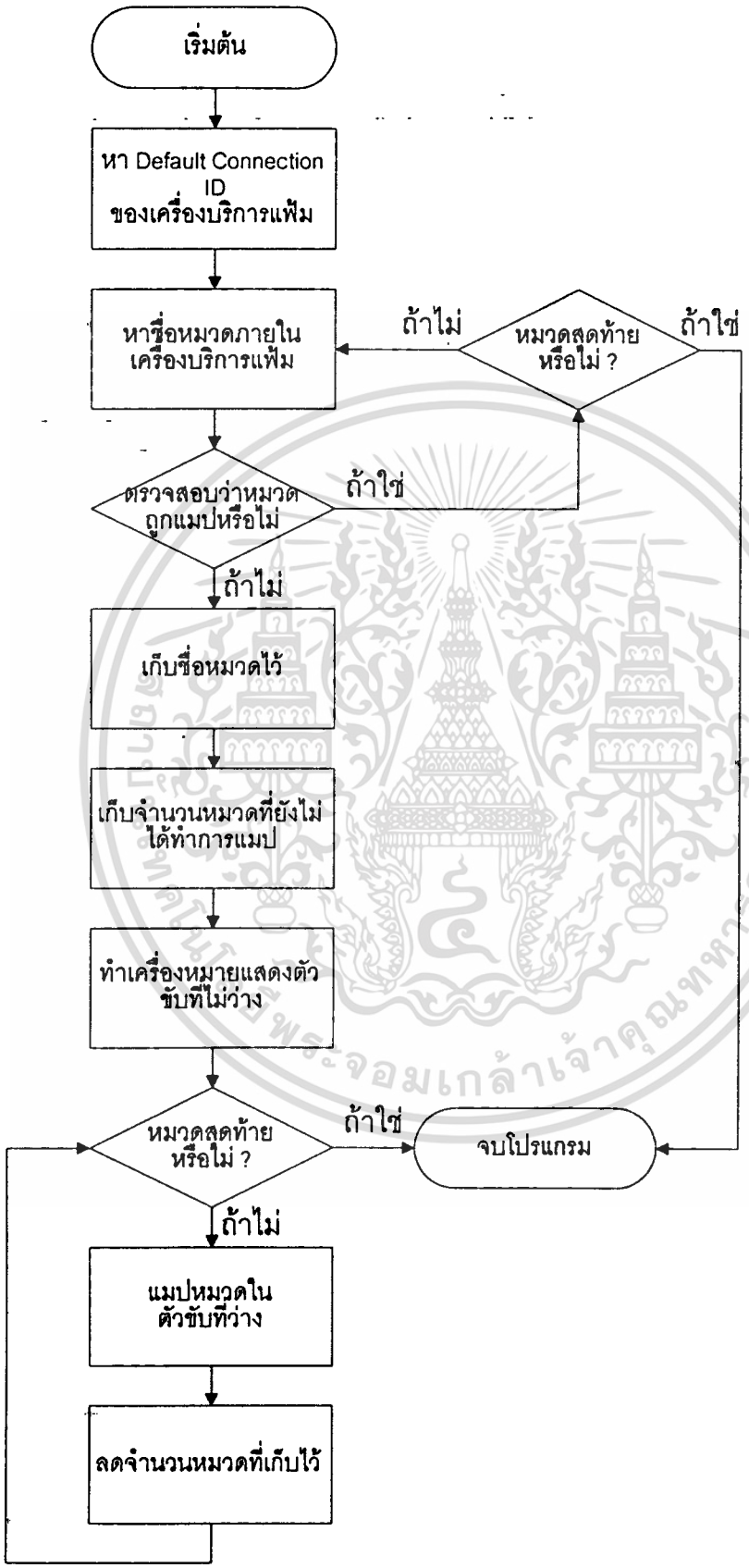
5.1.5 ขั้นตอนในการเขียนโปรแกรม

ในการเขียนโปรแกรมเพื่อการแมปหมวด ได้เลือกใช้ไลบรารีที่เชื่อมโยงการทำงานกับเน็ตแวร์ (NWAPI Library) ทำงานร่วมกับโปรแกรมบอร์แลนด์ซี ขั้นตอนในการทำงานมีดังนี้

1. หาคอนเน็คชันไอดีของเครื่องบริการแฟ้ม
2. ทำการตรวจสอบตัวขับว่ามีการใช้งานหรือไม่ ถ้ามีการใช้งาน, ให้ทำเครื่องหมายไว้ ถ้ายังไม่มีการใช้งาน, ให้เก็บไว้สำหรับการแมป
3. การค้นหาชื่อหมวดภายในเครื่องบริการแฟ้ม
4. ถ้าหมวดที่ค้นหาถูกแมปไว้เรียบร้อยแล้ว ค้นหาหมวดใหม่
5. ถ้าหมวดยังไม่ได้ทำการแมป ให้เก็บไว้ในตัวแปร
6. เมื่อทำการค้นหาหมวดเรียบร้อยแล้ว ให้ทำการแมปหมวดเหล่านั้น

ลำดับแผนภูมิสายงาน (flow chart) สามารถแสดงได้ดังรูปที่ 5.5 สำหรับโปรแกรมการทำงานจะแสดงไว้ในภาคผนวก ก.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 แผนภูมิสายงานการเขียนโปรแกรมแมปหมวด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การติดตั้งโปรแกรมค้นหาฐานข้อมูลลงบนระบบเน็ตแวร์

ในการวิจัย ได้ทดลองทำการทดสอบการทำงานกับโปรแกรมค้นหาฐานข้อมูล โปรแกรมที่ใช้ในการทดสอบ คือ โปรแกรม PROQUEST ซึ่งเป็นโปรแกรมที่ใช้ค้นหาฐานข้อมูลซีดีรอมในห้องสมุดกลางขณะนี้

ในการติดตั้งจะต้องทำตามขั้นตอนดังนี้

- เมาท์แผ่นซีดีรอมเข้ากับเครื่องบริการ แฟ้ม
- นำโปรแกรมค้นหาฐานข้อมูลมาลงที่ตัวขับบนเน็ตแวร์ ที่ผู้ใช้สามารถเรียกใช้ได้ (จะเป็น drive G หรือ drive F ก็ได้)

การเรียกใช้งาน

- แมปหมวดของแผ่นซีดีรอมให้เป็นตัวขับบนเน็ตแวร์
- กำหนดไดเรกทอรีที่ใช้ในการเก็บข้อมูลชั่วคราว ใช้คำสั่งดังนี้

G:\>SET FULTEMP=G:\TEMP

(หมายเหตุ : ถ้าไม่มีการกำหนดไดเรกทอรีที่ใช้เก็บ ข้อมูลชั่วคราว เวลาใช้งานโปรแกรมค้นหาฐานข้อมูลจะฟ้องข้อผิดพลาดว่า Search Engine Error เพราะว่าไม่สามารถเก็บข้อมูลชั่วคราวได้)

สำหรับโปรแกรมที่ได้พัฒนาขึ้นนั้นผู้ใช้สามารถเรียกใช้งานได้ทันที หลังจากที่ติดตั้งระบบซีดีรอมลงบนเน็ตแวร์แล้ว เพราะโปรแกรมจะกำหนดค่าต่างๆ ให้ โดยผู้ใช้ไม่ต้องกำหนดเอง

5.3 เทคนิคในการเขียนโปรแกรมเบี่ยงเบนทิศทาง

จุดประสงค์ของการเขียนโปรแกรมเพื่อเบี่ยงเบนทิศทางในวิทยานิพนธ์นี้คือ เพื่อใช้ในการพัฒนาระบบเครือข่ายฐานข้อมูลซีดีรอมนี้ต่อไปในอนาคต โปรแกรมเบี่ยงเบนทิศทางนี้สามารถใช้กับจอภาพแบบ VGA ที่มีโหมดการทำงาน คือ โหมด 3 หรือ โหมดการทำงานที่สามารถทำงานแทนกันได้

5.3.1 หลักในการเขียนโปรแกรม

โปรแกรมเบี่ยงเบนทิศทางนี้จะรับข้อมูลเข้าจากแฟ้มข้อมูลเข้า ซึ่งจะระบุถึงข้อมูลที่ต้องการจะค้นหา ฐานข้อมูลที่ต้องการจะค้นหา ลักษณะของผลลัพธ์ที่ต้องการ และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งของแฟ้มข้อมูลออก ซึ่งในการทำงานหลักของโปรแกรมจะใช้วิธีการดึงข้อมูลออกจากซีดีรอม เพื่อให้ได้ผลลัพธ์ตามที่ได้กำหนดไว้ในแฟ้มข้อมูลเข้า โดยจะออกคำสั่งให้โปรแกรมที่ใช้ในการดึงข้อมูลทำงานปรกติเหมือนได้รับคำสั่งจากผู้ใช้โดยตรง ซึ่งโปรแกรมเบี่ยงเบนทิศทางนี้จะทำหน้าที่ในการดูแลการทำงานของซีดีรอมว่ามีการทำงานถึงขั้นตอนใด และต้องการรับข้อมูลใดจากแป้นพิมพ์เพื่อที่จะทำงานได้ผลลัพธ์ออกมาตรงตามที่กำหนดไว้ในแฟ้มข้อมูลเข้า การดึงข้อมูลจะดึงข้อมูลจากหน่วยความจำของจอภาพเพื่อตรวจสอบว่าขณะนี้โปรแกรมกำลังงานอะไรอยู่ และเมื่อโปรแกรมทำงานมาถึงลำดับที่ต้องการแล้ว ก็จะป้อนข้อมูลให้กับโปรแกรมโดยใส่ข้อมูลเหล่านั้นลงในบัฟเฟอร์ของแป้นพิมพ์ การทำงานจะมีลักษณะวนซ้ำเช่นนี้ไปเรื่อย ๆ จนกว่าจะจบการทำงานของโปรแกรม

เมื่อเริ่มต้นทำงาน ภายหลังจากที่ได้ตรวจสอบแฟ้มข้อมูลเข้าแล้ว โปรแกรมเบี่ยงเบนทิศทางจะเริ่มการทำงานของโปรแกรมค้นหาฐานข้อมูล ซึ่งการแทรกการทำงานของโปรแกรมเบี่ยงเบนทิศทางในระหว่างที่โปรแกรมค้นหาของมูลกำลังทำงานอยู่ จะทำได้โดยการใช้การขัดจังหวะของสัญญาณนาฬิกา โดยทุกครั้งที่มีการขัดจังหวะของสัญญาณนาฬิกาเกิดขึ้น การทำงานจะเปลี่ยนไปสู่การทำงานในส่วนของโปรแกรมเบี่ยงเบนทิศทางเพื่อควบคุมการทำงานของโปรแกรมค้นหาข้อมูล

5.3.2 หลักในการจัดลำดับขั้นตอนการทำงาน

โปรแกรมเบี่ยงเบนทิศทางจะกำหนดลำดับขั้นตอนการทำงานของโปรแกรมค้นหาฐานข้อมูลภายหลังจากที่ได้ตรวจสอบแฟ้มข้อมูลเข้าแล้ว ซึ่งลำดับขั้นตอนการทำงานจะถูกกำหนดไว้ในแฟ้มชื่อ scr_opt.dat โปรแกรมเบี่ยงเบนทิศทางจะสั่งให้โปรแกรมค้นหาฐานข้อมูลทำงานตามขั้นตอนนี้จนกว่าจะสิ้นสุดขั้นตอนการทำงานซึ่งจะได้ผลลัพธ์ตามที่ต้องการเก็บไว้ในแฟ้ม ~temp.$$$ และจะนำแฟ้มนี้มาจัดรูปแบบอีกครั้งเพื่อให้ได้ผลลัพธ์ตามแฟ้มข้อมูลเข้า ซึ่งผลลัพธ์ที่ได้นี้จะเก็บไว้ในแฟ้มข้อมูลออก

