

บทที่ 4

การพัฒนาระบบ

อุปกรณ์ที่ใช้ในการพัฒนาระบบ

ดังที่ได้กล่าวไว้ในบทที่ 3 โปรแกรมที่จะพัฒนาขึ้นสำหรับระบบนี้ ประกอบไปด้วย 5 โปรแกรม คือโปรแกรมกำหนดโครงสร้างข้อมูล โปรแกรมกำหนดรูปแบบรายงานทางจอภาพโปรแกรมแสดงผลทางหน้าจอ โปรแกรมสร้างโปรแกรมภาษาโคบอล และ โปรแกรมจัดการแฟ้มข้อมูล ซึ่งการพัฒนาโปรแกรมทั้ง 5 นี้ผู้วิจัยได้ทำการพัฒนาบนเครื่องไมโครคอมพิวเตอร์ 16 บิตโดยมีคุณลักษณะของฮาร์ดแวร์ และ ซอฟต์แวร์ที่ใช้ดังนี้

1. คุณลักษณะของฮาร์ดแวร์

- ไมโครโพรเซสเซอร์ เบอร์ 8088
- หน่วยความจำ 640 กิโลไบต์
- หน่วยความจำสำรอง

โดยมี แผ่นบันทึกข้อมูล ความจุ 1 x 1.2 เมกะไบต์

และ 1 x 360 กิโลไบต์

- จอภาพ VGA

2. ซอฟต์แวร์

- Dos Version 3.0
- Turbo C Version 2.0
- Assembly 8086/8088
- Microsoft COBOL Version 3.0

รายละเอียดของโปรแกรมที่พัฒนา

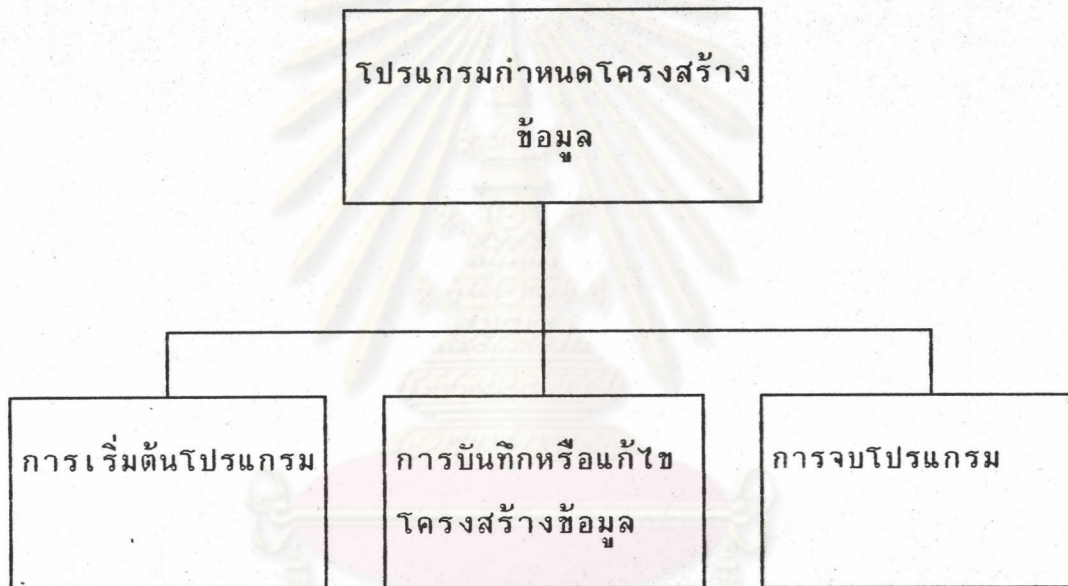
1. โปรแกรมกำหนดโครงสร้างแฟ้มข้อมูล

1.1 หน้าทีของโปรแกรม ทำการสร้างและแก้ไขหรือปรับปรุงโครงสร้างแฟ้มข้อมูล ของแฟ้มข้อมูลที่ใช้เป็นข้อมูลเข้าประเภท Btrieve ISAM และ Text

1.2 ข้อมูลเข้า ชื่อแฟ้มข้อมูล และโครงสร้างของแฟ้มข้อมูลนั้น

1.3 ผลลัพธ์ แฟ้มข้อมูลที่เก็บบันทึกโครงสร้างของแฟ้มข้อมูล

1.4 ขั้นตอนการประมวลผล การประมวลผลประกอบด้วย 3 ขั้นตอนดังแสดงในภาพที่ 4.1 ซึ่งมีรายละเอียดดังนี้



รูปที่ 4.1 ผังโครงสร้างโปรแกรมกำหนดโครงสร้างแฟ้มข้อมูล

1.4.1 การเริ่มต้นโปรแกรม ประกอบด้วย

- การรับชื่อแฟ้มข้อมูล
- การแสดงข้อความ คำสั่ง สัญลักษณ์ต่าง ๆ

ที่ใช้ในโปรแกรมนี้

- กรณีที่แฟ้มข้อมูลนั้นมีอยู่แล้วให้อ่านข้อมูลนั้นมา

แสดงบนจอภาพ

1.4.2 การบันทึกหรือแก้ไขโครงสร้างข้อมูลประกอบด้วย

- การบันทึกหรือแก้ไข ชื่อเขตข้อมูล

- การบันทึกหรือแก้ไข ประเภทของข้อมูล ซึ่งมี
2 ประเภทคือ N หมายถึงข้อมูลประเภทตัวเลข C หมายถึงข้อมูลประเภทอักษร

- การบันทึกหรือแก้ไข ความยาวของเขตข้อมูล
- การบันทึกหรือแก้ไข จำนวนตัวเลขหลังจุด

ทศนิยม

- การบันทึกหรือแก้ไข เขตข้อมูลที่เป็นคีย์หรือไม่เป็นคีย์ โดย Y หมายถึงเขตข้อมูลนั้นเป็นคีย์ และ N หมายถึงเขตข้อมูลนั้นไม่เป็นคีย์

- การบันทึกหรือแก้ไข หมายเลขคีย์

ในการบันทึกหรือการแก้ไขข้อมูลนี้ ผู้ใช้สามารถใช้ปุ่มต่าง ๆ บนแป้นพิมพ์เพื่ออำนวยความสะดวกในการบันทึกหรือการแก้ไข ดังแสดงไว้ในตารางที่ 4.1

ปุ่มคำสั่ง	หน้าที่
Ctrl+N	แทรกบรรทัด 1 บรรทัด
Ctrl+Y	ลบบรรทัด 1 บรรทัด
Pgup	เลื่อนเคอร์เซอร์ขึ้น 1 หน้า
Pgdn	เลื่อนเคอร์เซอร์ลง 1 หน้า
Up Arrow Key	เลื่อนเคอร์เซอร์ขึ้น 1 บรรทัด
Down Arrow Key	เลื่อนเคอร์เซอร์ลง 1 บรรทัด
Del	ลบตัวอักษร ณ ตำแหน่งของเคอร์เซอร์ 1 ตัว
Bksp	ลบตัวอักษร ณ ตำแหน่งก่อนหน้าของเคอร์เซอร์ 1 ตัว
Right Arrow Key	จะทำการเลื่อนเคอร์เซอร์ไปทางขวา 1 ตัว
Left Arrow Key	เลื่อนเคอร์เซอร์ไปทางซ้าย 1 ตัว

ตารางที่ 4.1 ปุ่มคำสั่งต่าง ๆ ที่ใช้ในการแก้ไขปรับปรุงข้อมูลของแฟ้มโครงสร้างข้อมูล

1.4.2 การจบโปรแกรม ผู้ใช้สามารถออกจากโปรแกรม

ได้ 2 วิธีคือ

- กดปุ่ม Ctrl + End หรือ กด Enter ณ จุดเริ่มต้นของเขตข้อมูลที่ว่างเพื่อให้โปรแกรมจัดเก็บข้อมูลต่าง ๆ ลงแฟ้มข้อมูล ก่อนที่จะจบโปรแกรม

- กดปุ่ม Esc เพื่อออกจากโปรแกรมโดยไม่มี การจัดเก็บข้อมูล

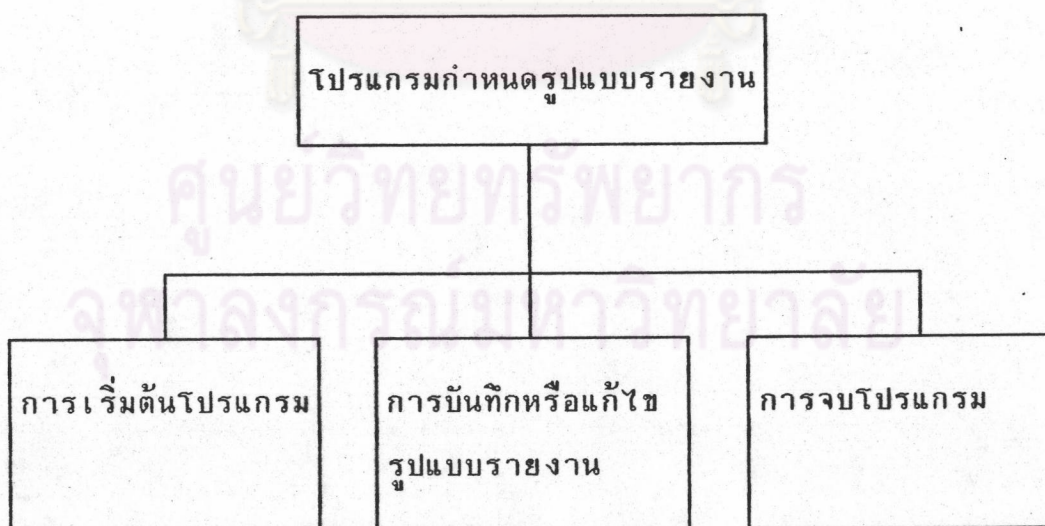
2. โปรแกรมกำหนดรูปแบบรายงาน

2.1 หน้าที่ของโปรแกรม รับรูปแบบรายงานที่ผู้ใช้กำหนดชั้นทางจอภาพแล้วเก็บไว้ในแฟ้มข้อมูลรูปแบบรายงาน และสร้างโปรแกรมภาษาระดับที่ 4

2.2 ข้อมูลเข้า ชื่อแฟ้มข้อมูลรูปแบบรายงาน ชื่อแฟ้มข้อมูลภาษาระดับที่ 4 และ แฟ้มข้อมูลต่าง ๆ ที่ต้องการพิมพ์รายงาน

2.3 ผลลัพธ์ แฟ้มข้อมูลรูปแบบรายงานและแฟ้มข้อมูลภาษาระดับที่ 4

2.4 ขั้นตอนการประมวลผล ประกอบไปด้วย 3 ขั้นตอนดังแสดงในรูปที่ 4.2



รูปที่ 4.2 ผังโครงสร้างโปรแกรมการกำหนดรูปแบบรายงาน

2.4.1 การเริ่มต้นโปรแกรม ประกอบด้วย

- การรับชื่อแฟ้มข้อมูลรูปแบบรายงาน และ ชื่อแฟ้มข้อมูลภาษาระดับที่ 4
- การแสดงข้อความ คำสั่ง และ สัญลักษณ์ต่าง ๆ ที่ใช้ในโปรแกรมนี้อ่านข้อมูลขึ้นมาแสดงผลทางจอภาพ
- กรณีที่ไม่มีแฟ้มข้อมูลรูปแบบรายงานอยู่ก่อนโปรแกรมจะรอรับข้อมูลต่าง ๆ คือ ประเภทของแฟ้มข้อมูลที่จะพิมพ์รายงาน สารบบของแฟ้มข้อมูลหลัก ชื่อแฟ้มข้อมูลหลัก เงื่อนไขที่จะใช้ในการนำข้อมูลมาสร้างรายงาน จำนวนบรรทัดต่อหน้า จำนวนตัวอักษรต่อบรรทัด

2.4.2 การบันทึกหรือแก้ไขรูปแบบรายงาน

ในการบันทึกหรือแก้ไขข้อมูลนี้ ผู้ใช้สามารถพิมพ์ปุ่มคำสั่งต่าง ๆ บนแป้นพิมพ์เพื่ออำนวยความสะดวกในการบันทึกหรือแก้ไข ดังแสดงไว้ในตารางที่ 4.2

ปุ่มฟังก์ชัน	ข้อมูลต่าง ๆ ที่รับเข้ามา
F1	ประเภทของแฟ้มข้อมูล สารบบของแฟ้มข้อมูลหลัก ชื่อแฟ้มข้อมูลหลัก เงื่อนไขที่ใช้ในการนำข้อมูลมาสร้างรายงาน จำนวนบรรทัดต่อหน้า จำนวนตัวอักษรต่อบรรทัด
F2	คุณลักษณะของข้อมูล

ตารางที่ 4.2 ปุ่มคำสั่งต่างๆ ที่ใช้ในการบันทึกหรือแก้ไขข้อมูล

ปุ่มฟังก์ชัน	ข้อมูลต่าง ๆ ที่รับเข้ามา
F3	สารบบของแฟ้มข้อมูลอื่น ๆ ชื่อแฟ้มข้อมูลและชื่อเขตข้อมูล คุณลักษณะของข้อมูล รูปแบบในการพิมพ์ของข้อมูล สูตรที่ใช้ในการคำนวณ
F4	ลบข้อมูลที่ได้มาจากปุ่มฟังก์ชัน F3
F5	ตำแหน่งของระดับต่าง ๆ
F6	ชื่อแฟ้มข้อมูลดัชนี
F7	ความสัมพันธ์ของแฟ้มข้อมูลต่าง ๆ
F10	การเปลี่ยน โหมด ระหว่างภาษาไทยและภาษาอังกฤษ
Left Arrow Key	เลื่อนเคอร์เซอร์มาทางด้านซ้าย 1 ตัว
Right Arrow Key	เลื่อนเคอร์เซอร์มาทางด้านขวา 1 ตัว
Home Key	เลื่อนเคอร์เซอร์ มายังตำแหน่งแรกของบรรทัดปัจจุบัน