ข้อมูลในแฟ้ม scr_opt.dat **จะประกอบด้วย 7 ส่วน คือ**

1. **แถวเริ่มต้นของสายอักขระ** (string)
2. **คอลัมน์เริ่มต้นของสายอักขระ**
3. **แถวสิ้นสุดของสายอักขระ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. คอลัมน์สิ้นสุดของสายอักขระ

5 . สายอักขระที่ปรากฏบนหน้าจอ

6. ค่า ลักษณะเฉพาะ ของสายอักขระ

7. สิ่งที่ต้องการให้ทำเมื่อสายอักขระนั้นปรากฏบนหน้าจอแล้ว ประกอบด้วย

- สัญลักษณ์ '@' -> รหัสที่ตามมาจะเป็นค่า Scan Code ไม่ใช่ ASCII Code

เช่น @F1 ('F1' จะเป็นรหัสของปุ่ม F1)

- รหัสแอสกี 241 -> หน่วงเวลา โดยจะไม่ประมวลผลอะไรในช่วงเวลานี้
- รหัสแอสกี 133 -> บันทึกข้อมูลจากหน้าจอลงแฟ้ม ใช้ในกรณีที่ระดับ ของผลลัพธ์ เป็น 1 (เฉพาะจำนวนข้อมูลที่พบ)
- รหัสแอสกี 239 -> กำหนดจำนวนข้อมูลที่ต้องการบันทึกลงแฟ้มในแต่ละครั้ง (กำหนดไว้ครั้งละ 10)
- รหัสแอสกี 240 -> ตรวจสอบว่าผลลัพธ์ของการค้นหาเป็น 0 หรือไม่
- รหัสแอสกีอื่นๆ -> ข้อมูลที่ต้องการใส่ลงในบัฟเฟอร์ของแป้นพิมพ์

โดยข้อมูลแต่ละส่วนจะคั่นด้วยเครื่องหมาย '|' ตัวอย่างเช่น

15|2|25|2|Search data|15|@ENTER หมายถึง ถ้าพบสายอักขระ "Search data " ซึ่งมีค่าลักษณะเฉพาะเป็น15 บนหน้าจอซึ่งเริ่มต้นที่ตำแหน่งที่ 15,2 และสิ้นสุดที่ตำแหน่งที่ 20,2 ให้กดปุ่ม ENTER (ใส่ค่ารหัสของปุ่ม ENTER ลงในบัฟเฟอร์ของแป้นพิมพ์) แต่ถ้าไม่พบสายอักขระที่ต้องการในตำแหน่งนั้นให้รอจนกว่าจะพบ

25|3|29|3|Input|31|cd-rom networking หมายถึง ถ้าพบสายอักขระ "Input " ซึ่งมีค่า ลักษณะเฉพาะเป็น 31 บนหน้าจอซึ่งเริ่มต้นที่ตำแหน่งที่ 25,3 และสิ้นสุดที่ตำแหน่งที่ 29,3 ให้ใส่สายอักขระ "cd-rom networking" ลงในบัฟเฟอร์ของแป้นพิมพ์ แต่ถ้าไม่พบสายอักขระที่ต้องการในตำแหน่งนั้นให้รอจนกว่าจะพบ

ข้อมูลเหล่านี้จะถูกอ่านจากแฟ้มและเก็บไว้ในหน่วยความจำ ในขณะที่โปรแกรมกำลังประมวลผลเพื่อความรวดเร็วในการทำงาน

ลักษณะของลำดับขั้นในการทำงานจะถูกกำหนดโดยลักษณะของผลลัพธ์ซึ่งจะมีลักษณะที่ตายตัวในแต่ละรูปแบบ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

keyword = search

database = dao

ลักษณะของผลลัพธ์แบบที่ 1 จะมีลำดับการทำงานดังนี้

10|8|38|8|DISSERTATION ABSTRACTS ONDISC|15|@DOWN

10|9|15|9|INSPEC|15|@ENTER

34|4|45|4|Search Entry|31|search$$ ($$ หมายถึง ENTER)

3|5|8|5|search|31|๐

3|5|8|5|search|31|@ENTER

78|5|78|5|□|31|๑

78|5|78|5|□|31|@ESC

34|4|45|4|Search Entry|31|@SHF+F10

29|17|31|17|Yes|15|@ENTER

ลักษณะของผลลัพธ์แบบที่ 2 จะมีลำดับการทำงานดังนี้

10|8|38|8|DISSERTATION ABSTRACTS ONDISC|15|@DOWN

10|9|15|9|INSPEC|15|@ENTER

34|4|45|4|Search Entry|31|search$$

3|5|8|5|search|31|๐

3|5|8|5|search|31|@ENTER

78|6|78|6| |31|@F4

24|16|28|16|START|15|@UP

11|13|20|13|File name:|63|๑

11|13|20|13|File name:|63|@UP

24|9|27|9|Long|15|๑

24|9|27|9|Long|15|@UP

24|8|28|8|Short|15|

24|8|28|8| Short|15|@ESC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

78|5|78|5|□|31|◉

78|5|78|5|□|31|@ESC

34|4|45|4|Search Entry|31|@SHF+F10

29|17|31|17|Yes|15|@ENTER

ลักษณะของผลลัพธ์แบบที่ 3 จะมีลำดับการทำงานดังนี้

10|8|38|8|DISSERTATION ABSTRACTS ONDISC|15|@DOWN

10|9|15|9|INSPEC|15|@ENTER

34|4|45|4|Search Entry|31|search$$

3|5|8|5|search|31|○

3|5|8|5|search|31|@ENTER

78|5|78|5|□|31|◉

78|5|78|5|□|31|@ESC

34|4|45|4|Search Entry|31|@SHF+F10

29|17|31|17|Yes|15|@ENTER

การที่โปรแกรมเบียงเบนทิศทางทำงานตามแฟ้ม scrptr.dat ทำให้สามารถนำโปรแกรมนี้ไปใช้ควบคุมการทำงานของโปรแกรมอื่นได้ แต่การทำงานเมื่อพบสายอักขระจะใช้ได้ในกรณีสัญลักษณ์ '@' และ รหัสแอสกีอื่น ๆ เท่านั้น เพราะสัญลักษณ์อื่นจะใช้ควบคุมการทำงานพิเศษสำหรับโปรแกรมค้นหาฐานข้อมูลนี้เท่านั้น

5.3.3 หลักในการตรวจสอบสายอักขระบนจอภาพ

การตรวจสอบสายอักขระบนจอภาพจะดึงข้อมูลมาจากหน่วยความจำของจอภาพ และตรวจสอบว่าเป็นสายอักขระที่ต้องการหรือไม่ รวมทั้งค่าลักษณะเฉพาะของสายอักขระนั้นจะต้องตรงตามที่ระบุด้วย ในการคำนวณหาตำแหน่งของหน่วยความจำตามตำแหน่งบนจอภาพที่กำหนดไว้จะคำนวณจาก

Location = ((row - 1) * 80 + (column - 1)) * 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าตำแหน่งที่คำนวณได้จะต้องไปรวมกับตำแหน่งเริ่มต้นของหน่วยความจำของจอภาพ คือ B800:0000 (โหมด 3)

เมื่อพบตำแหน่งที่ต้องการแล้ว ตำแหน่งแรกที่พบจะเป็นตำแหน่งที่เก็บค่าของรหัสแอสกีของตัวอักษรที่แสดงอยู่บนจอภาพ และตำแหน่งถัดไปจะเป็นค่าลักษณะเฉพาะของตัวอักษรนั้น โดยจะมีการคำนวณตำแหน่งของหน่วยความจำ 2 ค่า ค่าแรกจะเป็นตำแหน่งเริ่มต้นของสายอักขระที่ต้องการตรวจลอบ และค่าที่สองจะเป็นตำแหน่งสิ้นสุดของสายอักขระที่ต้องการตรวจสอบ ซึ่งจะตรวจสอบสายอักขระระหว่างค่าทั้งสองนี้

5.3.4 หลักในการใส่ข้อมูลลงในบัฟเฟอร์ของแป้นพิมพ์

การใส่ข้อมูลลงในบัฟเฟอร์ของแป้นพิมพ์จะจำลองการจัดการบัฟเฟอร์ให้เหมือนกับที่ไบออสทำ โดยจะใช้ตัวชี้ 2 ตัว ตัวชี้ตัวแรกจะชี้ไปที่ตำแหน่ง 041A ซึ่งเป็นหน่วยความจำที่เก็บค่าตำแหน่งของบัฟเฟอร์สำหรับการอ่านค่าจากบัฟเฟอร์ในครั้งต่อไป ตัวชี้ตัวที่สอง จะชี้ไปที่ 041C ซึ่งเป็นหน่วยความจำที่เก็บตำแหน่งของบัฟเฟอร์สำหรับใช้ใส่ข้อมูลใหม่ลงในบัฟเฟอร์

การใส่ ข้อมูลจะใส่ลงตามตำแหน่งของตัวชี้ตัวที่สอง ส่วนตัวชี้ตัวแรกจะมีการเปลี่ยนแปลงค่าเองเมื่อโปรแกรมมาอ่านข้อมูลในบัฟเฟอร์ ซึ่งการใส่ข้อมูลลงบัฟเฟอร์จะต้องพิจารณาเงื่อนไขต่อไปนี้

1. ตรวจดูว่าบัฟเฟอร์ว่างหรือไม่ จากค่าของตัวชี้ตัวแรกเท่ากับค่าของตัวชี้ตัวที่สอง ถ้าว่างก็ใส่ข้อมูลลงไปได้

2. เมื่อใส่ข้อมูลลงไปจะเพิ่มค่าของตัวชี้ตัวที่สองเสมอ และตรวจสอบดูว่าตำแหน่งของบัฟเฟอร์ที่ใส่ลงไปนั้นเกิน 043D หรือไม่ ถ้าเกินต้องกลับไปที่ 041E ใหม่ สำหรับตัวชี้ตัวแรกไม่ต้องมีการเปลี่ยนแปลงค่าเพราะจะมีการเปลี่ยนแปลงเองเสมอ

3 . ตรวจดูว่าบัฟเฟอร์เต็มหรือไม่ ถ้าบัฟเฟอร์เต็มต้องรอจนกว่าบัฟเฟอร์ว่างถึงจะใส่ข้อมูลลงไปได้

5.3.5 แฟ้มข้อมูลเข้า

แฟ้มข้อมูลเข้าจะประกอบด้วย

- สายอักขระที่ต้องการค้นหา (ประกอบด้วยหลายๆ สายอักขระใน1 แฟ้มได้ แต่ไม่เกิน 10)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฐานข้อมูลที่ใช้ค้นหา
- ลักษณะของผลลัพธ์ แบ่งเป็น
    - ระดับที่ 1 ประกอบด้วย จำนวนข้อมูลที่ค้นหาได้สำหรับสายอักขระนั้น ๆ
    - ระดับที่ 2 ประกอบด้วย ชนิดของเอกสาร, หัวข้อ, ชื่อผู้แต่ง, สำนักพิมพ์ และข้อความที่สำคัญอื่นๆ
    - ระดับที่ 3 ประกอบด้วย รายละเอียดเหมือนระดับที่ 2 แต่ประกอบด้วยรายละเอียดย่อของหนังสือ
- แฟ้มข้อมูลออก

ตัวอย่างแฟ้มข้อมูลเข้า

```
DATABASE = INSPEC
KEYWORD = CD-ROM NETWORKING
OUTLEVEL = 2
OUTPUT FILE = C:\OUTPUT.TXT
```

ซึ่งลักษณะของผลลัพธ์ที่แตกต่างกันจะได้แฟ้ม scr_opt.dat(สำดับในการทำงาน) แตกต่างกัน (หัวข้อ 5.3.2)

5.3.6 แผนภูมิสายงาน

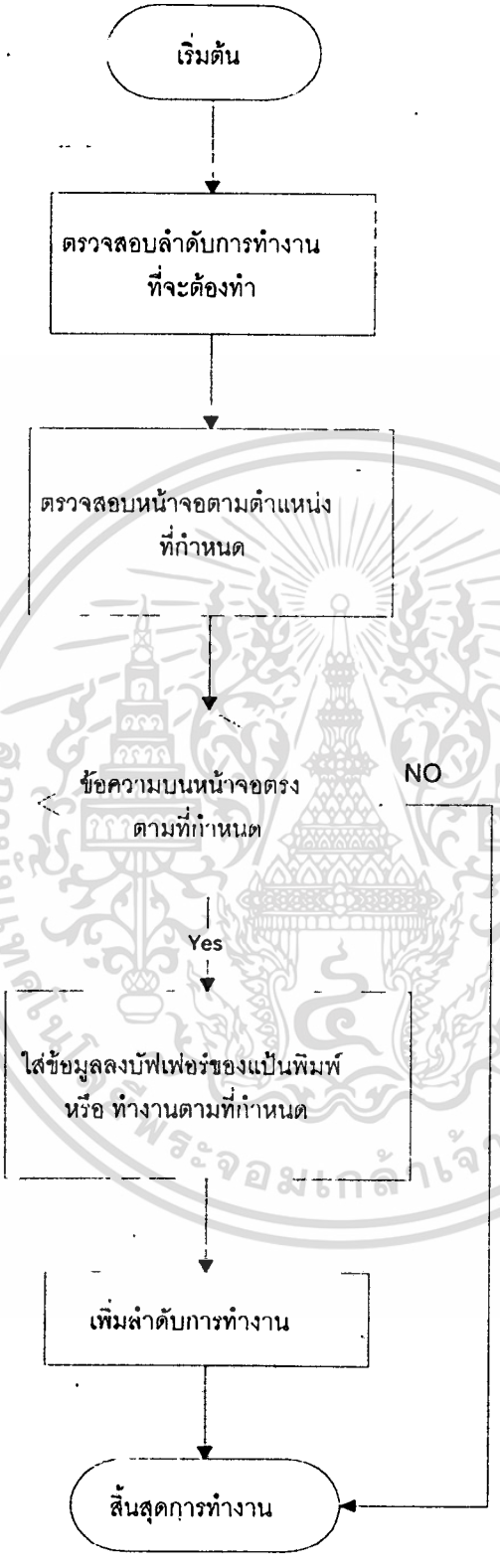
แผนภูมิสายงานแสดงการทำงานของโปรแกรมเบี่ยงเบนทิศทางแสดงในรูปที่ 5.6 และ 5.7 ในรูปที่ 5.6 จะเป็นการทำงานในส่วนหลักของโปรแกรม และเมื่อมีการขัดจังหวะสัญญาณนาฬิกาเข้ามา รูปที่ 5.7 แสดงส่วนที่ใช้บริการการขัดจังหวะ

สำหรับโปรแกรมการทำงานจะแสดงในภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.6 แผนภูมิสายงานแสดงการทำงานหลักของโปรแกรมเบี่ยงเบนทิศทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 แผนภูมิสายงานแสดงการทำงานเมื่อมีอินเทอร์รัพต์ของสัญญาณนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
ผลการทำงานของโปรแกรม

จากที่ได้ทดสอบการทำงานของโปรแกรมได้ผลดังนี้

- กรณีที่ใช้โปรแกรมในลักษณะของการเบี่ยงเบนทิศทาง

สามารถทำงานได้ปรกติ นอกจากว่าถ้ามีข้อผิดพลาดเกิดขึ้นระหว่างการทำงานทำให้การทำงานไม่เป็นไปตามลำดับที่กำหนดไว้โปรแกรม จะค้างอยู่ตรงที่ผิดพลาด ผู้ใช้ต้องออกจากโปรแกรมเองแล้วเรียกใช้งานใหม่ ซึ่งสาเหตุที่ทำให้การทำงานค้างเนื่องมาจากหลายกรณีดังนี้

- การป้อนข้อมูลลงบัฟเฟอร์ของแป้นพิมพ์เร็วกว่าที่ตัวโปรแกรมจะรับข้อมูล ทำให้รับข้อมูลไม่ครบจึงเกิดการค้างการทำงาน วิธีแก้ไขคือ ต้องตรวจสอบหน้าจอให้ดีว่าจะใช้ข้อความอะไรเป็นตัวกำหนดการทำงาน ควรจะเป็นข้อความที่พิมพ์ครั้งสุดท้ายของหน้าจอเพราะจะทำให้การทำงานสัมพันธ์กัน
- เกิดข้อผิดพลาดเนื่องจากภาวะแวดล้อมของการใช้โปรแกรมไม่สมบูรณ์ ทำให้โปรแกรมฟ้องข้อผิดพลาดออกมาและหยุดการทำงานไว้ วิธีแก้ไข คือ ต้องกำหนดภาวะแวดล้อมต่างๆ ให้ครบก่อนที่จะเรียกใช้งาน

ถ้ามีการเรียกใช้โปรแกรมติดต่อกันหลายครั้ง อาจจะทำให้โปรแกรมฟ้องว่าหน่วยความจำไม่พอได้ ซึ่งต้องออกจากโปรแกรมก่อนแล้วเรียกใช้งานใหม่ บางครั้งเครื่องอาจหยุดทำงานได้ สาเหตุเพราะว่า การทำงานแบบเบี่ยงเบนทิศทางต้องใช้หน่วยความจำเพิ่มในการบริการอินเทอร์รัพต์เวลาทำให้บางครั้งหน่วยความจำไม่พอได้

- การใช้งานโปรแกรมค้นหาฐานข้อมูลปรกติ

สามารถทำงานได้ปรกติไม่มีปัญหา

จากการทดสอบให้โปรแกรมทำงานพร้อม ๆ กันหลาย ๆ เครื่อง โดยได้ทดสอบให้มีการทำงานพร้อมกันมากที่สุด 5 เครื่อง ซึ่งสามารถทำงานได้ปรกติ การทำงานจะช้าลงเล็กน้อยเท่านั้น สำหรับในการใช้งานแบบเบี่ยงเบนทิศทางจะให้ผลในลักษณะเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
บทวิจารณ์และสรุป

การต่อเครือข่ายฐานข้อมูลซีดีรอมที่ได้พัฒนาขึ้นนี้ จุดประสงค์หลักเพื่อให้สามารถใช้งานฐานข้อมูลซีดีรอมในห้องสมุดกลางให้มีประสิทธิภาพเพิ่มขึ้น โดยการต่อฐานข้อมูลนี้เข้ากับเครือข่ายเน็ตแวร์ ซึ่งทำให้ผู้ใช้งานหลายๆ คนสามารถใช้งานฐานข้อมูลซีดีรอมได้ในเวลาเดียวกัน และยังสามารถใช้งานได้จากเครื่องอื่นๆ ที่ต่อเข้ากับ CAMPUS NETWORK ของสถาบันด้วย

โปรแกรมที่ได้พัฒนาขึ้น มีประโยชน์ในแง่ของการช่วยอำนวยความสะดวกในการเชื่อมต่อกับระบบเน็ตแวร์ เพื่อกำหนดสิ่งที่จำเป็นต่างๆ ในการใช้งานโปรแกรมค้นหาฐานข้อมูลให้สามารถทำงานได้บนเครือข่ายเน็ตแวร์ โดยผู้ใช้ไม่ต้องคำนึงถึงสิ่งเหล่านี้เพียงแต่เรียกใช้งานโปรแกรมปรกติเท่านั้น

สำหรับโปรแกรมค้นหาฐานข้อมูลบนซีดีรอมนั้น ไม่สามารถพัฒนาขึ้นมาได้เองสาเหตุอันเนื่องมาจากไม่มีข้อมูลเกี่ยวกับลักษณะการเก็บข้อมูลของฐานข้อมูลในซีดีรอม ในการวิจัยจึงเลือกใช้โปรแกรม ProQuest ซึ่งเป็นโปรแกรมที่ใช้ค้นหาฐานข้อมูลปรกติในห้องสมุดมาใช้เป็นแนวทางในการศึกษาเพื่อการพัฒนาโปรแกรมอรรถประโยชน์ต่าง ๆ

แต่การใช้งานระบบฐานข้อมูลซีดีรอมนี้ก็มีข้อจำกัดคือถ้ามีผู้ใช้มากคนเกินไปในเวลาเดียวกันจะทำให้การทำงานของระบบช้าลงได้หรือหยุดการทำงานโดยไม่ได้คาดหมาย เพราะต้องแบ่งการทำงานกันมากขึ้น รวมทั้งมีการใช้ทรัพยากรต่าง ๆ มากขึ้น ดังนั้นระบบนี้จึงเหมาะสมกับจำนวนผู้ใช้ในระดับหนึ่งเท่านั้น