ตารางที่ 4.2 ปุ่มคำสั่งต่างๆ ที่ใช้ในการบันทึกหรือแก้ไขข้อมูล (ต่อ)

- 2.4.3 การจบโปรแกรม ผู้ใช้สามารถออกจากโปรแกรมได้ 3 ลักษณะคือ
- กดปุ่ม F8 เพื่อให้โปรแกรมบันทึกข้อมูลลงแฟ้มข้อมูลโปรแกรมภาษาระดับที่ 4 และแฟ้มข้อมูลกำหนดรูปแบบรายงาน
 - กดปุ่ม F9 เพื่อให้โปรแกรมบันทึกข้อมูลลงแฟ้มข้อมูลกำหนดรูปแบบรายงานเพียงแฟ้มเดียว
 - กดปุ่ม Esc เพื่อออกจากโปรแกรมโดยไม่มี
- การจัดเก็บข้อมูล

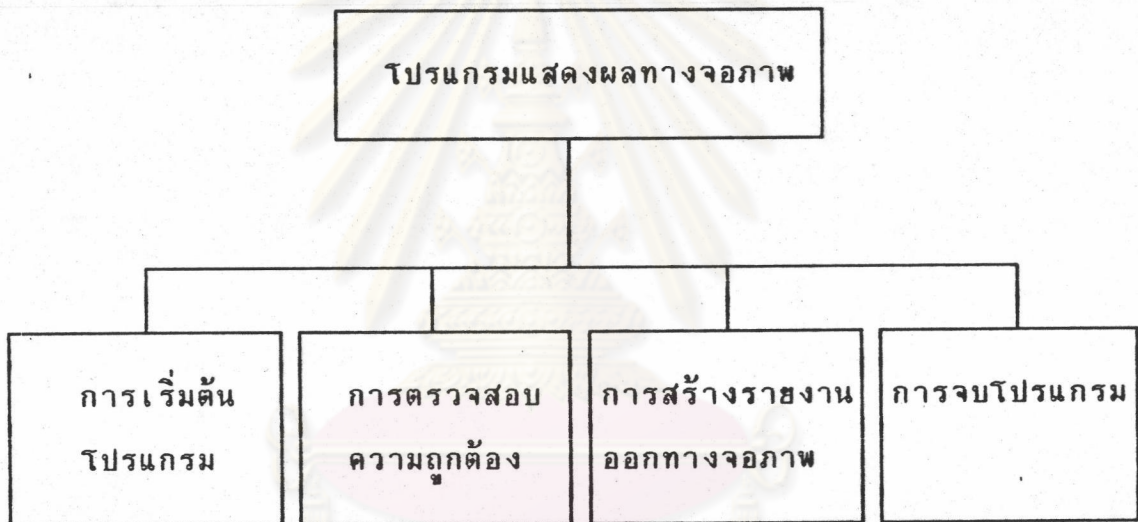
3. โปรแกรมแสดงผลทางจอภาพ

3.1 หน้าแรกของโปรแกรม จะอ่านข้อมูลจากแฟ้มข้อมูลโปรแกรม ภาษาระดับที่ 4 แฟ้มโครงสร้างข้อมูล และ แฟ้มข้อมูลต่าง ๆ ที่ต้องการพิมพ์ รายงาน มาทำการประมวลผลเพื่อออกรายงานให้ผู้ใช้ได้เห็นรูปแบบโดยสังเขป ทางจอภาพ โดยไม่แสดงส่วนของรายงานที่เกินคอลัมน์ที่ 80

3.2 ข้อมูลเข้า แฟ้มข้อมูลภาษาระดับที่ 4 แฟ้มโครงสร้างข้อมูล และ แฟ้มข้อมูลต่าง ๆ ที่ต้องการพิมพ์รายงาน

3.3 ผลลัพธ์ รายงานทางหน้าจอ

3.4 ขั้นตอนการประมวลผลประกอบด้วยขั้นตอนต่าง ๆ ดังแสดง ในภาพที่ 4.3



ภาพที่ 4.3 ผังโครงสร้างโปรแกรมแสดงผลทางจอภาพ

3.4.1 การเริ่มต้นโปรแกรม ประกอบด้วย

- การรับชื่อโปรแกรมภาษาระดับที่ 4
- การเตรียมจอภาพให้พร้อม สำหรับการสร้าง รายงาน

3.4.2 การตรวจสอบความถูกต้อง ของคำสั่งโปรแกรม ภาษาระดับที่ 4 ประกอบด้วย

- การอ่านคำสั่งโปรแกรมภาษาระดับที่ 4 มา ทีละคำสั่ง

- ตรวจสอบคำสิ่งที่อ่านเข้ามาว่า ถูกต้องตามหลักไวยากรณ์หรือไม่ กรณีที่มีความผิดพลาดเกิดขึ้นจะจัดเก็บข้อมูลแสดงความผิดพลาดนั้นเก็บไว้ และ จะแสดงข้อความผิดพลาดทั้งหมดทางจอภาพหลังจากอ่านคำสิ่งของโปรแกรมภาษาระดับที่ 4 มาจนหมดแล้ว และจะทำการจบโปรแกรมโดยไม่มีการสร้างรายงานออกทางจอภาพ กรณีที่ไม่มีข้อผิดพลาดเกิดขึ้นจะทำงานขั้นตอนถัดไป

3.4.3 การสร้างรายงานแสดงผลทางจอภาพ ประกอบไปด้วยขั้นตอนต่าง ๆ ดังต่อไปนี้

ก) การเริ่มต้นของการสร้างรายงาน ออกทางจอภาพ ซึ่งสามารถแบ่งออกเป็นขั้นตอนต่าง ๆ ดังต่อไปนี้

1) เปิดแฟ้มข้อมูลทุกแฟ้ม
2) อ่านข้อมูล จากแฟ้มข้อมูลหลักมา 1 ระเบียบ ถ้าแฟ้มข้อมูลหลักมีความสัมพันธ์กับแฟ้มอื่น ๆ ก็จะทำกรอ่านข้อมูลจากแฟ้มข้อมูลอื่น ๆ มาด้วย

3) ตรวจสอบดูว่า มีคำสิ่งเกี่ยวกับการคำนวณหรือไม่ ถ้ามีนำข้อมูลจากขั้นตอนที่ผ่านมาไปทำการคำนวณหาผลลัพธ์

4) ทำการพิมพ์หัวชุดรายงาน หัวรายงาน หัวรายการ และ หัวรายงานย่อย ตามตำแหน่ง รูปแบบ และ คุณลักษณะที่กำหนด

ข) การประมวลผล ของการสร้างรายงานออกทางจอภาพ จะวนตามขั้นตอนที่ 1 ถึง 5 ต่อไปนี้ จนกว่าจะหมดข้อมูลจากแฟ้มข้อมูลหลัก

1) ทำการพิมพ์รายการข้อมูล
2) เพิ่มตัวนับบรรทัด ถ้าตัวนับบรรทัดเท่ากับหรือมากกว่าจำนวนบรรทัดที่กำหนดในคำสั่ง Lineperpage ให้ทำการขึ้นหน้าใหม่

3) ถ้าเกิดการเปลี่ยนค่าในเขตข้อมูล ให้พิมพ์ท้ายรายงานย่อย และ หัวรายงานย่อย

4) อ่านข้อมูล จากแฟ้มข้อมูลหลักมา 1 ระเบียบ ถ้าแฟ้มข้อมูลหลักมีความสัมพันธ์กับแฟ้มอื่น ๆ ก็จะทำกรอ่านข้อมูลจาก

แฟ้มข้อมูลอื่น ๆ มาด้วย

5) ตรวจสอบดูว่า มีคำสั่งเกี่ยวกับการ
คำนวณหรือไม่ ถ้ามีนำข้อมูลจากขั้นตอนที่ผ่านมาไปทำการคำนวณหาผลลัพธ์

6) ถ้าจบแฟ้มข้อมูลหลักแล้ว จะทำขั้นตอน
ต่อไปนี้

- ทำการพิมพ์ท้ายรายงานย่อย ท้าย
รายงาน และท้ายชุดรายงาน

- ปิดแฟ้มข้อมูลทุกแฟ้ม

3.4.4 การจบโปรแกรม จะทำการเปลี่ยนโหมดของ
จอภาพให้กลับไปยังโหมดเดิม

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

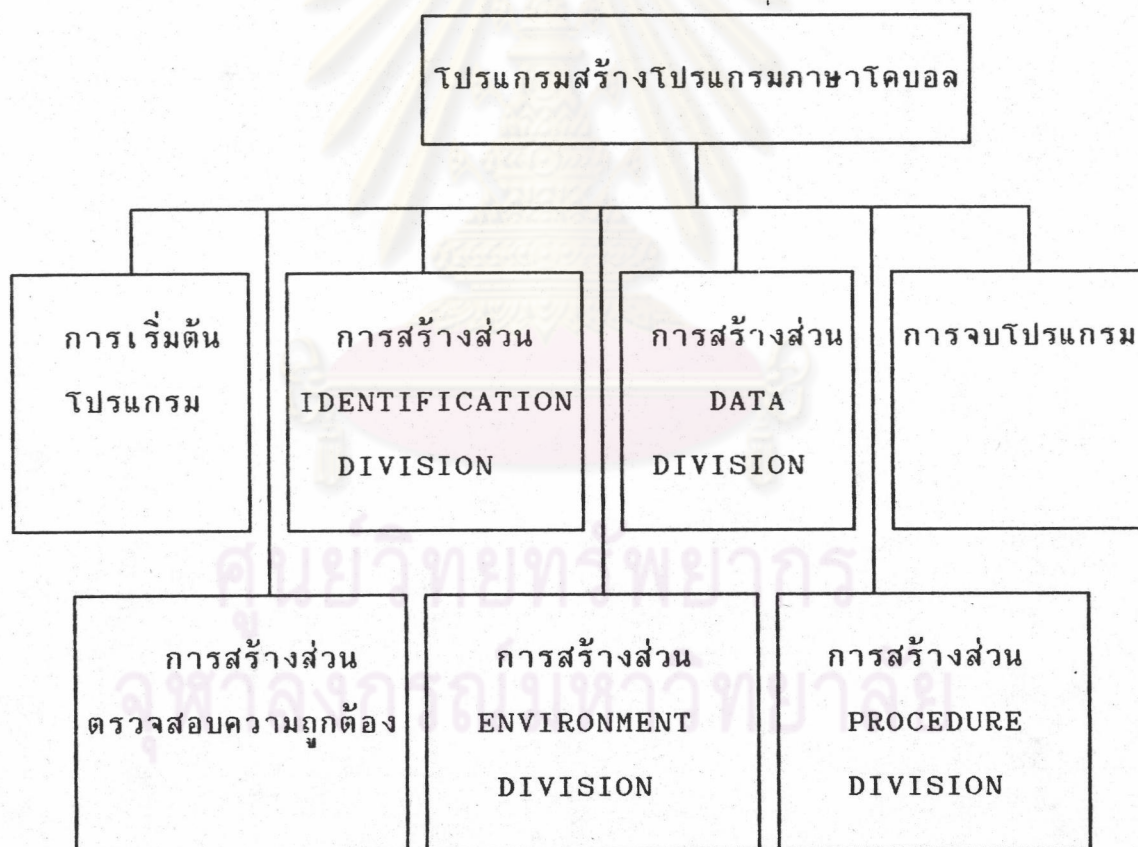
4. โปรแกรมสร้างโปรแกรมภาษาโคบอล

4.1 หน้าทีของโปรแกรม อ่านข้อมูลจาก แฟ้มข้อมูล โปรแกรมภาษาระดับที่ 4 แฟ้มโครงสร้างข้อมูล และ แฟ้มข้อมูลต่าง ๆ ที่ต้องการพิมพ์รายงาน มาทำการประมวลผล เพื่อสร้างโปรแกรมภาษาโคบอล

4.2 ข้อมูลเข้า ชื่อแฟ้มข้อมูลภาษาระดับที่ 4 ชื่อแฟ้มข้อมูลภาษาโคบอล ชื่อแฟ้มข้อมูลที่จะจัดเก็บผลลัพธ์ที่ได้จากโปรแกรมภาษาโคบอล แฟ้มโครงสร้างข้อมูล แฟ้มข้อมูลที่ต้องการพิมพ์รายงาน

4.3 ผลลัพธ์ แฟ้มข้อมูลภาษาโคบอล

4.4 ขั้นตอนการประมวลผลประกอบด้วยขั้นตอนต่าง ๆ ดังแสดงในรูปที่ 4.4



รูปที่ 4.4 ผังโครงสร้างโปรแกรมสร้างภาษาโคบอล

4.4.1 การเริ่มต้นโปรแกรม ประกอบไปด้วย

- การรับชื่อแฟ้มข้อมูล
- การเปิดแฟ้มข้อมูล

4.4.2 การตรวจสอบความถูกต้อง ของคำสั่งโปรแกรม ภาษาระดับที่ 4 ประกอบด้วย

- การอ่านคำสั่งโปรแกรมภาษาระดับที่ 4 มาที่
ละคำสั่ง

- ตรวจสอบคำสั่งนั้นว่าถูกต้องหรือไม่ กรณีที่มี
ความผิดพลาดเกิดขึ้นจะจัดเก็บข้อมูลแสดงความผิดพลาดนั้นเก็บไว้ และ จะแสดง
ข้อความผิดพลาดทั้งหมดผลทางจอภาพหลังจากอ่านคำสั่งโปรแกรมภาษาระดับที่ 4
มาจนหมดแล้ว และจะทำการจบโปรแกรมโดยไม่มีการสร้างโปรแกรมภาษาโคบอล
กรณีที่ไม่มีข้อผิดพลาดเกิดขึ้นจะทำขั้นตอนถัดไป

ก่อนที่จะกล่าวถึงขั้นตอนถัดไป จะขอแสดงสัญลักษณ์
ที่ใช้ในการสร้างโปรแกรมภาษาโคบอล ซึ่งมีรายละเอียดดังปรากฏในตารางที่ 4.3

สัญลักษณ์	ความหมาย
กลุ่มตัวอักษรตัวเต็ม กลุ่มตัวอักษรปกติ [...]	กลุ่มตัวอักษรที่เป็นส่วนหนึ่งของภาษาโคบอลที่ถูกสร้างขึ้น พารามิเตอร์ต่าง ๆ ทางเลือก จะมีหรือไม่มีก็ได้ การแบ่งแยกข้อความเพื่อความสะดวกต่อการอ่าน

ตารางที่ 4.3 สัญลักษณ์ที่ใช้ในการสร้างโปรแกรมภาษาโคบอล

จุฬาลงกรณ์มหาวิทยาลัย



4.4.3 การสร้างส่วน IDENTIFICATION DIVISION

โปรแกรมจะสร้างคำสั่งต่าง ๆ ต่อไปนี้

IDENTIFICATION DIVISION.