นอกจากได้พัฒนาโปรแกรมดังกล่าวแล้วยังได้พัฒนาโปรแกรมเพื่อเบี่ยงเบนทิศทางขึ้นเพื่อที่จะใช้เป็นแนวทางที่จะพัฒนาระบบเครือข่ายซีดีรอมขึ้นใช้เองในอนาคต แนวทางในการพัฒนาอาจทำได้ทั้งการพัฒนาเพื่อใช้งานระบบนี้ในระบบปฏิบัติการยูนิกส์หรือการพัฒนาเพื่อการใช้งานบนระบบปฏิบัติการดอสก็ได้ โดยการใช้เครื่องคอมพิวเตอร์เครื่องหนึ่งทำงานโปรแกรมค้นฐานข้อมูลนี้ตลอดเวลา และคอยรับไฟล์ข้อมูลเข้าที่ส่งมาจากสถานีงานและนำข้อมูลที่ได้นี้ไปประมวลผล จากนั้นจึงทำการส่งผลที่ได้ในรูปของไฟล์ข้อมูลกลับมายังสถานีงาน เพื่อทำการแสดงผลให้ผู้ใช้ต่อไป การทำเช่นนี้อาศัยแนวทางจากการเบี่ยงเบนทิศทางข้อมูลนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.
โปรแกรมควบคุมการแมปหมวดในเน็ตแวร์

```
/*****************************************************************************

    PROGRAM  :    VOLMAP.C
    USE      :    NWAPI  LIBRARY
    BY       :    APIRAMON  DAMRONGSIRI
                  APISIT    SUKSAKORN
    INPUT    :    NONE
    OUTPUT   :    NEW DRIVES FOR MAPPED VOLUME

*****************************************************************************/

#include <nit.h>
#include <ntt.h>
#include <alloc.h>
#include <conio.h>
#include <stdio.h>
#include <string.h>
#include <process.h>

int         localDrive;
int         drive[32] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
char        ServerName[48];
WORD        DefaultConnectionID;
NWDIR_INFO      *dirInfo;

void        SearchDrive(WORD);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void main()
{
        int     i;
        int     code;
        WORD        ID;

        clrscr();
        /* find default connection ID of main server */
        DefaultConnectionID = GetDefaultConnectionID();

        /* get file server name */
        GetFileServerName(DefaultConnectionID,ServerName);
        printf("Main Server Name : %s\n\n",ServerName);

        /* mark local drives */
        localDrive =  GetNumberOfLocalDrives();
        for (i  = 0;  i < localDrive; i++)
                drive[i] = 1;

        /* search drive for new volume and map it to new drive */
        SearchDrive(DefaultConnectionID);

/*************************************************************************

        ; mark search drives
        ; if you don't want to map to search drives
        for (i = localDrive; i < 32; i++)
        {
                driveletter[0] = 65+i;
                completionCode =IsSearchDrive(driveletter[0]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if (completionCode)
                drive[i] = 1;
        }
    ************************************************************/
    /* search for new attach server */
    for (ID = 1; ID <= 8; ID++)
    {
        /* Is that Connection ID in use? */
        code = IsConnectionIDInUse(ID);
        /* If Connection ID is in use */
        if (code)
        {
            /* find server name */
            strcpy(ServerName,"");
            GetFileServerName(ID,ServerName);

            /* if it is not Default Connection ID */
            if (ID != DefaultConnectionID)
            {
                printf("\nAttach Server Name : %s\n",ServerName);
                /* set Connection ID to new server */
                SetPreferredConnectionID(ID);
                /* search drive for new volume and map it to new drive */
                SearchDrive(ID);
            }
        }
    }

    /* set Connection ID to old server */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    SetPreferredConnectionID(DefaultConnectionID);
}

void SearchDrive(WORD ConnectionID)
{
    int     completionCode;
    int     volumeNumber, totalvolume = 0;
    int     i, j = 0, k;
    char    driveletter[2];
    char    VolumeName[16];
    char    volume[100], voltemp[100];
    char    command[26];
    char    path[255];
    BYTE    driveNumber, directoryHandle;

    /* clear variable */
    strcpy(volume,"");
    /* find volume name in volume table */
    for (volumeNumber = 0; volumeNumber < 32; volumeNumber++)
    {
        completionCode = GetVolumeName(volumeNumber,VolumeName);
        if (!completionCode && (strcmp(VolumeName,"") != 0))
        {
            printf("Volume Name : %s\n", VolumeName);

            /* find volume name in each drive */
            if (ConnectionID == DefaultConnectionID)
                for (driveNumber = 0; driveNumber < 32; driveNumber++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        /* get directory handle */
        directoryHandle = GetDirectoryHandle(driveNumber);

        /* allocate memory */
        dirInfo = (NWDIR_INFO *)malloc(sizeof(NWDIR_INFO));
        /* get directory information */
        completionCode = GetDirInfo(ConnectionID,
        directoryHandle,dirInfo);

        /* if it is complete */
        if (!completionCode)
        {
                /* clear variable */
                strcpy(voltemp,"");
                /* copy volume name from directory information */
                strcat(voltemp,dirInfo->volumeName);
                /* if it is already mapped volume, find the new one */
                if (strcmp(VolumeName,voltemp) == 0)
                        goto nextvolume;
                /* else*/
                else
                {
                        /* if it is last drive */
                        if (driveNumber == '\x1F')
                        {
                                /* copy volume name to temp */
                                strcat(volume,VolumeName);
                                strcat(volume," ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        /* increment total volume to map */
                                        totalvolume++;
                                    }
                                }
                            }
                            /* if it is not complete */
                            else
                            {
                                /* if it is last drive*/
                                if (driveNumber == '\x1F')
                                {
                                    /* copy volume name to temp */
                                    strcat(volume,VolumeName);
                                    strcat(volume," ");
                                    /* increment total volume to map */
                                    totalvolume++;
                                }
                            }
                        }
                    /* if it is not in main server */
                    else
                    {
                        /* copy volume name to temp */
                        strcat(volume,VolumeName);
                        strcat(volume," ");
                        /* increment total volume to map */
                        totalvolume++;
                    }
                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        /* find next volume */
        nextvolume :
        ;
}

printf("----------\n");

/* if it is in main server */
if (ConnectionID == DefaultConnectionID)
{
        /* find full path of directory */
        for (driveNumber = localDrive; driveNumber < 32; driveNumber++)
        {
                {
                        completionCode = GetFullPath(driveNumber,path);
                        if (!completionCode)
                        {
                                printf("%c:%s\n",65+driveNumber,path);
                                /* mark drive in used */
                                drive[driveNumber] = 1;
                        }
                }
        }
        printf("----------\n");
}

k = 0;
/* if total is not equal zero */
if (totalvolume != 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    while (totalvolume != 0)
    {
        /* find next empty drive */
        while (drive[k] == 1)
            k++;

        /* mark drive in used */
        drive[k] = 1;

        /* convert integer to character */
        driveletter[0] = 65+k;
        driveletter[1] = ':';
        driveletter[2] = '\0';

        /* clear variable */
        strcpy(voltemp,"");

        /* volume character counter */
        i = 0;
        /* find volume name to map */
        while (volume[j] != ' ')
        {
            voltemp[i] = volume[j];
            i++;
            j++;
        }
        voltemp[i++] = '\0';
        j++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            /* decrement map volume */
            totalvolume--;

            /* build map command */
            strcpy(command,"");
            if (ConnectionID != DefaultConnectionID)
            {
                strcat(command,ServerName);
                strcat(command,"\\");
            }
            strcat(command,voltemp);
            strcat(command,":");

            /* map next drive */
            completionCode = MapDrive(ConnectionID,0xFF,
            command,DRIVE_ADD,0,driveletter);

            /* show result */
            if (!completionCode)
            {
                printf("Volume %s is already mapped\n",voltemp);
                printf("%s\\%s\\%s\n",driveletter,ServerName,voltemp);
            }
            else
                printf("Can't map drive volume %s\n",voltemp);
        }
    }
    else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            printf("No new volume to map.\n");
        /* free memory */
        free(dirInfo);
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.
โปรแกรมเบี่ยงเบนข้อมูลเข้าและออก

```
/* ************************* MAIN PROJECT ****************************
                CD-ROM NETWORKING AND REDIRECTION
   ********************************************************************** */
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<ctype.h>
#include<string.h>
#include<dos.h>
#include<process.h>
#include"g:\project\convert.h"

#define TCK   0x1c                  /* Interrupt Timer */
#define _MAX  120                   /* Maximun of array size for keep screen option */
#define TRUE  1                     /* Case true */
#define FALSE 0                     /* Case false */
#define END   -1
#define _MAXKEY 12                  /* Maximun of keyword for search */

FILE *_fp;
FILE *_frt;
char keyword[_MAXKEY][50];
char press[_MAX][80];
char data[_MAX][80];                /* Store input screen data */
char attr[_MAX];                    /* Attribute of input screen data */
char outfile[30] = "\0";
```

เอกสารนี้จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char Tot[_MAXKEY][10];
int  outlevel=0;
int  screen[1][78];
int  _x=0,_y=0;
int  _IndexMark = 0;
int  ItemLine = 15;                    /* Line for check item have result */
int  mark[_MAXKEY][2];
int  Markset=0;                        /* To check line is already mark */
int  Markround=0;                      /* Count round of mark */
int  MarkCount=0;                      /* Count number of mark in one round */
int  MarkLine=5;                       /* Line of mark line */
int  srow[_MAX];                       /* Start row of coordinate of string */
int  scolumn[_MAX];                    /* Start column of coordinate of string */
int  erow[_MAX];                       /* End row of coordinate of string */
int  ecolumn[_MAX];                    /* End column of coordinate of string */
int  maximum=0;
int  data_index=0;                     /* Save index of input screen dat */
int  char_index=0;                     /* Save index of asc for put in keyboard buffer */
unsigned char asc[50];                 /* asc is code of keyboard for put in buffer */
unsigned char as,sc;

void interrupt process(void);
void interrupt (*OldInt1c)(void);      /* Old interrupt vector */
void PutBuffer(void);
void PutScankey(unsigned char ascii,unsigned scan);
void CheckFile(void);
void Read_data(char f_name[]);
void Gen_file1(char *database);
void Gen_last(int Id);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Gen_outfile(void);
void set_mark(void);
void SetMark_Last(void);
void CheckTotal(void);
void CreateOutFile(void);
void RWfile(FILE *fptr,FILE *fntr,int Total);
void SetSystem(void);
void _Pull_screen(int s_column,int s_row,int e_column,int e_row);
int  _Check_screen(int s_column,int s_row,int e_column,
                   int e_row,char *_st,char att);

/* ********************** Timer interrupt service routine ************************** */
void interrupt process(void)
{
 int test;
 int i=0;
 int index = 0;

 disable();
 /* Check text screen is correct */
 test = _Check_screen(scolumn[data_index],srow[data_index],
                      ecolumn[data_index],erow[data_index],
                      data[data_index],attr[data_index]);

 if(test == TRUE && data_index < maximum)
 /* Test = 1 screen is correct */
 {
  switch(press[data_index][0])
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case '๑':       /* Case for delay 1 timer tick */
 ++data_index;
break;

case ' ':       /* Case for save screen in output levet 1 */
 _Pull_screen(11,ItemLine,76,ItemLine);
 for(i=0;i<15;++i)
 {
   fputc(screen[0][_x],_fp);   /* Save on file */
   ++_x;
 }
 if(_x>=75)     /* End of line */
 {
  fputc('\n',_fp);
  ++ItemLine;
  ++data_index;
  _x=0;
 }
 break;

case '๑':       /* This case for mark item */

 if(MarkCount < 10)
 {
  set_mark();           /* Function for mark item */

  _Pull_screen(78,21,79,21);  /* Check for last item */
  if(Markset == 1 && screen[0][0] == ' ')
    MarkCount = 20;    /* MarkCount is 999 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 if(MarkLine == 5)      /* Check first time only */
 {
  _Pull_screen(44,4,46,4);  /* Check for one item */
  if(screen[0][0] == ' ' && screen[0][1] == '1'
       && screen[0][2] == ' ')
       MarkCount = 20;      /* Markcount is 999 */
 }
}

switch(MarkCount){
 case 25:                /* Check save is finish */
 case 15:
  test = _Check_screen(20,11,42,11,"Output to file complete",63);
  if(test == 1)
  {
       _Scan_Code("@ENTER",&as,&sc);   /* Press Enter */
       PutScankey(as,sc);
       MarkCount++;
       ++Markround;
  }
  break;

 case 26:
 case 16:                /* Press Shift+F9 */
  _Scan_Code("@SHF+F9",&as,&sc);
  PutScankey(as,sc);
  if(MarkCount == 26)
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ++data_index;
        MarkCount = 0;
        MarkLine = 4;
        Markset = 0;
        Markround = 0;
    }
    else MarkCount=0;
    break;

  case 24:          /* Save-only mark items */
  case 14:
    test = _Check_screen(23,12,34,12,"Marked Items",15);
    if(test == TRUE)
    {
        _Scan_Code("@ENTER",&as,&sc);       /* Press Enter */
        PutScankey(as,sc);
        MarkCount++;
    }
    break;

  case 23:          /* Starting save */
  case 13:
    test = _Check_screen(24,16,28,16,"START",15);
    if(test == TRUE)
    {
        _Scan_Code("@ENTER",&as,&sc);       /* Press Enter */
        PutScankey(as,sc);
        MarkCount++;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;

      case 22:          /* Press F4 for saye items */
      case 12:
       _Scan_Code("@F4",&as,&sc);
       PutScankey(as,sc);
       if(Markround == 0)
            MarkCount++;
       else MarkCount += 2;
       break;

      case 21:
      case 11:
       MarkCount++;
       break;

      case 20:
      case 10:
        MarkCount++;
        break;
    }

   if(MarkLine <= 21)     /* If mark line not end of line */
     ++MarkLine;          /* Increment mark line */
   break;

  case 'o':               /* Case for check keyword have 0 item */
   _Pull_screen(1,ItemLine,76,ItemLine);
   if(ItemLine < 22)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            press[data_index][index] == '$')
        {
            asc[i] = 0x0d;
            asc[i+1] = 0x1c;
            i += 2;
            index += 2;
        }
        else{
                _Convert_Code(press[data_index][index],&asc[i],&asc[i+1]);
                i += 2;
                ++index;
            }
    }
    asc[i] = '\0';
    PutBuffer();
    if(char_index == 0)   /* Check all character is put in buffer */
            ++data_index;
    break;
    }
}

enable();
(*OldInt1c)();      /* Call old timer interrupt service routine */
}

/* ***************** Put keyboard code in Keyboard buffer ************** */

void PutBuffer(void)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char far *last_addr;
char far *next_addr;
char far *last_char;
char far *next_char;
int  i;
int  buf_full;

next_addr = (char far *)0x0000041a;     /* Address of last addr */
next_char = (char far *)((int)*next_addr + 0x00000400);
                                        /* Compute address of next char */
last_addr = (char far *)0x0000041c;
last_char = (char far *)((int)*last_addr + 0x00000400);
                                        /* Compute address of last char */
i = char_index;
buf_full = 0;
if(last_char == next_char)          /* Check buffer is empty or not */
{
                                        /* While not end of string and buffer not full */
      while(asc[i] != '\0' && buf_full == 0)
      {
        *(last_char++) = asc[i];                /* Put code in buffer */
        ++i;
        *(last_char++) = asc[i];                /* Put code in buffer */
        ++i;

        if(last_char == (char far *)0x0000043e)         /* End of buffer */
              last_char = (char far *)0x0000041e;       /* Goto start buffer */

        *last_addr = *last_addr + 2;            /* Increment memory 0x0000041c */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(*last_addr > 0x3d)                    /* Check for it exceed 0x3d */
                  *last_addr = *last_addr - 0x20;

            *last_addr = *last_addr + 2;
            if(*last_addr == *next_addr)             /* Check for buffer is full */
                  buf_full = 1;
            *last_addr = *last_addr - 2;
          }
      }
     if(asc[i] == '\0')                              /* asc[i] = \0 is end of this code */
       char_index = 0;                               /* Clear char_index */
     else char_index = i;                            /* Save index i in char_index */
}
/* ************************************************************************ */

/* ************ Put scan keyborad code in buffer ********************** */

void PutScankey(unsigned char ascii,unsigned scan)
{
  char far *last_addr;
  char far *next_addr;
  char far *last_char;
  char far *next_char;

  next_addr = (char far *)0x0000041a;    /* Address of last_addr */
  next_char = (char far *)((int)*next_addr + 0x00000400);
                                         /* Compute address of next char */
  last_addr = (char far *)0x0000041c;
  last_char = (char far *)((int)*last_addr + 0x00000400);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        /* Compute address of last char */
  if(last_char == next_char)            /* Check buffer is empty or not */
   {
                                        /* While not end of string and buffer not full */
          *(last_char++) = ascii;       /* Put code in buffer */
          *(last_char++) = scan;        /* Put code in buffer */

          if(last_char == (char far *)0x0000043e)       /* End of buffer */
                last_char = (char far *)0x0000041e;     /* Goto start buffer */

          *last_addr = *last_addr + 2;          /* Increment memory 0x0000041c */
          if(*last_addr > 0x3d)                 /* Check for it exceed 0x3d */
                *last_addr = *last_addr - 0x20;

   }
}
/* ************************************************************** */

/* ************** Check screen is equal input string ************** */

int _Check_screen(int s_column,int s_row,int e_column,int e_row,char *_st,
        char att)
{
  char far *s_scrpst;
  char buffer[80];
  char attribute;
  int  s_location;
  int  e_location;
  int  test;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int _i;

 /* Compute address of screen */
s_location = ((s_row-1) * 80 + (s_column-1))*2;
e_location = ((e_row-1) * 80 + (e_column-1))*2;

s_scrpst = (char far *)0xb8000000 + s_location;          /* Adress of Video RAM */

_i=0;
while(s_location <= e_location)                          /* Check for end of screen */
{
        buffer[_i] = *s_scrpst;                          /* Save data of screen in buffer */
        s_scrpst = s_scrpst + 1;                         /* Ship for get data */
        attribute = *s_scrpst;
        s_scrpst = s_scrpst + 1;
        ++_i;
        s_location = s_location + 2;
}
buffer[_i] = '\0';                       /* Mark end of string */

_i = 0;
test = TRUE;                             /* Set test is TRUE */
while(buffer[_i] != '\0')
{
        if(buffer[_i] == _st[_i])        /* Check for screen is equal array st */
          ++_i;
        else{
                test = FALSE;            /* String FALSE */
                ++_i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
  }
  if(attribute == att)
  {;}
  else test = FALSE;
  return test;
}

/* ************************************************************** */

/* ******************** Check screen option file ********************** */

void CheckFile(void)
{
  FILE *fp;
  char buffer[80],tmp[5];
  int i=0,k=0;
  int chkblank=0;                    /* Check character | */
  int count=0;

  if((fp = fopen("scr_opt.dat","rt"))==NULL)
  {
    printf("Error in open file scr_opt.dat\n");
    exit(1);
  }

  while(fgets(buffer,80,fp)!=NULL)          /* Read input file */
  {
    if(ferror(fp))                          /* Check file error */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        printf("File error");
        exit(1);
}

while(buffer[i] != '\n' && buffer[i] != '\0')        /* Check end of line */
{
        if(buffer[i] == '|')
        {
         chkblank++;                                 /* Chang input option */
         ++i;
        }
        if(chkblank==0)                              /* Option 0 */
        {
         tmp[k] = buffer[i];
         ++i;
         ++k;
         if(buffer[i] == '|')
         {
                tmp[k] = '\0';