PROGRAM-ID. program-name.

DATE-WRITTEN. system-date.

โดยที่ program-name ได้จากชื่อแฟ้มข้อมูลภาษาโคบอล

system-date ได้จากระบบคอมพิวเตอร์

4.4.4 การสร้างส่วน ENVIRONMENT DIVISION

โปรแกรมจะสร้างคำสั่งต่าง ๆ ต่อไปนี้

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

ก) สำหรับแฟ้มข้อมูลประเภท ISAM

- สำหรับแฟ้มข้อมูลหลัก จะสร้างคำสั่งต่าง ๆ ต่อไปนี้

SELECT INPUT-FILE ASSIGN TO DISK

ORGANIZATION IS INDEXED

ACCESS MODE IS SEQUENTIAL.

RECORD KEY IS KEYINO = fieldnameINO [fieldnameINO].

โดยที่ filenumber หมายถึง หมายเลขแฟ้มข้อมูล

fieldname หมายถึง ชื่อเขตข้อมูล

- สำหรับแฟ้มข้อมูลอื่น ๆ จะสร้างคำสั่งต่อไปนี้อย่างไรตามจำนวนแฟ้มข้อมูลทั้งหมด - 1

```
SELECT INPUT-FILEfilename ASSIGN TO DISK
      ORGANIZATION IS INDEXED
      ACCESS MODE IS RANDOM
      RECORD KEY IS KEYINO = fieldnameINO [fieldnameINO]
```

โดยที่ filename หมายถึง หมายเลขแฟ้มข้อมูล
 fieldname หมายถึง ชื่อเขตข้อมูล

ข) สำหรับแฟ้มข้อมูล Text ประเภท COBOL Variable Length Record จะวนสร้างคำสั่งต่อไปนี้อย่างไรตามจำนวนแฟ้มข้อมูลทั้งหมด

```
SELECT INPUT-FILEfilename ASSIGN TO DISK
      ACCESS MODE IS SEQUENTIAL
      ORGANIZATION IS LINE SEQUENTIAL.
```

โดยที่ filename หมายถึง หมายเลขแฟ้มข้อมูล

ค) สำหรับแฟ้มข้อมูล Text ประเภท COBOL Fixed Length Record จะวนสร้างคำสั่งต่อไปนี้อย่างไรตามจำนวนแฟ้มข้อมูลทั้งหมด

```
SELECT INPUT-FILEfilename ASSIGN TO DISK
      ACCESS MODE IS SEQUENTIAL
      ORGANIZATION IS SEQUENTIAL
```

โดยที่ filename หมายถึง หมายเลขแฟ้มข้อมูล

ง) สำหรับเพิ่มข้อมูลทุกประเภท จะสร้างคำสั่ง
ต่อไปสำหรับเพิ่มข้อมูลออก(Output File)

```
SELECT OUTPUT-FILE ASSSIGN TO DISK
      ACCESS MODE IS SEQUENTIAL
      ORGANIZATION IS SEQUENTIAL
```

4.4.5 การสร้างส่วน DATA DIVISION จะสร้างคำสั่ง
ต่อไปนี้

```
DATA DIVISION.
```

ก) การสร้างส่วน FILE SECTION จะสร้าง
คำสั่งต่อไปนี้

```
FILE SECTION.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

1) สำหรับแฟ้มข้อมูล ISAM และ Text ประเภท COBOL Variable Length Record และ COBOL Fixed Length Record จะวนสร้างคำสั่งต่อไปในตามจำนวนแฟ้มข้อมูลทั้งหมด

FD INPUT-FILEfilename.

LABEL RECORD IS STANDARD

VALUE OF FILE-ID IS "filename"

DATA RECORD IS INfilename.

01 INfilename.

โดยที่ filename หมายถึง หมายเลขแฟ้มข้อมูล

filename หมายถึง ชื่อแฟ้มข้อมูล

จากนั้น จะวนสร้างรายชื่อเขตข้อมูล

ทั้งหมดของแฟ้มข้อมูลนี้

02 fieldnameINfilename PIC type(len)Vdec.

filename หมายถึง หมายเลขแฟ้มข้อมูล

filename หมายถึง ชื่อแฟ้มข้อมูล

fieldname หมายถึง ชื่อเขตข้อมูล

type หมายถึง X ถ้าเขตข้อมูลเป็นประเภท C

หมายถึง 9 ถ้าเขตข้อมูลเป็นประเภท N

len หมายถึง ความยาวของเขตข้อมูล

V หมายถึง ถ้ามีจุดทศนิยมด้วย

dec หมายถึง จำนวนตัวเลขหลังจุดทศนิยม

2) สำหรับแฟ้มข้อมูลทุกประเภท จะสร้าง

คำสั่งต่อไปนี้สำหรับแฟ้มข้อมูลออก(Output File)

FD OUTPUT-FILE.


LABEL RECORD IS STANDARD

VALUE OF FILE-ID IS "filename"

DATA RECORD IS OUT-REC.

01 OUTREC PIC X(pagewidth)

โดยที่ filename หมายถึง ชื่อแฟ้มข้อมูลที่จะจัดเก็บผลลัพธ์จากภาษาโคบอล
pagewidth หมายถึง จำนวนตัวอักษรต่อบรรทัด



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ข) การสร้างส่วน WORKING STORAGE SECTION จะสร้างส่วนต่าง ๆ ดังต่อไปนี้

WORKING STORAGE SECTION.

1) ถ้าหัวรายงานย่อย และรายการข้อมูล อยู่ในบรรทัดเดียวกันจะสร้างคำสั่งต่อไปนี้

77 CHANDDE PIC 9 VALUE 0.

2) นอกจากนี้จะสร้างคำสั่งต่อไปนี้

77 FIRSTPAGE PIC 9 VALUE 1.

77 LINENO PIC 9(2) VALUE 0.

77 LINECOUNT PIC 9(2) VALUE 0.

77 PAGELONG PIC 9(2) VALUE pagelong.

โดยที่ pagelong หมายถึง จำนวนบรรทัดต่อหน้าของรายงาน

3) ในกรณีที่รายงานที่ต้องการนั้น มีการคำนวณด้วยซึ่งการคำนวณจะมีอยู่ด้วยกัน 2 ลักษณะคือการคำนวณโดยใช้ฟังก์ชันและการคำนวณอื่น ๆ

(ก) สำหรับการคำนวณโดยใช้ฟังก์ชัน

มีรายละเอียดดังนี้

- การคำนวณหาค่าผลรวมจะ

สร้างตัวแปรต่อไปนี้

77 SUMvariablename PIC type(len)Vdec.

โดยที่ type len dec ได้มาจากรูปแบบในการพิมพ์ของข้อมูล
variablename หมายถึง ชื่อตัวแปร

- การคำนวณหาค่าเฉลี่ยโดย
มีสูตรที่ใช้ในการคำนวณดังนี้ ค่าเฉลี่ย = ผลรวมของข้อมูล / จำนวนข้อมูล ซึ่งจะ
สร้างตัวแปรต่าง ๆ ต่อไปนี้

77 SUMvariablename PIC type(len)Vdec.

77 COUNTvariablename PIC type(len)Vdec.

โดยที่ type len dec ได้มาจากรูปแบบในการพิมพ์ของข้อมูล
variablename หมายถึง ชื่อตัวแปร

- การคำนวณหาค่าความแปร
ปรวนโดยมีสูตรที่ใช้ในการคำนวณดังนี้ ค่าความแปรปรวน = (ผลรวมของข้อมูล²
- (ผลรวมทั้งหมด / จำนวนข้อมูล)²) / จำนวนข้อมูล และจะสร้างตัวแปรต่าง ๆ
ต่อไปนี้

77 SUMvariablename PIC type(len)Vdec.

77 SQRvariablename PIC type(len)Vdec.

77 COUNTvariablename PIC type(len)Vdec.

โดยที่ type len dec ได้มาจากรูปแบบในการพิมพ์ของข้อมูล
variablename หมายถึง ชื่อตัวแปร

- การคำนวณหาค่า ความ

เบี่ยงเบนมาตรฐานโดยมีสูตรที่ใช้ในการคำนวณดังนี้ ค่าความเบี่ยงเบนมาตรฐาน
= รากกำลังที่สองของค่าความแปรปรวน และ จะสร้างตัวแปรต่าง ๆ ต่อไปนี้

77 SUMvariablename PIC type(len)Vdec.

77 SQRvariablename PIC type(len)Vdec.

77 COUNTvariablename PIC type(len)Vdec.

โดยที่ type len dec ได้มาจากรูปแบบในการพิมพ์ของข้อมูล
variablename หมายถึง ชื่อตัวแปร

เนื่องจากต้องมีการคำนวณ

หาค่ารากกำลังที่สองด้วย จึงต้องสร้างตัวแปรต่าง ๆ ต่อไปนี้มาช่วยในการคำนวณ

77 FLAGSTD PIC 9 VALUE 0.

77 NOSTD PIC type(len)Vdec VALUE 0.

77 MINSTD PIC type(len)Vdec VALUE 0.

77 MIDSTD PIC type(len)Vdec VALUE 0.

77 DIVSTD PIC type(len)Vdec VALUE 0.

77 STD PIC PIC type(len)Vdec VALUE 0.

77 MAXSTD PIC type(len)Vdec VALUE 0.

โดยที่ type len dec ได้มาจากรูปแบบในการพิมพ์ของข้อมูล

(ข) ถ้าเป็นการคำนวณอื่น ๆ จะ

สร้างคำสั่งต่อไปนี้

```
77 variablename PIC type(len)Vdec.
```

โดยที่ type len dec ได้มาจากรูปแบบในการพิมพ์ของข้อมูล
variablename หมายถึง ชื่อตัวแปร

4) ในกรณีที่เป็นแฟ้มข้อมูลประเภท

Btrieve จะสร้างคำสั่งต่อไปนี้

```
77 BOPEN PIC 9(4) COMP-5 VALUE 0.
77 BCLOSE PIC 9(4) COMP-5 VALUE 1.
77 GETFIRST PIC 9(4) COMP-5 VALUE 12.
77 GETNEXT PIC 9(4) COMP-5 VALUE 6.
77 GETEQUAL PIC 9(4) COMP-5 VALUE 4.
77 STAT PIC 9(4) COMP-5 VALUE 0.
88 BEOF VALUE 9.
77 KEYNUMBER PIC 9(4) COMP-5 VALUE 0.
```

(ก) ถ้ามีแฟ้มข้อมูล เพียงแฟ้มข้อมูล

เดียวจะสร้างคำสั่งต่อไปนี้

```
77 LEN PIC 9(4) COMP-5 VALUE record size.
77 POSBLK PIC X(128).
โดยที่ record size หมายถึง ความยาวของระเบียน
no. of files หมายถึง จำนวนแฟ้มข้อมูล
```

(ข) ถ้ามีแฟ้มข้อมูลมากกว่า 1 แฟ้ม

จะสร้างคำสั่งต่อไปนี้

77 LEN OCCURS no. of files PIC 9(4) COMP-5 VALUE 0.

77 POSBLK OCCURS no. of files PIC X(128).

โดยที่ record size หมายถึง ความยาวของระเบียบ

no. of files หมายถึง จำนวนแฟ้มข้อมูล



5) การสร้างส่วนตัวแปร สำหรับการไว้

เป็นตัวกำหนดจุดสิ้นสุดของแฟ้มข้อมูล จะสร้างคำสั่งต่อไปนี้

01 MORE-DATA-SWITCH PIC X VALUE 'Y'

88 THERE-IS-MORE-DATA VALUE 'Y'

88 THERE-IS-NO-MORE-DATA VALUE 'N'.

6) การสร้างส่วนตัวแปร สำหรับการขึ้น

บรรทัดใหม่ จะสร้างคำสั่งต่อไปนี้

01 NEWLINE.

02 FILLER PIC X(pagewidth) VALUE SPACE.

02 FILLER PIC X(2) VALUE X"0A0D".

โดยที่ pagewidth หมายถึง ความกว้างของรายงาน

7) ในกรณีที่ เป็นแฟ้มข้อมูล dBASE จะ
 วนสร้างคำสั่งต่าง ๆ ต่อไปนี้ตามจำนวนแฟ้มข้อมูลทั้งหมด

(ก) สร้างชื่อแฟ้มข้อมูลโดยที่ถ้าเป็น
 แฟ้มข้อมูลใช้งาน จะสร้างคำสั่งต่อไปนี้

01 FNAME filename.

02 FILLER PIC X(length of filename) VALUE "filename
 .dbf".

02 FILLER PIC X VALUE X"00".

โดยที่ length of filename หมายถึง ความยาวของชื่อแฟ้มข้อมูล
 filename.dbf หมายถึง ชื่อแฟ้มข้อมูลใช้งาน

- ถ้าแฟ้มข้อมูลใช้งาน มีแฟ้ม

ข้อมูลดัชนีด้วยจะสร้างคำสั่งต่อไปนี้

01 FNAMENDX filename.

02 FILLER PIC X(length of filename) VALUE "filename
 .ndx".

02 FILLER PIC X VALUE X"00".

โดยที่ length of filename หมายถึง ความยาวของชื่อแฟ้มข้อมูล
 filename.ndx หมายถึง ชื่อแฟ้มข้อมูลดัชนี

(ข) ถ้ามีแฟ้มข้อมูล เพียงแฟ้มข้อมูล

เดียว จะสร้างคำสั่งต่อไปนี้

01 BUFFER PIC X(max. of record size).

01 RECLEN PIC XX.

01 FHAND PIC XX.

01 COUNTREC PIC X VALUE X"00".

01 HEADERDBF PIC XX VALUE headersize.

01 HBUFF.

02 FILLER PIC X(4).

02 TOTALREC PIC X(4).

02 FILLER PIC X(max. of record size-8)

โดยที่ max of record size หมายถึงความยาวระเบียบสูงสุดของแฟ้มข้อมูล
header size หมายถึง (จำนวนเขตข้อมูล+1)*32+2.

- ถ้าแฟ้มข้อมูลใช้งานมีแฟ้ม

ข้อมูลดัชนีด้วย จะสร้างคำสั่งต่อไปนี้

01 FHANDNDX PIC XX.

01 HBUFFNDX.

02 FILLER PIC X(14).

02 RECNO PIC X.

02 FILLER PIC X(3).

02 RECLENNDX PIC X.

02 FILLER PIC X(493).

(ค) ถ้ามีแฟ้มข้อมูลมากกว่า 1 แฟ้ม

จะสร้างคำสั่งต่อไปนี้

```

01 BUFFER PIC X(max. of record size).
01 RECLen PIC XX.
01 FHAND OCCURS no. of files PIC XX.
01 COUNTREC OCCURS no. of files PIC X VALUE X"00".
01 HEADERDBF OCCURS no. of files PIC XX.
01 HBUFF OCCURS no. of files.
    02 FILLER PIC X(4).
    02 TOTALREC PIC X(4).
    02 FILLER PIC X(max. of record size-8)
โดยที่ max of record size หมายถึงความยาวระเบียบสูงสุดของแฟ้มข้อมูล
no of files หมายถึง จำนวนแฟ้มข้อมูลทั้งหมด

```

- ถ้าแฟ้มข้อมูลใช้งานมีแฟ้ม

ข้อมูลดัชนีด้วยจะสร้างคำสั่งต่อไปนี้

```

01 FHANDNDX OCCURS no. of files PIC XX.
01 HBUFFNDX OCCURS no. of files.
    02 FILLER PIC X(14).
    02 RECNO PIC X.
    02 FILLER PIC X(3).
    02 RECLenNDX PIC X.
    02 FILLER PIC X(493).
โดยที่ no of files หมายถึง จำนวนแฟ้มข้อมูลทั้งหมด

```


8) ในกรณีที่เก็บแฟ้มข้อมูล Foxbase จะ
 วนสร้างคำสั่งต่อไปนี้อย่างไรตามจำนวนแฟ้มข้อมูลทั้งหมด

(ก) ถ้าเป็นแฟ้มข้อมูลใช้งานจะสร้าง
 คำสั่งต่อไปนี้

01 FNAMEfilename.

02 FILLER PIC X(length of filename) VALUE "filename
 .dbf".

02 FILLER PIC X VALUE X"00".

โดยที่ length of filename หมายถึง ความยาวของชื่อแฟ้มข้อมูล
 filename.dbf หมายถึง ชื่อแฟ้มข้อมูลใช้งาน

- ถ้าแฟ้มข้อมูลใช้งานนั้นมีแฟ้ม

ข้อมูลดัชนีด้วยจะสร้างคำสั่งต่อไปนี้

01 FNAMEIDX filename.

02 FILLER PIC X(length of filename) VALUE "filename
 .idx".

02 FILLER PIC X VALUE X"00".

โดยที่ length of filename หมายถึง ความยาวของชื่อแฟ้มข้อมูล
 filename.idx หมายถึง ชื่อแฟ้มข้อมูลดัชนี

(ข) ถ้ามีแฟ้มข้อมูลเพียงแฟ้มข้อมูล

เดียว จะสร้างคำสั่งต่อไปนี้

```
01 BUFFER PIC X(max. of record size).
01 RECLLEN PIC XX.
01 FHAND PIC XX.
01 COUNTREC PIC X VALUE X"00".
01 HEADERDBF PIC XX VALUE header size.
01 HBUFF.
    02 FILLER PIC X(4).
    02 TOTALREC PIC X(4).
    02 FILLER PIC X(max. of record size-8).
```

โดยที่ max of record size หมายถึง ความยาวระเบียบสูงสุดของ
แฟ้มข้อมูลทั้งหมด

header size

หมายถึง (จำนวนเขตข้อมูล+1)*32+2.

- ถ้าแฟ้มข้อมูลใช้งานมีแฟ้ม

ข้อมูลดัชนีด้วยจะสร้างคำสั่งต่อไปนี้

```
01 FHANDIDX PIC XX.
01 HBUFFIDX.
    02 FILLER PIC X(12).
    02 RECLLENIDX PIC X.
    02 FILLER PIC X(499).
```

(ค) ถ้ามีแฟ้มข้อมูลมากกว่า 1 แฟ้ม

จะสร้างคำสั่งต่อไปนี้

```
01 BUFFER PIC X(max. of record size).
01 RECLN PIC XX.
01 FHAND OCCURS no. of files PIC XX.
01 COUNTREC OCCURS no. of files PIC X VALUE X"00".
01 HEADERDBF OCCURS no. of files PIC XX.
01 HBUFF OCCURS no. of files.
    02 FILLER PIC X(4).
    02 TOTALREC PIC X(4).
    02 FILLER PIC X(max. of record size-8).
```

โดยที่ max of record size หมายถึง ความยาวระเบียบสูงสุดของ
แฟ้มข้อมูล

no of files

หมายถึง จำนวนแฟ้มข้อมูลทั้งหมด

ถ้าแฟ้มข้อมูลใช้งานมีแฟ้มข้อมูล

ดัชนีด้วย จะสร้างคำสั่งต่อไปนี้

```
01 FHANDIDX OCCURS no. of files PIC XX.
01 HBUFFIDX OCCURS no. of files.
    02 FILLER PIC X(12).
    02 RECLNIDX PIC X.
    02 FILLER PIC X(499).
```

โดยที่ no of files

หมายถึง จำนวนแฟ้มข้อมูลทั้งหมด

9) ในกรณีที่เป็นแฟ้มข้อมูล Btrieve จะ
วนทำหัวข้อต่อไปนี้อัตโนมัติตามจำนวนแฟ้มข้อมูลทั้งหมด

01 FNAMEfilename.

02 FILLER PIC X(length of filename) VALUE "filename"

02 FILLER PIC X VALUE X"00".

โดยที่ length of filename หมายถึง ความยาวของชื่อแฟ้มข้อมูล
filename หมายถึง ชื่อแฟ้มข้อมูล

(ก) ถ้าแฟ้มข้อมูลมีเขตข้อมูล ที่เป็น
คีย์เพียงเขตข้อมูลเดียวจะสร้างคำสั่งต่อไปนี้

01 KEYBUFFfilename PIC type(len)Vdec.

โดยที่ filename หมายถึง หมายเลขแฟ้มข้อมูล

type len dec ได้มาจากรูปแบบการพิมพ์ของข้อมูล

(ข) ถ้าแฟ้มข้อมูลมีเขตข้อมูล ที่เป็น
คีย์มากกว่า 1 เขตข้อมูลจะสร้างคำสั่งต่อไปนี้

01 KEYBUFFfilename-keynumber.

โดยที่ filename หมายถึง หมายเลขแฟ้มข้อมูล

keynumber หมายถึง หมายเลขคีย์

จำนวนเขตข้อมูลที่เป็นคีย์ที่มีหมายเลขเดียวกัน

02 KEYBUFFfieldnamefilenumber-keynumber PIC type(len)

Vdec.

โดยที่	fieldname	หมายถึง	ชื่อเขตข้อมูล
	filenumber	หมายถึง	หมายเลขแฟ้มข้อมูล
	keynumber	หมายถึง	หมายเลขคีย์
	type len dec	ได้มาจาก	รูปแบบการพิมพ์ของข้อมูล

10) ถ้าเป็นแฟ้มข้อมูล Text ประเภท

COBOL Fixed Length Record

(ก) ถ้ามีแฟ้มข้อมูลเพียงแฟ้มข้อมูล

เดียวจะสร้างคำสั่งต่อไปนี้

01 TOTALREC PIC 9(no. of digit) VALUE no. of record.

01 COUNTTOTALREC PIC 9(no. of digit) VALUE 0.

โดยที่ no. of digit หมายถึง ตัวเลขที่มากพอที่จะจัดเก็บจำนวน
ระเบียนทั้งหมดได้

no. of record หมายถึง จำนวนระเบียนทั้งหมด

จุฬาลงกรณ์มหาวิทยาลัย

(ข) ถ้ามีแฟ้มข้อมูลมากกว่า 1

แฟ้ม จะสร้างคำสั่งต่อไปนี้

01 TOTALREC OCCURS no. of files PIC 9 (no. of digit)
 VALUE 0.
 01 COUNTTOTALREC OCCURS no.of files PIC 9 (no. of digit)
 VALUE 0.

โดยที่ no. of files หมายถึง จำนวนแฟ้มข้อมูลทั้งหมด
 no. of digit หมายถึง ตัวเลขที่มากพอที่จะจัดเก็บจำนวนระเบียบ
 ทั้งหมดได้

11) ในกรณีที่เป็นแฟ้มข้อมูล Text
 ประเภท Other Fixed Length Record จะวนสร้างชื่อแฟ้มข้อมูลทั้งหมดตาม
 จำนวนแฟ้มข้อมูลทั้งหมด

01 FNAMEfilename.
 02 FILLER PIC X(length of filename) VALUE "filename"
 02 FILLER PIC X VALUE X"00".
 โดยที่ filename หมายถึง
 length of filename หมายถึง ความยาวของชื่อแฟ้มข้อมูล
 filename หมายถึง ชื่อแฟ้มข้อมูล

(ก) ถ้ามีแฟ้มข้อมูลเพียงแฟ้มข้อมูล

เดี๋ยวจะสร้างคำสั่งต่อไปนี้

```

01 BUFFER PIC X(max. of record size).
01 RECLEN PIC XX.
01 FHAND PIC XX.
01 TOTALREC PIC X(4) VALUE X"no. of record".
01 COUNTTOTALREC PIC X(4) VALUE X"00000000".

```

โดยที่ max. of record size หมายถึง ความยาวสูงสุดของระเบียน
no. of record หมายถึง จำนวนระเบียนทั้งหมด

(ข) ถ้ามีแฟ้มข้อมูลมากกว่า 1

แฟ้ม จะสร้างคำสั่งต่อไปนี้

```

01 BUFFER PIC X(max. of record size).
01 RECLEN PIC XX.
01 FHAND OCCURS no.of files PIC XX.
01 TOTALREC PIC X(4) OCCURS no. of files VALUE X
  "no. of record".
01 COUNTTOTALREC PIC X(4) OCCURS no.of files VALUE
  X"00000000".

```

โดยที่ max. of record size หมายถึง ความยาวสูงสุดของระเบียน
no. of files หมายถึง จำนวนแฟ้มข้อมูลทั้งหมด
no. of record หมายถึง จำนวนระเบียนทั้งหมด

12) ในกรณีที่เป็นแฟ้มข้อมูลของ dBASE Foxbase Btrieve และ Text ประเภท Other Fixed Length Record จะสร้างหัวข้อต่าง ๆ ต่อไปนี้ตามจำนวนแฟ้มข้อมูลทั้งหมด

01 INfilename.

โดยที่ filename หมายถึงเลขแฟ้มข้อมูล

แฟ้มข้อมูลแต่ละแฟ้มก็จะสร้างหัวข้อ
ต่อไปนี้ตามจำนวนเขตข้อมูลทั้งหมดของแฟ้มข้อมูลนี้

02 'fieldname'INfilename PIC type(len)Vdec.

โดยที่ fieldname หมายถึงชื่อเขตข้อมูล

filename หมายถึงเลขแฟ้มข้อมูล

type len dec ได้มาจากรูปแบบการพิมพ์ของข้อมูล

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

13) สร้างระดับต่าง ๆ ของรายงานโดย
ที่ชื่อของแต่ละระดับแสดงได้ดังตารางที่ 4.4

ชื่อระดับ	ความหมาย
RH	ชื่อของระดับหัวรายงานของแต่ละชุด
PH	ชื่อของระดับหัวรายงานของแต่ละหน้า
CL	ชื่อของระดับหัวรายงานของแต่ละรายการ
CH	ชื่อของระดับหัวรายงานเมื่อเกิดการเปลี่ยนค่าในเขตข้อมูล
DE	ชื่อของรายการ
CF	ชื่อของระดับสรุปท้ายรายงานเมื่อเกิดการเปลี่ยนค่าในเขตข้อมูล
PF	ชื่อของระดับหัวรายงานของแต่ละหน้า
RF	ชื่อของระดับหัวรายงานของแต่ละชุด

ตารางที่ 4.4 แสดงชื่อของระดับต่าง ๆ

จำนวนบรรทัดทั้งหมด
จะวนสร้างค่าสิ่งต่าง ๆ ต่อไปนี้ตาม
- สร้างชื่อของแต่ละระดับ โดยจะ
สร้างค่าสิ่ง ต่อไปนี้

01 levelname:rownumber.

โดยที่ levelnumber หมายถึง ชื่อของแต่ละระดับของรายงานโดยมีรายละเอียดดังตารางที่ 4.4

rownumber หมายถึง หมายเลขของบรรทัดโดยมีค่าเริ่มต้นเป็น 0 แล้วเพิ่มค่าครั้งละ 1 ตามจำนวนบรรทัดทั้งหมดของระดับนี้

หลังจากที่สร้างชื่อแต่ละระดับไปแล้ว

จะสร้างหัวข้อต่อไปนี้อย่างตามจำนวนข้อมูลทั้งหมดที่มีอยู่ในบรรทัดนั้น ๆ

- ถ้าข้อมูลในบรรทัดนั้นเป็นค่าคงที่

จะสร้างคำสั่งต่อไปนี้

02 FILLER PIC X(length of constant value) VALUE

"constant value".

โดยที่ length of constant value หมายถึง ความยาวของค่าคงที่

constant value หมายถึง ค่าคงที่

- ถ้าข้อมูลในบรรทัดนั้นเป็นตัวแปร

จะสร้างคำสั่งต่อไปนี้

02 variablename!levelname!rownumber PIC type(len)Vdec.

โดยที่ variablename หมายถึง ชื่อตัวแปร

levelname หมายถึง ชื่อของแต่ละระดับของรายงาน

rownumber หมายถึง หมายเลขของบรรทัดโดยมีค่าเริ่มต้นเป็น

0 แล้วเพิ่มค่าครั้งละ 1 ตามจำนวนบรรทัดทั้งหมดของระดับนี้

type len dec ได้มาจากรูปแบบการพิมพ์

14) การสร้างส่วนตัวแปรต่าง ๆ สำหรับ

เขตข้อมูลคอนโทรลเบรค จะสร้างคำสั่งต่อไปนี้

01 CT.

ตามจำนวนเขตข้อมูลคอนโทรลเบรคทั้งหมด

02 fieldnameCT PIC type(len)Vdec.

โดยที่ fieldname หมายถึง ชื่อเขตข้อมูล

type len dec ได้มาจากรูปแบบการพิมพ์



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4.4.6 การสร้างส่วน PROCEDURE DIVISION จะ

สร้างคำสั่งต่อไปนี้