                scolumn[count] = atoi(tmp);          /* Save start column in scolumn */
                k = 0;
         }
        }
        else
        if(chkblank==1)                              /* Option 1 */
        {
         tmp[k] = buffer[i];
         ++i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  ++k;
  if(buffer[i] == '|')
  {
        tmp[k] = '\0';
        srow[count] = atoi(tmp);        /* Save start row in srow */
        k = 0;
  }
}
else
if(chkblank==2)                         /* Option 2 */
{
  tmp[k] = buffer[i];
  ++i;
  ++k;
  if(buffer[i] == '|')
  {
        tmp[k] = '\0';
        ecolumn[count] = atoi(tmp);     /* Save end column in ecolumn */
        k = 0;
  }
}
else
if(chkblank==3)                         /* Option 3 */
{
  tmp[k] = buffer[i];
  ++i;
  ++k;
  if(buffer[i] == '|')
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        tmp[k] = '\0';
        erow[count] = atoi(tmp);          /* Save end row in erow */
        k = 0;
  }
 }
 else
 if(chkblank==4)                          /* Option 4 */
 {
  data[count][k] = buffer[i];             /* Put screen text in data */
  ++k;
  ++i;
  if(buffer[i] == '|')
  {
        data[count][k] = '\0';
        k=0;
  }
 }
 else
 if(chkblank==5)                          /* Option 5 */
 {
  tmp[k] = buffer[i];
  ++i;
  ++k;
  if(buffer[i] == '|')
  {
        tmp[k] = '\0';
        attr[count] = (char)atoi(tmp);    /* Put attribute of screen */
        k = 0;
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        else
        if(chkblank==6)                         /* Option 6 */
        {
          press[count][k] = buffer[i];          /* Put keyboard press in press */
          ++i;
          ++k;
          if(buffer[i] == '\n' || buffer[i] == '\0')
          {
                press[count][k] = '\0';
                k = 0;
          }
        }
    }

    maximum++;
    count++;                                    /* Increment count */
    i=0;
    chkblank=0;                                 /* Initialize chkblank */
  }
}
/* ********************************************************************** */

/* ****************** Read data from input file ************************ */

void Read_data(char f_name[])
{
  FILE *_fpt;
  char buffer[80];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char sub_str[80];
char database[7] = "\0";
int  i=0;
int  index=0;
int  blank=0;
int  type;
int  back;
int  IndexKey = 0;

if((_fpt = fopen(f_name,"rt"))==NULL)
{
  printf("Error in open file %s",f_name);
  exit(1);
}

while(fgets(buffer,80,_fpt) != NULL)     /* Data not EOF */
{
        back = 1;                        /* Set case is back '=' */
        /* Process by one character */
        while(buffer[i] != '\0' && buffer[i] != '\n')
        {
          switch(back)
          {
            case 1:                      /* Now string is in front of '=' */
              if(buffer[i] != '=')
              {
                  back = 1;
                  if(buffer[i] != ' ')   /* Cancel character '=' */
                  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            sub_str[index] = tolower(buffer[i]);
            ++index;
        }
    }
    else{
            back = 0;               /* Set state of string to back '=' */
            sub_str[index] = '\0';
            index = 0;
            /* Search type of command */
            if(strcmp(sub_str,"database")==0)
                    type = 0;
            else
            if(strcmp(sub_str,"keyword")==0)
                    type = 1;
            else
            if(strcmp(sub_str,"outputlevel")==0)
                    type = 2;
            else
            if(strcmp(sub_str,"outputfile")==0)
                    type = 3;
        }
    break;
case 0:                 /* Now string is back '=' */
        if(blank == 0 && buffer[i] == ' ')
        {;}             /* Character is back '=' but in front of first data */
        else {
                blank = 1;  /* Now all character is data */
                sub_str[index] = tolower(buffer[i]);
                ++index;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }

            break;
          }
          ++i;
        }
        sub_str[index] = '\0';
        blank = 0;
        /* Check type of data and save in memory */
        if(type == 0)
          strcpy(database,sub_str);
        else if(type == 1)
            {
                strcpy(keyword[IndexKey],sub_str);
                ++IndexKey;
            }
            else if(type == 2)
                        outlevel = atoi(sub_str);
                else if(type == 3)
                                strcpy(outfile,sub_str);

        index = 0;
        i = 0;
    } /* End while */
    fclose(_fpt);
    /* Create screen option file */
    i=0;

    Gen_file1(database);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
while(i != IndexKey)
{
  Gen_file2(i);
  ++i;
}
Gen_last(i-1);

switch(outlevel)  /* Check level of output file */
{
  case 1:
    if((_fp=fopen(outfile,"wt"))==NULL)
    {
        printf("Error in open file %s",outfile);
        exit(0);
    }
    break;

  case 2:
  case 3:
    Gen_outfile();   /* Generate output file */
    break;
}

}

/* ************************************************************************ */

/* ********************* Generate screen option file ********************** */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Gen_file1(char *database)
{
 char str[3];

 if((_frt = fopen("scr_opt.dat","wt"))==NULL)
 {
  printf("Error in open file scr_opt.dat");
  exit(1);
 }

 if((strcmp(database,"\0"))!=0)
 {
  if((strcmp(database,"inspec"))==0)  /* Case disc is inspec database */
  {
        fputs("10|8|38|8|DISSERTATION ABSTRACTS ONDISC|15|@DOWN\n",_frt);
        fputs("10|9|15|9|INSPEC|15|@ENTER\n",_frt);
  }
  if((strcmp(database,"dao"))==0)    /* Case disc is dao database */
  {
        fputs("10|8|38|8|DISSERTATION ABSTRACTS ONDISC|15|@ENTER\n",_frt);
  }
 }
 else printf("Database Error: Not define database");
}

/* ********************************************************************** */

/* ******************* Generate screen option file ********************** */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Gen_file2(int Id)
{
 char str[3];
 int key_length;

 key_length = strlen(keyword[Id]);      /* Save length of keyword */

 if((strcmp(keyword[Id],"\0"))!=0)
 {
  /* Generate command for keyword */
  fputs("34|4|45|4|Search Entry|31|",_frt);
  fputs(keyword[Id],_frt);
  fputs("$$\n",_frt);        /* $$ is same as @ENTER */
 }

 switch(outlevel){          /* Generate command for check output */
  case 0:
  break;

  case 3:
    key_length = key_length + 3 - 1;
    itoa(key_length,str,10);
    fputs("3|5|",_frt);
    fputs(str,_frt);
    fputs("|5|",_frt);
    fputs(keyword[Id],_frt);
    fputs("|31|o\n",_frt);

    fputs("3|5|",_frt);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  fputs(str,_frt);
  fputs("|5|",_frt);
  fputs(keyword[Id],_frt);
  fputs("|31|@ENTER\n",_frt);
  fputs("78|5|78|5|□|31|๑\n",_frt);          /* For save output on file */
  fputs("78|5|78|5|□|31|@ESC\n",_frt);   /* For save output on file */
break;

case 2:
  key_length = key_length + 3 - 1;
  itoa(key_length,str,10);
  fputs("3|5|",_frt);
  fputs(str,_frt);
  fputs("|5|",_frt);
  fputs(keyword[Id],_frt);
  fputs("|31|๐\n",_frt);

  fputs("3|5|",_frt);
  fputs(str,_frt);
  fputs("|5|",_frt);
  fputs(keyword[Id],_frt);
  fputs("|31|@ENTER\n",_frt);

  fputs("78|6|78|6| |31|@F4\n",_frt);
  fputs("24|16|28|16|START|15|@UP\n",_frt);
  fputs("11|13|20|13|File name:|63|๑\n",_frt);
  fputs("11|13|20|13|File name:|63|@UP\n",_frt);
  fputs("24|9|27|9|Long|15|๑\n",_frt);
  fputs("24|9|27|9|Long|15|@UP\n",_frt);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    fputs("24|8|28|8|Short|15| \n",_frt);
    fputs("24|8|28|8|Short|15|@ESC\n",_frt);
    fputs("78|5|78|5|□|31|◉\n",_frt);          /* For save output on file */
    fputs("78|5|78|5|□|31|@ESC\n",_frt);   /* For save output on file */
   break;

   case 1:
    key_length = key_length + 3 - 1;
    itoa(key_length,str,10);
    fputs("3|5|",_frt);
    fputs(str,_frt);
    fputs("|5|",_frt);
    fputs(keyword[Id],_frt);
    fputs("|31| \n",_frt);

    fputs("78|5|78|5| |31|@F3\n",_frt);
   break;
  }
}

/* ************************************************************** */

/* ********************** Generate screen option file ****************** */
void Gen_last(int Id)
{
 int key_length;
 char str[4];

 switch(outlevel){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 case 2:
 case 3:
   fputs("34|4|45|4|Search Entry|31|@SHF+F10\n",_frt);
   fputs("29|17|31|17|Yes|15|@ENTER\n",_frt);

   fclose(_frt);
 break;

 case 1:
   fputs("78|5|78|5| |31|@SHF+F10\n",_frt);
   fputs("29|17|31|17|Yes|15|@ENTER\n",_frt);
   fclose(_frt);
 break;
 }
}

/* ********************************************************************** */

/* ******************** Pull data from Video RAM ********************** */

void _Pull_screen(int s_column,int s_row,int e_column,int e_row)
{
 char far *s_scrptr;
 int s_location,e_location;          /* Position of screen
 int row;
 int ro=0,co=0;
 char ch;

 row = s_row;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  while(row-1 != e_row)
  {
    /* Compute address of Video RAM */
    s_location = ((row-1) * 80 + (s_column-1))*2;
    e_location = ((row-1) * 80 + (e_column-1))*2;

    s_scrptr = (char far *)0xb8000000 + s_location;        /* Adress of Video RAM */

    while(s_location <= e_location)                        /* Check for end of screen */
    {
          screen[ro][co] = *s_scrptr;
          ++co;

          s_scrptr = s_scrptr + 2;

          s_location = s_location + 2;  /* Compute location of current screen */
    }
    screen[ro][co] = '\n';
    screen[ro][co+1] = '\0';
    co = 0;
    ++ro;
    row = row + 1;
  }
}

/* ************************************************************************ */

/* ********************** Generate output file ************************** */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Gen_outfile(void)
{
 FILE *_fpt;
 char buffer[80];
 char sub_str[40];
 char temp[30][80];
 int  i=0;
 int  j;
 int  index=0;
 int  find=0;
 /* Set output file to ~temp.S$$ */
 if((_fpt = fopen("outopts.dat","rt"))==NULL)
 {
  printf("Error in open file outopts.dat");
  exit(1);
 }

 while((fgets(buffer,80,_fpt))!=NULL)
 {
        strcpy(temp[index],buffer);
        ++index;
 }

 i=0;
 j=0;
 while(i != index)        /* Find string "driveavail" */
 {
        while(temp[i][j] != '\0' && temp[i][j] != '\n' && find == 0)
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        switch(temp[i][j])
        {
                case '=':
                 buffer[j] = '=';
                 buffer[j+1] = '\0';
                 if((strcmp(buffer,"driveavail="))==0)
                 {
                        strcat(buffer,"~temp.$$$");
                        strcat(buffer," B:\\ \n");
                        find = 1;
                        for(j=0;j <= (strlen(buffer));++j)
                         temp[i][j] = buffer[j];
                        temp[i][j+1] = '\0';
                        i = index - 1;
                 }
                 ++j;
                 break;

                default:
                 buffer[j] = temp[i][j];
                 find = 0;
                 ++j;
                 break;
          }
        }
        ++i;
        j=0;
}
fclose(_fpt);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    /* Write change in outopts.dat */
    if((_fpt = fopen("outopts.dat","wt"))==NULL)
    {
      printf("Error in open file outopts.dat");
      exit(1);
    }

    i=0;
    while(i != index)
    {
            fputs(temp[i],_fpt);
            ++i;
    }
    fclose(_fpt);
}

/* ************************************************************** */

/* ************************* Mark items for save ********************** */