```
PROCEDURE DIVISION.
```

ก) การกำหนดค่าคงที่ให้กับตัวแปรต่าง ๆ จะ
สร้างคำสั่ง ต่อไปนี้ก็ต่อเมื่อมีแฟ้มข้อมูลมากกว่า 1 แฟ้ม

```
INIT-RTN.
```

1) สำหรับแฟ้มข้อมูล dBASE และ
Foxbase จะสร้างคำสั่งต่อไปนี้ตามจำนวนแฟ้มข้อมูลทั้งหมด

```
MOVE "header size" TO HEADERDBF(filenumber+1)
```

โดยที่ header size หมายถึง ความยาวของส่วนเริ่มต้นของแฟ้มข้อมูล
แต่ละแฟ้มซึ่งคำนวณได้จาก $\text{header size} = (\text{จำนวนเขตข้อมูล} + 1) * 32 + 2$
filenumber หมายถึง หมายเลขของแฟ้มข้อมูล

2) สำหรับแฟ้มข้อมูลข้อมูล Btrieve จะ
สร้างคำสั่งต่อไปนี้ตามจำนวนแฟ้มข้อมูลทั้งหมด

```
MOVE record size TO LEN (filenumber+1)
```

โดยที่ record size หมายถึง ความยาวของแต่ละระเบียน
filenumber หมายถึง หมายเลขแฟ้มข้อมูล

3) สำหรับแฟ้มข้อมูล Text ประเภท COBOL Fixed Length Record และ Other Fixed Length Record จะสร้างคำสั่งต่อไป้ตามจำนวนแฟ้มข้อมูลทั้งหมด

```
MOVE no.of record TO TOTALREC(filenumber+1)
```

โดยที่ no.of record หมายถึง จำนวนระเบียบข้อมูลทั้งหมดของแฟ้มข้อมูล
filenumber หมายถึง หมายเลขแฟ้มข้อมูล

ข) การเปิดแฟ้มข้อมูลต่าง ๆ จะสร้างคำสั่งต่อไป้

```
OPEN-OUTPUT-RTN.
```

```
OPEN OUTPUT OUTPUT-FILE.
```

```
OPEN-INPUT-RTN.
```

1) สำหรับแฟ้มข้อมูล dBASE

(ก) ถ้ามีแฟ้มข้อมูลใช้งานเพียงแฟ้ม

ข้อมูลเดียวจะสร้างคำสั่งต่อไป้

```
CALL "_OPEN" USING FNAMEO,FHAND.
```

```
MOVE header size TO RECLN.
```

```
CALL "_READHEAD" USING FHAND,HBUFF,RECLN.
```

โดยที่ header size หมายถึง ความยาวของส่วนเริ่มต้นของแฟ้มข้อมูล
แต่ละแฟ้มซึ่งคำนวณได้จาก $\text{header size} = (\text{จำนวนเขตข้อมูล} + 1) * 32 + 2$

- ถ้าเพิ่มข้อมูลใช้งานมีเพิ่ม

ข้อมูลดัชนีด้วยจะสร้างคำสั่งต่อไปนี้

```
MOVE X"0200" TO RECLN.
CALL "_READHEAD" USING FHANDNDX, HBUFFNDX, RECLN.
```

(๒) ถ้ามีเพิ่มข้อมูลใช้งานมากกว่า 1
เพิ่ม จะสร้างคำสั่งต่าง ๆ ต่อไปนี้ตามจำนวนเพิ่มข้อมูลทั้งหมด

```
CALL "_OPEN" USING FNAMEfilename, FHAND(filename+1).
MOVE header size TO RECLN.
CALL "_READHEAD" USING FHAND(filename + 1), HBUFF
(filename+1), RECLN.
```

โดยที่ filename หมายถึง หมายเลขของเพิ่มข้อมูล

header size หมายถึง ความยาวของส่วนหัวของเพิ่มข้อมูล

แต่ละเพิ่มซึ่งคำนวณได้จาก $header\ size = (จำนวน\ เขต\ ข้อมูล + 1) * 32 + 2$

ถ้าเพิ่มข้อมูลใช้งานมีเพิ่มข้อมูล

ดัชนีด้วย จะสร้างคำสั่งต่อไปนี้

```
MOVE X"0200" TO RECLN.
CALL "_READHEAD" USING FHANDNDX(filename + 1),
HBUFFNDX(filename+1), RECLN.
```

โดยที่ filename หมายถึง หมายเลขของเพิ่มข้อมูล

2) สำหรับแฟ้มข้อมูล Foxbase

(ก) ถ้ามีแฟ้มข้อมูลใช้งานเพียงแฟ้ม

ข้อมูลเดียวจะสร้างคำสั่งต่อไปนี้

```
CALL "_OPEN" USING FNAMEO, FHAND.
```

```
MOVE header size TO RECLEN.
```

```
CALL "_READHEAD" USING FHAND, HBUFF, RECLEN.
```

โดยที่ header size หมายถึง ความยาวของส่วนหัวของแฟ้มข้อมูล

แต่ละแฟ้มซึ่งคำนวณได้จาก $header\ size = (จำนวนเขตข้อมูล + 1) * 32 + 2$

ถ้าแฟ้มข้อมูลใช้งานมีแฟ้มข้อมูล

ดัชนีด้วย จะสร้างคำสั่งต่อไปนี้

```
MOVE X"0200" TO RECLEN.
```

```
CALL "_READHEAD" USING FHANDIDX, HBUFFIDX, RECLEN.
```

(ข) ถ้ามีแฟ้มข้อมูลใช้งานมากกว่า 1

แฟ้ม จะสร้างคำสั่งต่อไปนี้ตามจำนวนแฟ้มข้อมูลทั้งหมด

```
CALL "_OPEN" USING FNAMEfilename, FHAND(filename+1).
```

```
MOVE header size TO RECLEN.
```

```
CALL "_READHEAD" USING FHAND(filename+1), HBUFF
```

```
(filename+1), RECLEN.
```

โดยที่ filename หมายถึง หมายเลขของแฟ้มข้อมูล

header size หมายถึง ความยาวของส่วนหัวของแฟ้มข้อมูลแต่

ละแฟ้มซึ่งคำนวณได้จาก $header\ size = (จำนวนเขตข้อมูล + 1) * 32 + 2$

ถ้าเพิ่มข้อมูลใช้งานมีเพิ่มข้อมูล

ดัชนีด้วย จะสร้างคำสั่งต่อไปนี้

```
MOVE X"0200" TO RECLEN.
```

```
CALL "_READHEAD" USING FHANDIDX(filenummer + 1),  
HBUFFIDX(filenummer+1),RECLEN.
```

โดยที่ filenummer หมายถึง หมายเลขของแฟ้มข้อมูล

3) สำหรับแฟ้มข้อมูล Btrieve

(ก) ถ้ามีแฟ้มข้อมูล เพียงแฟ้มข้อมูล

เดียว

(1) ถ้ามีหมายเลขคีย์เดียวกัน

จะสร้างคำสั่งต่อไปนี้

```
MOVE keynumber TO KEYNUMBER.
```

```
CALL "_BTRV" USING BOPEN,STAT,POSBLK,INO,LEN,  
KEYBUFFfilenummer,KEYNUMBER.
```

โดยที่ keynumber หมายถึง หมายเลขคีย์

filenummer หมายถึง หมายเลขแฟ้ม

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

(2) ถ้ามีหมายเลขคีย์ต่างกัน

จะสร้างคำสั่งต่อไปนี้

```
MOVE key_number TO KEYNUMBER.
CALL "_BTRV" USING BOPEN, STAT, POSBLK, INO, LEN,
KEYBUFFfilenumber-keynumber, KEYNUMBER.
```

โดยที่ filenumber หมายถึง หมายเลขแฟ้ม

key_number หมายถึง หมายเลขคีย์ที่มีค่าน้อยที่สุดของแฟ้มข้อมูลนี้

(ข) ถ้ามีแฟ้มข้อมูลมากกว่า 1 แฟ้ม

จะวนสร้างหัวข้อต่อไปนตามจำนวนแฟ้มข้อมูลทั้งหมด

(1) ถ้ามีหมายเลขคีย์เดียวกัน

จะสร้างคำสั่งต่อไปนี้

```
MOVE keynumber TO KEYNUMBER.
CALL "_BTRV" USING BOPEN, STAT, POSBLK(filenumber+1),
INfilenumber, LEN(filenumber+1), KEYBUFFfilenumber,
KEYNUMBER.
```

โดยที่ filenumber หมายถึง หมายเลขแฟ้มข้อมูล

keynumber หมายถึง หมายเลขคีย์



(2) ถ้ามีหมายเลขคีย์ต่างกัน

จะสร้างคำสั่งต่อไปนี้

```
MOVE key_number TO KEYNUMBER.
CALL "_BTRV" USING BOPEN,STAT,POSBLK(filenummer + 1) ,
  INfilenummer,LEN(filenummer + 1),KEYBUFFfilenummer
  -keynumber,KEYNUMBER.
```

โดยที่ key_number หมายถึง หมายเลขคีย์ที่มีค่าน้อยที่สุดของแฟ้มข้อมูลนี้
keynumber หมายถึง หมายเลขคีย์

4) สำหรับแฟ้มข้อมูล ISAM จะสร้างคำสั่งต่อไปนี้ตามจำนวนแฟ้มข้อมูลทั้งหมด

```
OPEN INPUT INPUT-FILEfilenummer.
```

โดยที่ filenummer หมายถึง หมายเลขแฟ้มข้อมูล

5) สำหรับแฟ้มข้อมูล Text ประเภท COBOL Variable Length Record และ ประเภท COBOL Fixed Length Record จะสร้างคำสั่งต่อไปนี้

```
OPEN INPUT INPUT-FILE0.
```

6) สำหรับแฟ้มข้อมูล Text ประเภท

Other Fixed Length Record

- ถ้ามีแฟ้มข้อมูลเพียงแฟ้มข้อมูลเดียว

จะสร้างคำสั่งต่อไปนี้

```
CALL "_OPEN" USING FNAME0,FHAND.
```

- ถ้ามีแฟ้มข้อมูลมากกว่า 1 แฟ้มจะ

สร้างคำสั่งต่อไปนี้

```
CALL "_OPEN" USING FNAME(filenumber+1),FHAND(filenumber  
+1).
```

โดยที่ filenumber หมายถึง หมายเลขแฟ้มข้อมูล

- ค) การอ่านระเบียนแรกของแฟ้มข้อมูลหลักมา 1

ระเบียน

- 1) สำหรับแฟ้มข้อมูล Btrieve จะสร้าง

คำสั่งต่อไปนี้

```
MOVE keynumber TO KEYNUMBER.
```

โดยที่ keynumber หมายถึง หมายเลขคีย์

(ก) ถ้ามีแฟ้มข้อมูล เพียงแฟ้มข้อมูล

เดียว

- ถ้ามีหมายเลขคีย์เดียวกัน

จะสร้างคำสั่งต่อไปนี้

```
CALL "_BTRV" USING GETFIRST, STAT, POSBLK, INO, LEN, KEYBUFFO,
KEYNUMBER.
```

- ถ้ามีหมายเลขคีย์ต่างกันจะ

สร้างคำสั่งต่อไปนี้

```
CALL "_BTRV" USING GETFIRST, STAT, POSBLK, INO, LEN, KEYBUFFO-
keynumber, KEYNUMBER.
```

โดยที่ keynumber หมายถึง หมายเลขคีย์

(ข) ถ้ามีแฟ้มข้อมูลมากกว่า 1 แฟ้ม

- ถ้ามีหมายเลขคีย์เดียวกัน

จะสร้างคำสั่งต่อไปนี้

```
CALL "_BTRV" USING GETFIRST, STAT, POSBLK(filenum+1) ,
INfilenum, LEN(filenum+1), KEYBUFFfilenum,
KEYNUMBER.
```

โดยที่ filenum หมายถึง หมายเลขแฟ้ม

- ถ้ามีหมายเลขคีย์ต่างกันจะ

สร้างคำสั่งต่อไปนี้

```
CALL "_BTRV" USING GETFIRST,STAT,POSBLK(filenum+1),
      INfilenum,LEN(filenum + 1),KEYBUFFfilenum-
      keynumber,KEYNUMBER.
```

โดยที่ filenum หมายถึง หมายเลขแฟ้มข้อมูล
keynumber หมายถึง หมายเลขคีย์

7) สำหรับแฟ้มข้อมูลประเภทอื่น ๆ จะ

สร้างคำสั่งต่อไปนี้

```
PERFORM READ-RTNO.
```

ง) การสร้างส่วนในการวนลูป MAIN-RTN
รoutines (Routine) จนกว่าจะหมดแฟ้มข้อมูลหลัก

1) สำหรับแฟ้มข้อมูล dBASE และ

Foxbase

(ก) ถ้ามีแฟ้มข้อมูล เพียงแฟ้มข้อมูล
เดียว จะสร้างคำสั่งต่อไปนี้

```
PERFORM MAIN-RTN
      UNTIL COUNTTOTALREC > TOTALREC
      OR THERE-IS-NO-MORE-DATA.
```

(ข) ถ้ามีแฟ้มข้อมูลมากกว่า 1 แฟ้ม

จะสร้างคำสั่งต่อไปนี้

```
PERFORM MAIN-RTN
      UNTIL COUNTTOTALREC(1) > TOTALREC(1)
      OR THERE-IS-NO-MORE-DATA.
```

2) สำหรับแฟ้มข้อมูล Btrieve จะสร้างคำสั่งต่าง ๆ ต่อไปนี้

```
PERFORM MAIN-RTN UNTIL BEOF.
```

3) สำหรับแฟ้มข้อมูล ISAM และ แฟ้มข้อมูล Text ประเภท COBOL Variable Length Record และ ประเภท COBOL Fixed Length Record จะสร้างคำสั่งต่อไปนี้

```
PERFORM MAIN-RTN
      UNTIL THERE-IS-NO-MORE-DATA.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4) สำหรับเพิ่มข้อมูล Text ประเภท

Other Fixed Length Record

- ถ้ามีเพิ่มข้อมูล เพียงเพิ่มข้อมูล
เดียวจะสร้างคำสั่งต่อไปนี้

```
PERFORM MAIN-RTN
  UNTIL COUNTTOTALREC > TOTALREC
  OR THERE-IS-NO-MORE-DATA.
```

- ถ้ามีเพิ่มข้อมูลมากกว่า 1 แฝ้ม
จะสร้างคำสั่งต่อไปนี้

```
PERFORM MAIN-RTN
  UNTIL COUNTTOTALREC(1) > TOTALREC(1)
  OR THERE-IS-NO-MORE-DATA.
```

จ) การสร้างส่วนจบโปรแกรมของส่วน
PROCEDURE DIVISION จะสร้างคำสั่ง ต่อไปนี้

```
END-REPORT-RTN.
```

1) การสร้างส่วนในการพิมพ์ส่วนของท้าย
รายงานย่อย ท้ายรายงาน ท้ายชุดรายงาน

(ก) ถ้ามีการสั่งพิมพ์ ท้ายรายงาน
ย่อย จะวนสร้างคำสั่งต่าง ๆ ต่อไปนี้ ตามจำนวนเขตข้อมูลคอนโทรลเบรคโดย
เรียงระดับความสำคัญจากน้อยไปมาก

PERFORM CF-fieldnameRTN.

โดยที่ fieldname หมายถึง ชื่อเขตข้อมูล

(ข) ถ้ามีการสั่งพิมพ์ท้ายรายงานจะ
สร้างคำสั่งต่อไปนี้

PERFORM PF-RTN.

(ค) ถ้ามีการสั่งพิมพ์ท้ายชุดรายงาน
จะสร้างคำสั่งต่อไปนี้

PERFORM RF-RTN.

จ) การสร้างส่วนในการปิดแฟ้มข้อมูลเข้าต่าง ๆ

1) สำหรับแฟ้มข้อมูล dBASE

(ก) ถ้ามีแฟ้มข้อมูลใช้งานเพียงแฟ้ม

ข้อมูลเดียวจะสร้างคำสั่งต่อไปนี้

CALL "_CLOSE" USING FHAND.

ถ้าเพิ่มข้อมูลใช้งานมีเพิ่มข้อมูล

ดังนี้ด้วย จะสร้างคำสั่งต่อไปนี้

```
CALL "_CLOSE" USING FHANDNDX.
```

(ข) ถ้ามีเพิ่มข้อมูลใช้งานมากกว่า 1

เพิ่ม จะสร้างคำสั่งต่อไปนี้ตามจำนวนเพิ่มข้อมูลทั้งหมด

```
CALL "_CLOSE" USING FHAND(filename+1).
```

โดยที่ filename หมายถึง หมายเลขเพิ่ม

ถ้าเพิ่มข้อมูลใช้งานมีเพิ่มข้อมูล

ดังนี้ด้วย จะสร้างคำสั่งต่อไปนี้

```
CALL "_CLOSE" USING FHANDNDX(filename+1).
```

โดยที่ filename หมายถึง หมายเลขเพิ่ม

2) สำหรับเพิ่มข้อมูล Foxbase

(ก) ถ้ามีเพิ่มข้อมูลใช้งานเพียงเพิ่ม

ข้อมูลเดียวจะสร้างคำสั่งต่อไปนี้

```
CALL "_CLOSE" USING FHAND.
```

ถ้าเพิ่มข้อมูลใช้งานมีเพิ่มข้อมูล

ดัชนีด้วย จะสร้างคำสั่งต่าง ๆ ต่อไปนี้

```
CALL "_CLOSE" USING FHANDIDX.
```

(ข) ถ้ามีเพิ่มข้อมูลใช้งานมากกว่า 1

เพิ่ม จะสร้างคำสั่งต่าง ๆ ต่อไปนี้ตามจำนวนเพิ่มข้อมูลทั้งหมด

```
CALL "_CLOSE" USING FHAND(filenummer+1).
```

โดยที่ filenummer หมายถึง หมายเลขเพิ่มข้อมูล

ถ้าเพิ่มข้อมูลใช้งานมีเพิ่มข้อมูล

ดัชนีด้วย จะสร้างคำสั่งต่อไปนี้

```
CALL "_CLOSE" USING FHANDIDX(filenummer+1).
```

โดยที่ filenummer หมายถึง หมายเลขเพิ่มข้อมูล

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3) สำหรับแฟ้มข้อมูล Btrieve

(ก) ถ้ามีแฟ้มข้อมูลใช้งานเพียงแฟ้ม

ข้อมูลเดียว

- ถ้ามีหมายเลขคีย์เดียวกัน

จะสร้างคำสั่งต่อไปนี้

```
MOVE keynumber TO KEYNUMBER.
CALL "_BTRV" USING BCLOSE, STAT, POSBLK, INO, KEYBUFFO,
KEYNUMBER.
```

โดยที่ keynumber หมายถึง หมายเลขคีย์

- ถ้ามีหมายเลขคีย์ต่างกัน

จะสร้างคำสั่งต่อไปนี้

```
MOVE key_number TO KEYNUMBER.
CALL "_BTRV" USING BCLOSE, STAT, POSBLK, INO, KEYBUFFO-
key_number, KEYNUMBER.
```

โดยที่ key_number หมายถึงหมายเลขคีย์ที่มีค่าน้อยที่สุด

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

(ข) ถ้ามีแฟ้มข้อมูลใช้งานมากกว่า

1 แฟ้ม

- ถ้ามีหมายเลขคีย์เดียวกัน

จะสร้างคำสั่งต่อไปนี้

```
MOVE keynumber TO KEYNUMBER.
CALL "_BTRV" USING BCLOSE,STAT,POSBLK(filenum+1),
      INfilenum,KEYBUFFfilenum-keynumber,KEYNUMBER.
```

โดยที่ keynumber หมายถึง หมายเลขคีย์

filenum หมายถึง หมายเลขแฟ้มข้อมูล

- ถ้ามีหมายเลขคีย์ต่างกันจะ

สร้างคำสั่งต่อไปนี้

```
MOVE key_number TO KEYNUMBER.
CALL "_BTRV" USING BCLOSE,STAT,POSBLK(filenum+1),
      INfilenum,KEYBUFFfilenum-key_number,KEYNUMBER.
```

โดยที่ key_number หมายถึง หมายเลขคีย์ที่มีค่าน้อยที่สุด

filenum หมายถึง หมายเลขแฟ้มข้อมูล

4) สำหรับแฟ้มข้อมูล ISAM จะวนสร้างคำสั่งต่าง ๆ ต่อไปนี้ตามจำนวนแฟ้มข้อมูลทั้งหมด

```
CLOSE INPUT-FILEfilenum.
```

โดยที่ filenum หมายถึง หมายเลขแฟ้มข้อมูล

5) สำหรับแฟ้ม Text ประเภท COBOL Variable Length Record และ ประเภท COBOL Fixed Length Record จะสร้างคำสั่งต่อไปนี้

```
CLOSE INPUT-FILE.
```

6) สำหรับแฟ้ม Text ประเภท Other Fixed Length Record

(ก) ถ้ามีแฟ้มข้อมูล เพียงแฟ้มข้อมูล เดียว จะสร้างคำสั่งต่อไปนี้

```
CALL "_CLOSE" USING FHAND.
```

(ข) ถ้ามีแฟ้มข้อมูลมากกว่า 1 แฟ้ม จะสร้างคำสั่งต่อไปนี้ตามจำนวนแฟ้มข้อมูลทั้งหมด

```
CALL "_CLOSE" USING FHAND(filenum+1).
```

โดยที่ filenum หมายถึง หมายเลขแฟ้มข้อมูล

ค) การสร้างส่วนในการปิดแฟ้มข้อมูลออก และ สร้างส่วนจบโปรแกรมของภาษาโคบอล จะสร้างคำสั่งต่อไปนี้

```
CLOSE OUTPUT-FILE.
```

```
STOP RUN.
```

ช) การสร้างส่วน MAIN-RTN รุกิน เป็นรุกิน
หลักของโปรแกรมภาษาโคบอลที่ถูกสร้างขึ้น

(1) จะสร้างคำสั่งในการเคลื่อนย้ายเขตข้อมูลของแฟ้มข้อมูลหลักตามจำนวนเขตข้อมูลทั้งหมดที่ถูกกำหนดให้พิมพ์เป็น หัวชุด รายงาน หัวรายงาน หัวรายการ หัวรายงานย่อย รายการข้อมูล ท้ายรายงานย่อย ท้ายรายงาน ท้ายชุดรายงาน

```
MOVE fieldnameINO TO fieldname:levelname.
```

โดยที่ fieldname หมายถึง ชื่อเขตข้อมูลของแฟ้มข้อมูลหลัก
levelname หมายถึง ชื่อของระดับต่าง ๆ

(2) ถ้ามีแฟ้มข้อมูลมากกว่า 1 แฟ้ม จะ
ต้องทำการอ่านแฟ้มข้อมูลอื่น ๆ มาด้วย

(ก) สำหรับแฟ้มข้อมูล dBASE จะ
วนสร้างคำสั่งต่าง ๆ ต่อไปนี้ตาม จำนวนแฟ้มข้อมูลทั้งหมด -1

```
CALL "_REWIND" USING FHAND(filenum+1).
```

```
MOVE X"00000000" TO COUNTTOTALREC (filenum+1).
```

โดยที่ filenum หมายถึง หมายเลขแฟ้มข้อมูล

คู่มือวิทยากร
จุฬาลงกรณ์มหาวิทยาลัย

สร้างคำสั่งต่อไปนี้

```
CALL "_REWIND" USING FHANDNDX(filenumbe+1).
MOVE X"00" TO COUNTREC(filenumbe+1).
MOVE X"00" TO RECNO(filenumbe+1).
MOVE X"00" TO RECLEN.
CALL "_READHEAD" USING FNAMEfilenumbe, FHAND
      (filenumbe+1).
PERFORM READ-RTNfilenumbe UNTIL relation
      OR THERE-IS-NO-MORE-DATA.
IF THERE-IS-NO-MORE-DATA
      GO TO END-REPORT-RTN.
```

โดยที่ filenumbe หมายถึง หมายเลขแฟ้มข้อมูล
relation หมายถึง ความสัมพันธ์ระหว่างแฟ้มข้อมูล

(ข) สำหรับแฟ้มข้อมูล Foxbase

จะวนสร้างคำสั่งต่อไปนี้ตามจำนวนแฟ้มข้อมูลทั้งหมด -1

```
CALL "_REWIND" USING FHAND(filenumbe+1).
MOVE X"00000000" TO COUNTTOTALREC(filenumbe+1).
โดยที่ filenumbe หมายถึง หมายเลขแฟ้มข้อมูล
```

สร้างคำสั่งต่อไปนี้

```
CALL "_REWIND" USING FHANDIDX(filenumbe+1).
MOVE X"00" TO COUNTREC(filenumbe+1).
MOVE X"00" TO RECNO(filenumbe+1).
MOVE X"00" TO RECLen.
CALL "_READHEAD" USING FNAMEfilenumbe, FHAND
      (filenumbe+1).
PERFORM READ-RTNfilenumbe UNTIL relation
      OR THERE-IS-NO-MORE-DATA.
IF THERE-IS-NO-MORE-DATA
      GO TO END-REPORT-RTN.
```

โดยที่ filenumbe หมายถึง หมายเลขแฟ้มข้อมูล
relation หมายถึง ความสัมพันธ์ระหว่างแฟ้มข้อมูล

(ค) สำหรับแฟ้มข้อมูล Btrieve

จะวนสร้างคำสั่งต่อไปนี้ตามจำนวนแฟ้มข้อมูลทั้งหมด -1

```
MOVE keynumber TO KEYNUMBER.
```

โดยที่ keynumber หมายถึง หมายเลขคีย์

- ถ้ามีหมายเลขคีย์เดียวกัน

จะสร้างคำสั่งต่อไปนี้

MOVE fieldname TO KEYBUFFfilenumber.

โดยที่ fieldname หมายถึง ชื่อเขตข้อมูลที่เคยกำหนดจากคำสั่ง relation
filenumber หมายถึง หมายเลขแฟ้มข้อมูล

- ถ้ามีหมายเลขคีย์ต่างกัน

จะสร้างคำสั่งต่อไปนี้

MOVE fieldname TO KEYBUFFfilenumber-key_number.