void set_mark(void)
{
    int test;

    test = _Check_screen(62,24,73,24,"Loading Text",110);
    if(test == 1)
    { return; }
    if(MarkLine == 22)              /* If end of line go to new function */
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    SetMark_Last();              /* Function check and mark item at line 22 */
    return;
  }
  /* Next item check at upper line for blank */
  if(MarkCount >= 1)
   _Pull_screen(3,MarkLine-1,10,MarkLine-1);
        /* First item check at next line for blank */
  else _Pull_screen(3,MarkLine+1,10,MarkLine+1);

        /* If new item mark */
  if(screen[0][2] == ' ' && screen[0][3] == ' ' && screen[0][4] == ' ')
  {
      _Scan_Code("@F9",&as,&sc);
      PutScankey(as,sc);
      ++MarkCount;     /* Increment Markcount */
      Markset = 1;     /* Set markset = 1 is mark already */
      return;
  }
        /* Check this MarkLine; if it is blank Markset = 0 */
  _Pull_screen(3,MarkLine,10,MarkLine);
  if(screen[0][2] == ' ' && screen[0][3] == ' ' && screen[0][4] == ' ')
  {
   Markset = 0;      /* Set markset = 0 is not mark already */
  }
  _Scan_Code("@DOWN",&as,&sc);    /* Press DOWN key */
  PutScankey(as,sc);
}

/* ********************************************************************** */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* ************************ Mark items in last line ****************** */

void SetMark_Last(void)
{
  switch(Markset){
   case 0:                              /* Markset = 0 is not already mark */
    _Scan_Code("@F9",&as,&sc);          /* Mark this item */
    PutScankey(as,sc);
    ++MarkCount;                        /* Increment Mark count */
    Markset = 1;                        /* Markset = 2 is new case for line 22 */
   break;

   case 1:                              /* Markset = 1 is Mark already */
    _Scan_Code("@DOWN",&as,&sc);
    PutScankey(as,sc);
    Markset = 2;                        /* Already mark set Markset to 2 */
   break;

   case 2:                              /* Markset = 2 for line = 22 */
    _Pull_screen(3,21,10,21);
    if(screen[0][2] == ' ' && screen[0][3] == ' ' &&
          screen[0][4] == ' ' && screen[0][5] == ' ')
    {     /* Mark item press F9 */
          Markset = 3;                  /* Set Markset to 3 is Mark */
    }
    else {  /* Not new item press DOWN key */
              _Scan_Code("@DOWN",&as,&sc);
              PutScankey(as,sc);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
    break;

    case 3:     /* Markset = 3 is Mark and Increment count */
     _Scan_Code("@F9",&as,&sc);
     PutScankey(as,sc);
     ++MarkCount;          /* Increment MarkCount */
     Markset = 1;          /* Set Markset to 1 is Mark normally */
    break;
  }
}

/* ************************************************************** */

/* ************************************************************** */

void CheckTotal(void)
{
 int i;
 int j=0;

 for(i=61;i<75;++i)
 {       /* Put total item and save in Total */
  if(screen[0][i] >= '0' && screen[0][i] <= '9')
  {
   Tot[_IndexMark][j] = screen[0][i];
   ++j;
  }
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Tot[_IndexMark][j] = '\0';
  mark[_IndexMark][1] = 0;    /* Set mark = 0 that have data */
  ++_IndexMark;
  mark[_IndexMark][1] = END;
}
```

```
void CreateOutFile(void)
{
  FILE *fptr;
  FILE *fntr;
  int  i;
  char str_num[7];

  if((fntr = fopen(outfile,"wt"))==NULL)
  {
    printf("Error in open file %s",outfile);
    exit(0);
  }

  if((fptr = fopen("~temp.$$$","rt"))==NULL)
  {
    i=0;          /* Check all keyword have 0 item */
    while(mark[i][1] != END)
    {
      mark[i][1] = atoi(Tot[i]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if(mark[i][1] == 0)
    {;}
    else{
            printf("Error in open file ~temp.txt\n");
            exit(0);
          }
    ++i;
  }
  i=0;          /* If it have 0 item sure. Save in file */
  while(mark[i][1] != END)
  {
    fputs("==== Keyword   : ",fntr);
    fputs(keyword[i],fntr);
    fputs(" =====\n",fntr);
    fputs("==== Total Data : 0\n\n\n",fntr);
    ++i;
  }
  fclose(fntr);
  exit(1);          /* Normal Terminate */
}

i=0;              /* Save output in user file */
while(mark[i][1] != END)
{
  mark[i][1] = atoi(Tot[i]);   /* Case keyword have 0 item */
  if(mark[i][1] != 0)
  {
    fputs("==== Keyword   : ",fntr);
    fputs(keyword[i],fntr);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      fputs(" =====\n",fntr);
      fputs("==== Total Data : ",fntr);
      itoa(mark[i][1],str_num,10);
      fputs(str_num,fntr);
      fputs("\n\n\n",fntr);
      RWfile(fptr,fntr,mark[i][1]);
    }
    else {
            fputs("==== Keyword    : ",fntr);
            fputs(keyword[i],fntr);
            fputs(" =====\n",fntr);
            fputs("==== Total Data : 0\n\n\n",fntr);
          }
    ++i;
  }

  fclose(fntr);
  fclose(fptr);
}

/* ************************************************************************ */

/* ****************** Function for write user output file ************** */

void RWfile(FILE *fptr,FILE *fntr,int Total)
{
  int Count=0;
  int round=0;
  char buffer[80];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int length;

while(fgets(buffer,79,fptr)!=NULL)
 {
  if(strcmp(buffer,"\n")==0)
  {
      if(round == 0)
      {
       round = 1;
      }
      fputs(buffer,fntr);
  }
  else {
        if(round == 1)
        {
         ++Count;
         if(Count >= Total)
         {
             length = strlen(buffer);
             fseek(fptr,-(long)length-1,SEEK_CUR);
             return;
         }
        }
        fputs(buffer,fntr);
        round = 0;
       }
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* ********************************************************************** */

/* ********************** Set config of netware system ****************** */

void SetSystem(void)
{
 FILE *fp;
 char buffer[80];
 char server[10][15];
 char user[10][15];
 char attach[30];
 char logout[30];
 int  i;
 int  j;
 int  k;
 int  count=0;

 if((fp=fopen("netinfo.dat","rt"))==NULL)
 {
  printf("Error in open file netinfo.dat\n");
  exit(0);
 }
 i=0;
 j=0;
 while(fgets(buffer,79,fp)!=NULL)
 {
  while(buffer[i] != '=')
    ++i;
  ++i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  k=0;
  while(buffer[i] != '/'&& buffer[i] != '\\')
  {
    server[j][k] = buffer[i];      /* Save all server name */
    ++i;
    ++k;
  }
  server[j][k] = '\0';

  k=0;
  while(buffer[i] != '\n')
  {
    user[j][k] = buffer[i];      /* Save all user name */
    ++i;
    ++k;
  }
  user[j][k] = '\0';

  ++j;
  ++count;
  i=0;
}
fclose(fp);

for(i=0;i<count;++i)
{
  strcpy(attach,"attach ");      /* Attach cd-rom server */
  strcat(attach,server[i]);
  strcat(attach,user[i]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    system(attach);
  }
   /* Create config file */
  if((fp=fopen("cfg.bat","wt"))==NULL)
  {
    printf("Error in open file cfg.bat");
  }
  fputs("@set fultemp=g:\\temp\n",fp);
  fputs("@g:\\proquest\\part2.exe\n",fp);
  fclose(fp);

  system("md g:\\temp");
  system("volmap");
  system("cfg.bat");

  for(i=0;i<count;++i)          /* Finish task logout from server */
  {
   strcpy(logout,"logout ");
   strcat(logout,server[i]);
   system(logout);
  }
  system("del cfg.bat");
  system("rd g:\\temp");
 }

 /* ********************************************************************** */

 /* ********************* Main program ********************************** */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
main(int argc,char *argv[])
{
 int level;

 switch(argc){
  case 1:
   printf("Too few parameter to run this program\n\n");
   printf("USE: project [Option | Inputfile]\n\n");
   printf("[Option]\n");
   printf("  -n : Run Proquest with interact users\n");
   printf("  -r Inputfile : Redirect Inputfile option ");
   printf("in file scr_opt.dat\n");
   printf("  Inputfile : Redirect Proquest\n\n\n");
   printf(".....Press any key to continue.....\n");
   getch();
   exit(0);
  break;

  case 2:
   if(argv[1][0] == '-')
   {
        switch(argv[1][1]){  /* Use this program with interact with user */
          case 'n':
          case 'N':
             SetSystem();
          break;

          case 'r':
          case 'R':
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        printf("Too few parameter\n");
        printf("Option -r : require input file\n");
        printf("\007");
      break;

      default:
        printf("Unknown option %s\n",argv[1]);
        printf("\007"); _
      break;
    }
}
else {
      Read_data(argv[1]);
      CheckFile();      /* Send data in file to memory */

      OldInt1c = getvect(TCK);
      setvect(TCK,process);

      if(outlevel == 1)
      {
        fprintf(_fp,"Keyword                    ");
        fprintf(_fp,"                  Total\n");
        fprintf(_fp,"================================");
        fprintf(_fp,"========================================\n");
      }

      SetSystem();

      setvect(TCK,OldInt1c);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(outlevel == 1)
              fclose(_fp);
            else{
                  CreateOutFile();  /* Create complete output file */
                  system("del ~temp.$$$");
                }
          }
break;

case 3:
 if(argv[1][0] == '-')
 {
       switch(argv[1][1]){
         case 'r':
         case 'R':
           CheckFile();

           OldInt1c = getvect(TCK);
           setvect(TCK,process);

           spawnl(P_WAIT,argv[2],NULL);

           setvect(TCK,OldInt1c);
        break;

         case 'n':
         case 'N':
           printf("Too more parameter to this option\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            printf("\007");
            break;

            default:
              printf("Unknown option %s\n",argv[1]);
              printf("\007");
            break;
          }
      }
      else printf("More input file\n");
    break;

  }
}

/* ************************************************************ */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* ************************************ FILE CONVERT.H ************************************ */
                    FOR CONVERT STRING TO ASCII CODE OR SCAN CODE
                              BY .   APIRAMON   DAMRONGSIRI
                                     APISIT        SUKSAKORN
  ********************************************************************************************* */

unsigned char ScanCode[35] =
 { 0x3b,0x3c,0x3d,0x3e.0x3f,0x40,0x41,0x42,0x43,0x44,
   0x85,0x86,0x47,0x48,0x49,0x4b,0x4d,0x4f,0x50,0x51,
   0x1c,0x01,0x5d,0x5c,0x5b,0x5a,0x59,0x58,0x57,0x56,0x55,
   0x54 };

char Key[35][10] = { "@F1" ,"@F2" ,"@F3","@F4","@F5","@F6","@F7","@F8",
                     "@F9" ,"@F10","@F11","@F12","@HOME","@UP","@PGUP",
                     "@LEFT","@RIGHT","@END","@DOWN","@PGDN","@ENTER",
                     "@ESC","@SHF+F10","@SHF+F9","@SHF+F8","@SHF+F7",
                     "@SHF+F6","@SHF+F5","@SHF+F4","@SHF+F3","@SHF+F2",
                     "@SHF+F1" };

unsigned char _key_code1[65] =
 { ' ',0x20,0x39,'!',0x21,0x02,'"',0x22,0x28,'#',0x23,0x04,'$',0x24,0x05,
   '%',0x25,0x06,'&',0x26,0x08,'\'',0x27,0x28,'(',0x28,0x0a,')',0x29,0x0b,
   '*',0x2a,0x09,'+',0x2b,0x0d,',',0x2c,0x33,'-',0x2d,0x0c,'.',0x2e,0x34,
   '/',0x2f,0x35,'0',0x30,0x0b,'1',0x31,0x02,'2',0x32,0x03,'3',0x33,0x04,
   '\r',0x0d,0x1c,0xff };

unsigned char _key_code2[62] =
 { '4',0x34,0x05,'5',0x35,0x06,'6',0x36,0x07,'7',0x37,0x08,'8',0x38,0x09,
   '9',0x39,0x0a,':',0x3a,0x27,';',0x3b,0x27,'<',0x3c,0x33,'=',0x3d,0x0d,
   '>',0x3e,0x34,'?',0x3f,0x35,'@',0x40,0x03,'A',0x41,0x1e,'B',0x42,0x30,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    'C',0x43,0x2e,'D',0x44,0x20,'E',0x45,0x12,'F',0x46,0x21,'G',0x47,0x22,
    0xff };

unsigned char _key_code3[62] =