โดยที่ fieldname หมายถึง ชื่อเขตข้อมูลที่เคยกำหนดจากคำสั่ง relation
filenumber หมายถึง หมายเลขแฟ้มข้อมูล
key_number หมายถึง หมายเลขคีย์ที่มีค่าน้อยที่สุด

จากนั้นจะสร้างคำสั่งดังนี้

PERFORM READ-RTNfilenumber.

IF BEOF GO TO END-REPORT-RTN.

โดยที่ filenumber หมายถึง หมายเลขแฟ้มข้อมูล

(ง) สำหรับแฟ้มข้อมูล ISAM จะ
วนสร้างคำสั่งต่าง ๆ ต่อไปนี้ตามจำนวนแฟ้มข้อมูลทั้งหมด -1

```
MOVE fieldname TO fieldnameINfilenumber.
```

```
PERFORM READ-RTNfilenumber.
```

```
IF THERE-IS-NO-MORE-DATA
```

```
    GO TO END-REPORT-RTN.
```

โดยที่ filenumber หมายถึง หมายเลขแฟ้มข้อมูล

fieldname หมายถึง ชื่อเขตข้อมูลที่เคยกำหนดจากคำสั่ง relation

(จ) สำหรับแฟ้ม Text ประเภท
COBOL Variable Length Record จะวนสร้างคำสั่งต่าง ๆ ต่อไปนี้

```
OPEN INPUT INPUT-FILEfilenumber.
```

```
PERFORM READ-RTNfilenumber UNTIL relation.
```

```
CLOSE INPUT-FILEfilenumber.
```

```
IF THERE-IS-NO-MORE-DATA
```

```
    GO TO END-REPORT-RTN.
```

โดยที่ filenumber หมายถึง หมายเลขแฟ้มข้อมูล

relation หมายถึง ความสัมพันธ์ระหว่างแฟ้มข้อมูล

(ฉ) สำหรับแฟ้มข้อมูล Text

ประเภท COBOL Fixed Length Record จะวนสร้างคำสั่งต่าง ๆ ต่อไปนี้

```

MOVE 0 TO COUNTTOTALREC(filenumbe+1).
PERFORM READ-RTNfilenumber UNTIL relation
    OR THERE-IS-NO-MORE-DATA.
IF THERE-IS-NO-MORE-DATA
    GO TO END-REPORT-RTN.

```

โดยที่ filenumber หมายถึง หมายเลขแฟ้มข้อมูล
 relation หมายถึง ความสัมพันธ์ระหว่างแฟ้มข้อมูล

(ช) สำหรับแฟ้มข้อมูล Text

ประเภท Other Fixed Length Record จะวนสร้างคำสั่งต่าง ๆ ต่อไปนี้

```

CALL "_REWIND" USING FHAND(filenumbe+1).
MOVE X"00000000" TO COUNTTOTALREC(filenumbe+1).
PERFORM READ-RTNfilenumber UNTIL relation
    OR THERE-IS-NO-MORE-DATA
GO TO END-REPORT-RTN.

```

โดยที่ filenumber หมายถึง หมายเลขแฟ้มข้อมูล
 relation หมายถึง ความสัมพันธ์ระหว่างแฟ้มข้อมูล

(3) สร้างส่วนในการเคลื่อนย้ายเขตข้อมูล
ที่ได้จากแฟ้มข้อมูลอื่น ๆ ไปยังตัวแปรของระดับต่าง ๆ จะวนสร้างหัวข้อต่อไปนี้ตาม
จำนวนแฟ้มข้อมูลทั้งหมด -1

```
MOVE fieldnameINfilenumber TO fieldname:levelname.
```

โดยที่ fieldname หมายถึง ชื่อเขตข้อมูล

filenumber หมายถึง หมายเลขแฟ้มข้อมูล

levelname หมายถึง ชื่อของระดับต่าง ๆ

(4) สร้างส่วนที่ใช้ในการคำนวณของระดับ
ต่าง ๆ จะวนสร้างคำสั่งต่าง ๆ ต่อไปนี้ตามจำนวนคำสั่ง COMPUTE ทั้งหมดของ
ระดับต่าง ๆ

(ก) การคำนวณโดยใช้ฟังก์ชัน

- การคำนวณหาค่าผลรวม

```
COMPUTE SUMvarname = SUMvarname + varname2
```

โดยที่ varname หมายถึง ชื่อตัวแปร

varname2 หมายถึง ชื่อตัวแปรหรือชื่อเขตข้อมูลหรือตัวเลข

- การคำนวณหาค่าตัวนับ

```
COMPUTE COUNTvarname = COUNTvarname + 1
```

โดยที่ varname หมายถึง ชื่อตัวแปร

- การคำนวณหาค่าเฉลี่ย

```
COMPUTE SUMvarname = SUMvarname + varname2
```

```
COMPUTE COUNTvarname = COUNTvarname + 1
```

โดยที่ varname หมายถึง ชื่อตัวแปร

varname2 หมายถึง ชื่อตัวแปรหรือชื่อเขตข้อมูลหรือตัวเลข

- การคำนวณหา ค่าความ

แปรปรวน และ ความเบี่ยงเบนมาตรฐาน

```
COMPUTE SUMvarname = SUMvarname + varname2
```

```
COMPUTE COUNTvarname = COUNTvarname + 1
```

```
COMPUTE SQRvarname = SQRvarname + (varname**2)
```

โดยที่ varname หมายถึง ชื่อตัวแปร

varname2 หมายถึง ชื่อตัวแปรหรือชื่อเขตข้อมูลหรือตัวเลข

นอกจากนี้แล้ว ประโยคคำสั่ง

COMPUTE ทุกคำสั่ง ที่ถูกสร้างขึ้นจะมีคำสั่งต่อไปนี้ด้วย

```
ON SIZE ERROR DISPLAY "SIZE OF varname ERROR"
```

```
GO TO CLOSE-FILES-RTN.
```

โดยที่ varname หมายถึง ชื่อตัวแปร



(๒) การคำนวณอื่น ๆ

```

COMPUTE varname = expression
      ON SIZE ERROR DISPLAY "SIZE OF varname ERROR"
      GO TO CLOSE-FILES-RTN.

```

โดยที่ varname หมายถึง ชื่อตัวแปร
 expression หมายถึง สูตรที่ใช้ในการคำนวณ

(5) สร้างส่วนสำหรับ การพิมพ์รายงาน

หน้าแรกจะสร้างคำสั่งต่อไปนี้

```

PERFORM levelname.

```

โดยที่ levelname หมายถึง ชื่อของระดับต่าง ๆ

ถ้ามีการกำหนดเขตข้อมูลที่จะใช้เป็น

เขตข้อมูลคอนโทรลเบรค

- ถ้ารายการข้อมูลและหัวรายงาน

ย่อยอยู่ในบรรทัดเดียวกัน จะสร้างคำสั่งต่อไปนี้

```

MOVE linenum TO LINENO.

```

โดยที่ linenum หมายถึง จำนวนบรรทัดของหัวรายงานย่อย

ถ้ารายการข้อมูลและหัวรายงานย่อย

ไม่อยู่ในบรรทัดเดียวกันจะสร้างคำสั่งต่อไปนี้

```
MOVE 0 TO LINENO
```

(6) สร้างส่วนในการขึ้นหน้าใหม่

```
IF LINENO >= linenumber
```

```
PERFORM NEWPAGE-RTN.
```

โดยที่ linenumber หมายถึง (จำนวนบรรทัดที่พิมพ์ในรายการข้อมูล
+ ระยะห่างของแต่ละรายการ)*จำนวนบรรทัดต่อหน้า

(7) สร้างส่วนในการพิมพ์ รายการข้อมูล

จะสร้างคำสั่งต่อไปนี้

(ก) ถ้ารายการข้อมูล และ

หัวรายงานย่อยอยู่ในบรรทัดเดียวกันจะสร้างคำสั่งต่อไปนี้

```
IF CHANDDE = 0
```

```
PERFORM DE-RTN
```

```
ELSE
```

```
MOVE 0 TO CHANDDE.
```

(ข) ถ้ารายการข้อมูล

และ

หัวรายงานย่อยไม่อยู่ในบรรทัดเดียวกันจะสร้างคำสั่งต่อไปนี้

PERFORM DE-RTN.

(8) สร้างส่วนในการ อ่านระเบียบต่อไป

ของแฟ้มข้อมูลหลัก จะสร้างคำสั่งต่อไปนี้

PERFORM READ-RTNO.

(9) ถ้ามีการสั่งพิมพ์ท้ายรายงานย่อยและ

มีการสั่งพิมพ์ค่าของเขตข้อมูลด้วย จะสร้างคำสั่งต่อไปนี้

PERFORM CTLBRKCF-RTN.

(10) ถ้ามีการกำหนดเขตข้อมูลสำหรับเป็น

เขตข้อมูลคอนโทรลเบรคจะสร้างคำสั่งต่อไปนี้

PERFORM CONTROL-RTN.

ณ) การสร้างส่วน CONTROL-RTN รุกั้น จะสร้างคำสั่งต่อไปนี้อยู่ตามจำนวนเขตข้อมูลคอนโทรลเบรค โดยเรียงความสำคัญจากมากไปน้อย

IF fieldnameCT NOT = fieldnameINO

โดยที่ fieldname หมายถึง ชื่อเขตข้อมูล

1) ถ้ามีการสั่งพิมพ์ ทำรายงานย่อย เนื่องจากเขตข้อมูลคอนโทรลเบรคตัวนี้ จะวนสร้างชื่อเขตข้อมูลตามจำนวนเขตข้อมูลคอนโทรลเบรคที่มีระดับความสำคัญน้อยกว่า หรือ เท่ากับ เขตข้อมูลคอนโทรลเบรคตัวปัจจุบัน

PERFORM CF-fieldnameRTN.

โดยที่ fieldname หมายถึง ชื่อเขตข้อมูล

2) ถ้ามีการสั่งขึ้นหน้าใหม่ เนื่องจากเขตข้อมูลคอนโทรลเบรคตัวนี้ จะสร้างคำสั่งต่อไปนี้อยู่

PERFORM NEW-PAGE-RTN.


3) สร้างส่วนในการเคลื่อนย้าย ค่าของข้อมูลไปยังเขตข้อมูลคอนโทรลเบรคจะสร้างคำสั่งต่อไปนี้อยู่

PERFORM CTRLBRK-RTN.

4) ถ้ามีการสั่งพิมพ์หัวรายงานย่อย เนื่องจาก
จากเขตข้อมูลคอนโทรลเบรคตัวนี้ จะวนสร้างชื่อเขตข้อมูลตามจำนวนเขตข้อมูล
คอนโทรลเบรคที่มีระดับความสำคัญน้อยกว่าหรือเท่ากับเขตข้อมูลคอนโทรลเบรคตัว
ปัจจุบัน

PERFORM CH-fieldnameRTN.

โดยที่ fieldname หมายถึง ชื่อเขตข้อมูล



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ญ) การสร้างส่วน READ-RTN รุกิน เป็นรุกิน

ในการอ่านแฟ้มข้อมูลต่าง ๆ

1) สำหรับแฟ้มข้อมูล dBASE จะสร้าง

คำสั่งต่อไปนี้ตามจำนวนแฟ้มข้อมูลทั้งหมด

READ-RTNfilename.

โดยที่ filename หมายถึง หมายเลขแฟ้มข้อมูล

ข้อมูลเดียว

(ก) ถ้ามีแฟ้มข้อมูลใช้งานเพียงแฟ้ม

- ถ้าแฟ้มข้อมูลใช้งาน มีแฟ้ม

ข้อมูลดัชนีด้วยจะสร้างคำสั่งต่อไปนี้

CALL "_READNDX" USING FHANDNDX, RECLENNDX, BUFFER, COUNTREC,
RECNO, FHAND, HEADERDBF, RECLEN, COUNTTOTALREC, TOTALREC.

- ถ้าแฟ้มข้อมูลใช้งาน ไม่มี

แฟ้มข้อมูลดัชนีจะสร้างคำสั่งต่อไปนี้

CALL "_READ" USING FHAND, BUFFER, RECLEN, COUNTTOTALREC,
TOTALREC.