  { 'H',0x48,0x23,'I',0x49,0x17,'J',0x4a,0x24,'K',0x4b,0x25,'L',0x4c,0x26,
    'M',0x4d,0x32,'N',0x4e,0x31,'O',0x4f,0x18,'P',0x50,0x19,'Q',0x51,0x10,
    'R',0x52,0x13,'S',0x53,0x1f,'T',0x54,0x14,'U',0x55,0x16,'V',0x56,0x2f,
    'W',0x57,0x11,'X',0x58,0x2d,'Y',0x59,0x15,'Z',0x5a,0x2c,'[',0x5b,0x1a,
    0xff };

unsigned char _key_code4[62] =
  { '\\',0x5c,0x2b,']',0x5d,0x1b,'^',0x5e,0x07,'_',0x5f,0x0c,'`',0x60,0x29,
    'a',0x61,0x1e,'b',0x62,0x30,'c',0x63,0x2e,'d',0x64,0x20,'e',0x65,0x12,
    'f',0x66,0x21,'g',0x67,0x22,'h',0x68,0x23,'i',0x69,0x17,'j',0x6a,0x24,
    'k',0x6b,0x25,'l',0x6c,0x26,'m',0x6d,0x32,'n',0x6e,0x31,'o',0x6f,0x18,
    0xff };

unsigned char _key_code5[62] =
  { 'p',0x70,0x19,'q',0x71,0x10,'r',0x72,0x13,'s',0x73,0x1f,'t',0x74,0x14,
    'u',0x75,0x16,'v',0x76,0x2f,'w',0x77,0x11,'x',0x7 8,0x2d,'y',0x79,0x15,
    'z',0x7a,0x2c,'{',0x7b,0x1a,'|',0x7c,0x2b,'}',0x7d ,0x1b,'~',0x7e,0x29,
    0xff };

/* ************ Convert Ascii code to keyboard code ******************** */

_Convert_Code(char _chr,unsigned char *ascii,unsigned char *scan)
{
  int _i,_code;
  int _found=0;                          /* Check for found value */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
_i = 0;
_code = toascii(_chr);                  * Convert character to ASCII code */

if(_code >= 13 && _code <= 51)
{
        while(_key_code1[_i] != 0xff)     * 0xff is end of list */
        {
          if(_key_code1[_i] == _chr && _found == 0)
          {
                *ascii = _key_code1[++_i];          /* Put ASCII code in ascii */
                *scan = _key_code1[++_i];           * Put scan code in scan */
                ++_i;
                _found = 1;
          }                                         /* Increment index */
          else _i = _i+3;
        }
}

else if(_code >= 52 && _code <= 71)
{
        while(_key_code2[_i] != 0xff)               /* 0xff is end of list */
        {
          if(_key_code2[_i] == _chr && _found == 0)
          {
                *ascii = _key_code2[++_i];          /* Put ASCII code in ascii */
                *scan = _key_code2[++_i];           /* Put scan code in scan */
                ++_i;
                _found = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }                                   /* Increment index */
            else _i = _i+3;
        }
}

else if(_code >= 72 && _code <= 91)
{
        while(_key_code3[_i] != 0xff)           /* 0xff is end of list */
        {
          if(_key_code3[_i] == _chr && _found == 0)
          {
                *ascii = _key_code3[++_i];      /* Put ASCII code in ascii */
                *scan = _key_code3[++_i];       /* Put scan code in scan */
                ++_i;
                _found = 1;
          }                                     /* Increment index */
          else _i = _i+3;
        }
}

else if(_code >= 92 && _code <= 111)
{
        while(_key_code4[_i] != 0xff)           /* 0xff is end of list */
        {
          if(_key_code4[_i] == _chr && _found == 0)
          {
                *ascii = _key_code4[++_i];      /* Put ASCII code in ascii */
                *scan = _key_code4[++_i];       /* Put scan code in scan */
                ++_i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                _found = 1;
            }                                   /* Increment index */
            else _i = _i+3;
        }
    }

    else if(_code >= 112 && _code <= 126)
    {
        while(_key_code5[_i] != 0xff)           /* 0xff is end of list */
        {
            if(_key_code5[_i] == _chr && _found == 0)
            {
                *ascii = _key_code5[++_i];      /* Put ASCII code in ascii */
                *scan = _key_code5[++_i];       /* Put scan code in scan */
                ++_i;
                _found = 1;
            }                                   /* Increment index */
            else _i = _i+3;
        }
    }
    return;
}
/* ********************************************************************** */

/* ********** Conver code of scan key ********** */

_Scan_Code(char *_str,unsigned char *ascii,unsigned char *scan)
{
    int _i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  _i = 0;
  while((strcmp(Key[_i],_str))!=0)                    /* Detect scan code index */
  {
          ++_i;
  }
  if((strcmp(_str,"@ENTER"))==0)
    *ascii = 0x0d;
  else if((strcmp(_str,"@ESC"))==0)
              *ascii = 27;
      else *ascii = 0x00;                             /* Scan code ascii is 0x00 */

  *scan = ScanCode[_i];
  return;
}

/* ************************************************************************ */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค
เน็ตแวร์ 4.xx

ในราวกลางปี 1993 บริษัทโนเวลได้วางตลาดเน็ตแวร์รุ่นใหม่ออกมาคือ " เน็ตแวร์ 4.xx " ซึ่งมีหลักการที่แตกต่างไปจากเดิมที่ใช้ในเวอร์ชัน 3.xx ก็คือ เน็ตแวร์ 4.xx มุ่งไปยังระบบเครือข่ายขนาดใหญ่ที่ต่อร่วมกันเป็น ระบบเครือข่ายเดียวในลักษณะของส่วนกลาง (Global) หรือ ระบบเครือข่ายทั่วโลก (Worldwide Network) ซึ่งเหมาะกับบริษัทหรือผู้ใช้งานเครือข่ายที่มีขนาดใหญ่ ที่ต้องการเชื่อมต่อเครื่องบริการเป็นจำนวนสิบ ๆ ตัวเข้าด้วยกัน เน็ตแวร์ 4.xx จะช่วยให้การใช้งานโดยรวมทั้งหมดสะดวกขึ้น แต่ก็แน่นอนว่าประสิทธิภาพในและความเร็วในการใช้งานย่อมลดลงเนื่องจากระบบปฏิบัติการต้องดูแลระบบเครือข่ายที่มีขนาดใหญ่มากขึ้น สำหรับผู้ใช้เน็ตแวร์ขนาดกลางและขนาดเล็กที่มีจำนวนของเครื่องบริการไม่กี่ตัวต่อเชื่อมกันก็ควรใช้ เน็ตแวร์ 3.xx ทั้งนี้เนื่องจากไม่มีความจำเป็นที่ต้องทำให้เครือข่ายทั้งหมดรวมเป็นหนึ่งเดียว และความเร็วในการใช้งานของเวอร์ชัน 3.xx ก็เร็วกว่าด้วย

การบีบข้อมูล

สิ่งที่เพิ่มเข้ามาใน เน็ตแวร์ 4.xx อีกอย่างหนึ่งก็คือระบบการจัดการเกี่ยวกับเนื้อที่บนจานแข็งซึ่งทำให้มีเนื้อที่เพิ่มขึ้นโดยการลดขนาดข้อมูลก่อนเก็บลงจานแข็งโดยใช้เทคนิคการบีบข้อมูลในลักษณะแฟ้มต่อแฟ้ม (File by File Compression) ซึ่งจะลดขนาดของแฟ้มข้อมูลลงได้ประมาณครึ่งหนึ่งของขนาดเดิม จานแข็งจึงสามารถเก็บข้อมูลได้เพิ่มขึ้นถึงสองเท่า นอกจากนี้ยังสามารถกำหนดได้ว่า จะให้มีการบีบข้อมูลในแฟ้มใด เพื่อให้แฟ้มที่ใช้บ่อย ไม่ต้องถูกลดขนาดให้เสียเวลา นอกจากนี้แฟ้มที่ไม่ค่อยได้เรียกใช้จะถูก เน็ตแวร์ 4.xx ย้ายออกจากจานแข็งไปเก็บลงในเทปหรืออุปกรณ์เก็บข้อมูลสำรองอื่น ๆ เพื่อให้จานแข็งมีเนื้อที่มากขึ้น และเมื่อมีการเรียกใช้แฟ้มเหล่านั้น เน็ตแวร์ 4.xx จะทำการย้ายแฟ้มจากเทปหรืออุปกรณ์เก็บข้อมูลสำรองกลับมาในจานแข็งดังเดิมโดยอัตโนมัติ ทำให้การใช้เนื้อที่บนจานแข็งเดประโยชน์สูงสุด

การจัดการหน่วยความจำ

การจัดหน่วยความจำของ เน็ตแวร์ 4.xx ก็ปรับปรุงให้ขึ้นจาก เวอร์ชัน 3.xx คือมอดูลต่าง ๆ เมื่อเลิกใช้งานแล้ว สามารถคืนหน่วยความจำนั้นกลับให้เครื่องบริการนำไปใช้อย่างอื่นต่อไป ซึ่งใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เน็ตแวร์ 3.xx จะทำได้ยากกว่า นอกจากนี้ยังสามารถเลือกให้มอดูลระบบของ เน็ตแวร์ (NLM) ทำงานในชั้นต่าง ๆ ของหน่วยความจำได้ โดยหน่วยความจำในวงแหวนชั้นในสุด หรือ Ring 0 จะเป็นชั้นที่มีความสามารถสูงสุด แต่มีระบบการป้องกันต่ำ คือโปรแกรมสามารถทำอะไรก็ได้ หน่วยความจำในวงถัดออกมาคือ Ring 1, 2 และ 3 จะมีความสามารถต่ำกว่าแต่มีระบบการป้องกันสูงขึ้น การเลือกให้ NLM ใดทำงานในหน่วยความจำชั้นไหนขึ้นอยู่กับผู้ดูแลเน็ตเวิร์กนั้น ๆ

คำสั่งใน เน็ตแวร์ 4.xx

คำสั่งต่าง ๆ ใน เน็ตแวร์ 4.xx จะมีบางส่วนเพิ่มเข้ามาใหม่และบางส่วนเปลี่ยนแปลงไปจากของเดิมเล็กน้อยเมื่อเทียบกับเวอร์ชัน 3.12 ซึ่งพอจะสรุปได้ดังนี้คือ

คำสั่งอรรถประโยชน์ของเครื่องบริการ

ABORT REMIRROR	เป็นโปรแกรมอรรถประโยชน์ของเครื่องบริการ ใช้สำหรับหยุดการทำงานควบคู่กัน (Mirrored) ของจานแข็งสองตัวที่กำลังคัดลอกข้อมูลให้ตรงกัน
KEYB	เป็นโปรแกรมอรรถประโยชน์ของเครื่องบริการสำหรับเลือกแป้นพิมพ์ภาษาต่าง ๆ เช่น อังกฤษ, ฝรั่งเศส, เยอรมัน บนแป้นพิมพ์
LANGUAGE	ใช้กำหนดภาษาที่ NLM มอดูลต่าง ๆ จะใช้แสดงผลและข้อความบนจอภาพ
LIST DEVICES	ใช้แสดงรายละเอียดของอุปกรณ์ต่าง ๆ ที่ต่ออยู่กับเครื่องบริการ
MAGAZINE	ใช้ยืนยันการทำงานของคำสั่ง Insert Magazine และ Remove Magazine บนเครื่องบริการว่ามีผลเป็นอย่างไร
MEDIA	ใช้ยืนยันการทำงานของคำสั่ง Insert Media และ Remove Media บนเครื่องบริการว่ามีผลเป็นอย่างไร
MIRROR STATUS	ใช้แสดงจานแข็งทั้งหมดที่มีการทำงานควบคู่กัน(Mirrored Disk) ตลอดจนสภาวะของมัน
REMIRROR PARTITION	ใช้สำหรับจานแข็งส่วนที่เคยกำหนดให้ทำงานแบบการทำงานควบคู่กันไว้นั้น เริ่มมีการทำงานควบคู่กันใหม่
SCAN FOR NEW DEVICE	ใช้สำหรับตรวจสอบอุปกรณ์โดยเฉพาะจานแข็ง ว่ามีส่วนใดเพิ่มเติมเข้ามาในระบบบ้างหลังจากเครื่องบริการทำงานครั้งที่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งแบบ NLM ของเครื่องบริการ

CDROM	เป็นมอดูลแบบ NLM บนเครื่องบริการสำหรับเรียกใช้ ซีดีรอม โดยมอง ซีดีรอม เป็น หมวด แบบที่อ่านข้อมูลได้อย่างเดียว
DOMAIN	เป็นมอดูลสำหรับสร้างระบบปฏิบัติการโดเมนและกำหนดการทำงานให้อยู่ในหน่วยความจำที่ 1, 2 หรือ 3 ได้ เพื่อให้ NLM ที่พัฒนาโดยบริษัทอื่น ๆ ทำงานได้โดยไม่มีผลกระทบต่อระบบปฏิบัติการ
DSMERGE	เป็นมอดูลสำหรับรวมไดเรกทอรีของเน็ตแวร์ สองระบบเข้าเป็นไดเรกทอรีเดียวกัน
DSREPAIR	เป็นมอดูลสำหรับแก้ไขและซ่อมแซมไดเรกทอรีของเน็ตแวร์
NPRINTER	เป็นมอดูลสำหรับกำหนดให้เครื่องพิมพ์ที่ต่ออยู่กับเครื่องบริการใดก็ตาม ทำหน้าที่เป็นเครื่องพิมพ์รวมของเครือข่าย
NUT	เป็นมอดูลสำหรับใช้งานร่วมกับ NLM เวอร์ชัน 3.11 ที่มีการเรียกใช้ฟังก์ของ NUT
NWSNUT	เป็นมอดูลสำหรับจัดเตรียมส่วนเชื่อมต่อผู้ใช้งานเข้ากับ NLM มอดูลต่างๆ เช่น MONITOR และ SERVMAN
PUPGRADE	เป็นมอดูลสำหรับปรับเวอร์ชันการทำงานเกี่ยวกับเครื่องพิมพ์, การใช้คำสั่งพิมพ์ในเครือข่าย และการกำหนดคุณสมบัติของเครื่องพิมพ์
RPL	เป็นมอดูลสำหรับบูตเครื่องพีซี ที่ไม่มีจานบันทึก (disk) ติดตั้งอยู่ในเครื่อง ให้ใช้งานเครือข่ายได้โดยบูตผ่านการ์ดแลนในเครื่อง
RTDM	เป็นมอดูลสำหรับโอนย้ายไฟล์ที่ไม่ได้ใช้งานไปเก็บในอุปกรณ์เก็บข้อมูลสำรอง
SBACKUP	เป็นมอดูลที่ใช้เก็บข้อมูลสำรองในเครือข่าย ซึ่งเราสามารถสั่งเก็บข้อมูลสำรองจากที่ไหนในเครือข่ายก็ได้
SERVMAN	เป็นมอดูลที่ใช้ตั้งค่าพารามิเตอร์ต่างๆ ในไฟล์ NCF ของเครือข่าย รวมทั้งเก็บข้อมูลเกี่ยวกับข้อมูลต่างๆ , หมวด และรายละเอียดของเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สามารถสำเร็จลงได้ด้วยดี ก็เนื่องมาจากได้รับความช่วยเหลือและคำแนะนำในการทำงานจากบุคคลต่าง ๆ ดังนั้นคณะผู้จัดทำจึงขอแสดงความขอบคุณ

- อาจารย์เอื้อน ปิ่นเงิน อาจารย์ที่ปรึกษา
- อาจารย์อัครินทร์ คุณกิตติ อาจารย์ภาควิชาวิศวกรรมอิเล็กทรอนิกส์
- อาจารย์เทอดศักดิ์ ลิ่วหาทอง อาจารย์ภาควิชาวิศวกรรมอิเล็กทรอนิกส์
- อาจารย์เสถียรพงษ์ ทุยะนันท์ อาจารย์จากมหาวิทยาลัยเทคโนโลยีมหานคร
- อาจารย์สมชาย ผู้ช่วยเหลือในด้านเครื่องบริการซีดีรอม

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

ก. เอกสารอ้างอิงที่เป็นวารสารภาษาอังกฤษ จัดเรียงลำดับดังนี้

1. Ed Perratore, " NETWORKING CD-ROMS : The Power of Shared Access ", PC MAGAZINE, December 31,1991, pp 333-363
2. James Karney, " Sharing a CD-ROM", PC MAGAZINE, February 22, 1994, pp 171-178

ข. เอกสารอ้างอิงที่เป็นหนังสือภาษาไทยและภาษาอังกฤษ จัดเรียงลำดับดังนี้

1. วิศวกรรมสถานแห่งประเทศไทย ในพระบรมราชูปถัมป์, " ศัพท์เทคนิควิศวกรรมคอมพิวเตอร์ ", โรงพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 155 หน้า, 2537
2. ต้น ตันท์สุทธิ์วงศ์ และ สุพจน์ ปุณณชัยยะ, " คู่มือการใช้งาน NetWare ฉบับสมบูรณ์ เวอร์ชั่น 3.12 และ 4.x ", บริษัท โปรวิชั่น จำกัด, 501 หน้า, 2537
3. ดร. ดวงแก้ว สวามิภักดิ์, " เข้าใจการทำงาน IBM PC Inside the IBM PC and PS/2 โดย ปีเตอร์ นอร์ตัน ", บริษัทซีเอ็ดยูเคชั่นจำกัด, 264 หน้า, 2534
4. ธันวา ศรีประโมง, " การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม ", มหาวิทยาลัยเทคโนโลยีมหานคร, 739 หน้า, 2537
5. M. Sandra Wood, " CD-ROM Implementtation and Networking in Health Sciences Libraries ", The Haworth Press,Inc., 160 p.
6. Marry Kay Duggan, " CD-ROM IN THE LIBRARY ", G.K. Hall & Co.,126 p., 1990
7. 6. Novel, Inc., " NetWare SYSTEM INTERFACE TECHNICAL OVERVIEW ", Novel Incorported, 333 p., 1989
8. Novel, Inc., " NetWare C Interface-DOS Volume I ", Novel Incorported, 622 p., 1990
9. Novel, Inc., " NetWare C Interface-DOS Volume II ", Novel Incorported, 622 p., 1990

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. Charles G. Rose, " Programmer's Guide to NetWare ", McGraw-Hill, Inc., 1990

11. Norman Desmarais, " CD-ROM Local Area networks : A User's Guid ",Meckler, 130 p., 1991

12. Michael Tischer, " PC INTERN SYSTEM PROGRAMMING The Encyclopedia of DOS Programming Know How ", Library of Congress Cataloging-in-Publication Data, 1320 p., 1992

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้