นอกจากนี้ จะสร้างคำสั่งต่อไปนี้

```
MOVE BUFFER TO INO.
IF COUNTTOTALREC > TOTALREC
    MOVE 'N' TO MORE-DATA-SWITCH
GO TO END-REPORT-RTN.
```

- (ข) ถ้ามีแฟ้มข้อมูลใช้งานมากกว่า 1 แฟ้ม จะสร้างคำสั่งต่อไปนี้
- ถ้าแฟ้มข้อมูลใช้งาน มีแฟ้มข้อมูลดัชนีด้วย จะสร้างคำสั่งต่อไปนี้

```
CALL "_READNDX" USING FHANDNDX,(filenumber+1),RECLENNDX
(filenumber+1),BUFFER,COUNTREC(filenumber+1),RECNO
(filenumber + 1),FHAND(filenumber + 1),HEADERDBF
(filenumber + 1),RECLLEN,COUNTTOTALREC(filenumber+1),
TOTALREC(filenumber+1).
```

โดยที่ filenumber หมายถึง หมายเลขแฟ้มข้อมูล

- ถ้าแฟ้มข้อมูลใช้งาน ไม่มีแฟ้มข้อมูลดัชนี จะสร้างคำสั่งต่อไปนี้

```
CALL "_READ" USING FHAND(filenumber+1),BUFFER,RECLLEN,
COUNTTOTALREC(filenumber+1),TOTALREC(filenumber+1).
```

โดยที่ filenumber หมายเลขแฟ้มข้อมูล

นอกจากนี้ จะสร้างคำสั่งต่อไปนี้

```
MOVE BUFFER TO IN(filenum+1).
IF COUNTTOTALREC(filenum+1) > TOTALREC(filenum+1)
    MOVE 'N' TO MORE-DATA-SWITCH
    GO TO END-REPORT-RTN.
```

โดยที่ filenum หมายถึง หมายเลขแฟ้มข้อมูล

2) สำหรับแฟ้มข้อมูล Foxbase จะสร้างคำสั่งต่อไปนี้ตามจำนวนแฟ้มข้อมูลทั้งหมด

```
READ-RTNfilenum.
```

โดยที่ filenum หมายถึง หมายเลขแฟ้มข้อมูล

(ก) ถ้ามีแฟ้มข้อมูลใช้งานเพียงแฟ้ม

ข้อมูลเดียว

- ถ้าแฟ้มข้อมูลใช้งาน มีแฟ้ม

ข้อมูลดัชนีด้วยจะสร้างคำสั่งต่อไปนี้

```
CALL "_READIDX" USING FHANDIDX, RECLNIDX, BUFFER, COUNTREC,
    RECNO, FHAND, HEADERDBF, RECLN, COUNTTOTALREC, TOTALREC.
```

- ถ้าเพิ่มข้อมูลใช้งาน ไม่มี

เพิ่มข้อมูลดัชนีจะสร้างคำสั่งต่อไปนี้

```
CALL "_READ" USING FHAND, BUFFER, RECLN, COUNTTOTALREC,
TOTALREC.
```

นอกจากนี้จะสร้างคำสั่งต่อไปนี้

```
MOVE BUFFER TO INO.
IF COUNTTOTALREC > TOTALRECMOVE 'N' TO MORE-DATA-SWITCH
GO TO END-REPORT-RTN.
```

(ข) ถ้ามีเพิ่มข้อมูลใช้งานมากกว่า 1

เพิ่ม จะสร้างคำสั่งต่อไปนี้ตามจำนวนเพิ่มข้อมูลทั้งหมด

- ถ้าเพิ่มข้อมูลใช้งาน มีเพิ่ม

ข้อมูลดัชนีด้วยจะสร้างคำสั่งต่อไปนี้

```
CALL "_READIDX" USING FHANDIDX, (filenumber+1), RECLNIDX (
filenumber+1), BUFFER, COUNTREC(filenumber+1), RECNO (
filenumber + 1), FHAND(filenumber + 1), HEADERDBF (
filenumber + 1), RECLN, COUNTTOTALREC(filenumber+1),
TOTALREC(filenumber+1).
```

โดยที่ filenumber หมายถึง หมายเลขเพิ่มข้อมูล

- ถ้าเพิ่มข้อมูลใช้งาน ไม่มี

เพิ่มข้อมูลดัชนีจะสร้างคำสั่งต่อไปนี้

```
CALL "_READ" USING FHAND(filenumbe+1),BUFFER,RECLEN,
      COUNTTOTALREC(filenumbe+1),TOTALREC(filenumbe+1).
```

โดยที่ filenumber หมายถึง หมายเลขเพิ่มข้อมูล

นอกจากนี้จะสร้างคำสั่งต่อไปนี้

```
MOVE BUFFER TO IN(filenumbe+1).
IF COUNTTOTALREC(filenumbe+1) > TOTALREC(filenumbe+1)
  MOVE 'N' TO MORE-DATA-SWITCH
  GO TO END-REPORT-RTN.
```

โดยที่ filenumber หมายถึง หมายเลขเพิ่มข้อมูล

3) สำหรับเพิ่มข้อมูล Btrieve

(ก) สำหรับเพิ่มข้อมูลหลัก จะสร้าง

คำสั่งต่อไปนี้

```
READ-RTNO.
```

(1) ถ้ามีแฟ้มข้อมูลเพียงแฟ้ม

ข้อมูลเดียว

- ถ้ามีหมายเลขคีย์

เดียวกัน จะสร้างคำสั่งต่อไปนี้

```
MOVE keynumber TO KEYNUMBER.
CALL "_BTRV" USING GETNEXT, STAT, POSBLK, INO, LEN, KEYBUFFO,
KEYNUMBER.
```

โดยที่ keynumber หมายถึง หมายเลขคีย์

- ถ้ามีหมายเลขคีย์ต่าง

กัน จะสร้างคำสั่งต่อไปนี้

```
MOVE key_number TO KEYNUMBER.
CALL "_BTRV" USING GETNEXT, STAT, POSBLK, INO, LEN, KEYBUFFO-
key_number, KEYNUMBER.
```

โดยที่ key_number หมายถึง หมายเลขคีย์ที่มีค่าน้อยที่สุด

(2) ถ้ามีแฟ้มมากกว่า 1 แฟ้ม

- ถ้ามีหมายเลขคีย์

เดียวกันจะสร้างคำสั่งต่อไปนี้

```
MOVE keynumber TO KEYNUMBER.
CALL "_BTRV" USING GETNEXT, STAT, POSBLK(1), INO, LEN(1),
KEYBUFFO, KEYNUMBER.
```

โดยที่ keynumber หมายถึง หมายเลขคีย์

- ถ้ามีหมายเลขคีย์ต่าง

กันจะสร้างคำสั่งต่อไปนี้

```
MOVE key_number TO KEYNUMBER.
CALL "_BTRV" USING GETNEXT, STAT, POSBLK(1), INO, LEN(1),
KEYBUFF0-keynumber, KEYNUMBER.
```

โดยที่ keynumber หมายถึง หมายเลขคีย์

key_number หมายถึง หมายเลขคีย์ที่มีค่าน้อยที่สุด

(ข) สำหรับแฟ้มข้อมูลอื่น ๆ จะ

สร้างคำสั่งต่อไปนี้ตามจำนวนแฟ้มข้อมูล - 1

- ถ้ามีหมายเลขคีย์ เดียวกัน

จะสร้างคำสั่งต่อไปนี้

```
READ-RTNfilename.
MOVE keynumber TO KEYNUMBER.
CALL "_BTRV" USING GETEQUAL, STAT, POSBLK(filename+1),
INfilename, LEN(filename+1), KEYBUFFfilename,
KEYBUFFER.
```

โดยที่ filename หมายถึง หมายเลขแฟ้มข้อมูล

keynumber หมายถึง หมายเลขคีย์

- ถ้ามีหมายเลขคีย์ต่างกันจะ
วนสร้างคำสั่งต่าง ๆ ต่อไปนี้ตามจำนวนหมายเลขคีย์ที่แตกต่างกัน

```
READ-RTNfilename-keynumber.
MOVE keynumber TO KEYNUMBER.
CALL "_BTRV" USING GETEQUAL,STAT,POSBLK(filename+1),
      INfilename,LEN(filename+1),KEYBUFFfilename
      -keynumber,KEYBUFFER.
```

โดยที่ filename หมายถึง หมายเลขแฟ้มข้อมูล
keynumber หมายถึง หมายเลขคีย์

(ค) นอกจากนี้จะสร้างคำสั่งต่อไปนี้
- ถ้ามีแฟ้มข้อมูล เพียงแฟ้ม
ข้อมูลเดียวจะสร้างคำสั่งต่อไปนี้

```
MOVE BUFFER TO INO.
```

- ถ้ามีแฟ้มข้อมูลมากกว่า 1
แฟ้มจะสร้างคำสั่งต่อไปนี้

```
MOVE BUFFER TO IN(filename+1).
โดยที่ filename หมายถึง หมายเลขแฟ้มข้อมูล
```

นอกจากนี้ จะสร้างคำสั่งต่อไปนี้

```
IF BEOF
    GOTO END-REPORT-RTN.
```

4) สำหรับเพิ่มข้อมูล ISAM

(ก) สำหรับเพิ่มข้อมูลหลัก จะสร้าง

คำสั่งต่อไปนี้

```
READ-RTNO.
    READ INPUT-FILEO
    AT END MOVE 'N' TO MORE-DATA-SWITCH
    GO TO END-REPORT-RTN.
```

(ข) สำหรับเพิ่มข้อมูลอื่น ๆ จะสร้าง

คำสั่งต่อไปนี้ตามจำนวนเพิ่มข้อมูล - 1

```
READ-RTNfilename.
    READ INPUT-FILEfilename
    INVALID KEY DISPLAY "EROR KEY FIELD"
    MOVE 'N' TO MORE-DATA-SWITCH
    GO TO END-REPORT-RTN.
```

โดยที่ filename หมายถึง หมายเลขเพิ่มข้อมูล

5) สำหรับแฟ้มข้อมูล Text ประเภท COBOL Fixed Length Record จะสร้างคำสั่งต่อไปตามจำนวนแฟ้มข้อมูลทั้งหมด

```
READ-RTNfilename.
```

```
  READ INPUT-FILEfilename INTO INfilename
```

```
  AT END MOVE 'N' TO MORE-DATA-SWITCH
```

```
  GO TO END-REPORT-RTN.
```

โดยที่ filename หมายถึง หมายเลขแฟ้มข้อมูล

6) สำหรับแฟ้มข้อมูล Text ประเภท Other Fixed Length Record

(ก) ถ้ามีแฟ้มข้อมูล เพียงแฟ้มข้อมูล เดียว จะสร้างคำสั่งต่อไปนี้

```
READ-RTNO.
```

```
ADD 1 TO COUNTTOTALREC.
```

```
IF COUNTTOTALREC > TOTALREC GO TO END-REPORT-RTN.
```

```
READ INPUT-FILEO INTO INO.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

(ข) ถ้ามีแฟ้มข้อมูลมากกว่า 1 แฟ้ม

จะสร้างคำสั่งต่อไปนี้ตามจำนวนแฟ้มข้อมูลทั้งหมด

```

READ-RTNfilename.
ADD 1 TO COUNTTOTALREC(filename+1).
IF COUNTTOTALREC(filename+1) > TOTALREC(filename+1).
    GO TO END-REPORT-RTN.
READ INPUT-FILEfilename INTO INfilename.
โดยที่ filename หมายถึง หมายเลขแฟ้มข้อมูล

```

7) สำหรับแฟ้มข้อมูล Text ประเภท

Other Fixed Length Record

(ก) ถ้ามีแฟ้มข้อมูล เพียงแฟ้มข้อมูล

เดียวจะสร้างคำสั่งต่อไปนี้

```

READ-RTNO.
MOVE record size TO RECLEN.
CALL "_READ" USING FHAND, BUFFER, RECLEN, COUNTTOTALREC,
    TOTALREC.
MOVE BUFFER TO INO.
IF COUNTTOTALREC > TOTALREC
    MOVE 'N' TO MORE-DATA-SWITCH
    GO TO END-REPORT-RTN.
โดยที่ record size หมายถึง ความยาวของ 1 ระเบียบ

```

(ข) ถ้ามีแฟ้มข้อมูลมากกว่า 1 แฟ้ม

จะวนสร้างคำสั่งต่าง ๆ ต่อไปนี้ตามจำนวนแฟ้มข้อมูลทั้งหมด

```

READ-RTNfilename.
MOVE record size TO RECLEN.
CALL "_READ" USING FHAND(filename+1), BUFFER, RECLEN,
      COUNTTOTALREC(filename+1), TOTALREC(filename+1).
MOVE BUFFER TO INfilename.
IF COUNTTOTALREC(filename+1) > TOTALREC(filename+1)
  MOVE 'N' TO MORE-DATA-SWITCH
  GO TO END-REPORT-RTN.

```

โดยที่ record size หมายถึง ความยาวของ 1 ระเบียบ
filename หมายถึง หมายเลขแฟ้มข้อมูล

ฉ) การสร้างส่วนระดับต่าง ๆ จะวนสร้างคำสั่งต่าง ๆ ต่อไปนี้ ตามจำนวนระดับต่าง ๆ ทั้งหมด

1) สร้างชื่อของ รุกัน ระดับต่าง ๆ

(ก) สำหรับระดับ หัวชุดรายงาน
หัวรายงาน หัวรายการ รายการข้อมูลท้ายรายงาน ท้ายชุดรายงาน จะสร้างคำสั่งต่อไปนี้

```
levelname-RTN.
```

โดยที่ levelname หมายถึง ชื่อของระดับต่าง ๆ

(ข) สำหรับหัวรายงานย่อย

levelnameCH-RTN.

โดยที่ levelname หมายถึง ชื่อของระดับต่าง ๆ

(ค) สำหรับท้ายรายงานย่อย

levelnameCF-RTN.

โดยที่ levelname หมายถึง ชื่อของระดับต่าง ๆ

2) สร้างส่วนของการขึ้นบรรทัดใหม่

PERFORM NEWLINE-RTN VARYING LIECOUNT

FROM 1 BY 1 UNTIL LINECOUNT > lineno

โดยที่ lineno หมายถึง ระยะห่างระหว่างระดับก่อนหน้า กับ ระดับปัจจุบัน

3) ถ้ามีการสั่งคำนวณต่าง ๆ ในระดับนี้

(ก) การคำนวณโดยใช้ฟังก์ชัน

- การคำนวณหา ค่าผลรวม

และค่าตัวนับ

MOVE varname TO varname!levelname!levelnumber.

โดยที่ varname หมายถึง ชื่อตัวแปร

levelname หมายถึง ชื่อระดับ

- การคำนวณหาค่าเฉลี่ย

```

COMPUTE varname = varname/COUNTvarname
      ON SIZE ERROR DISPLAY "SIZE OF varname ERROR"
      GO TO CLOSE-FILES-RTN.

```

```

MOVE varname TO varname:levelname:levelnumber.

```

โดยที่ varname หมายถึง ชื่อตัวแปร

levelname หมายถึง ชื่อระดับ

levelnumber หมายถึง หมายเลขต่าง ๆ ของระดับนี้

- การคำนวณหาค่าความ

เบี่ยงเบนมาตรฐาน

```

COMPUTE STD = SQRTvarname / COUNTvarname - (varname /
      COUNTvarname)
      ON SIZE ERROR DISPLAY "SIZE OF varname ERROR"
      GO TO CLOSE-FILES-RTN.

```

```

MOVE 0 TO MINSTD.

```

```

MOVE maxvar TO MAXSTD.

```

```

MOVE 1 TO FLAGSTD.

```

```

MOVE 0 TO NOSTD.

```

```

PERFORM SQRT-RTN UNTIL FLAGSTD NOT = 1.

```

```

MOVE MIDSTD TO varname:levelname:levelnumber.

```

โดยที่ varname หมายถึง ชื่อตัวแปร

maxvar หมายถึง ค่าสูงสุดที่เป็นไปได้ของตัวแปร

levelname หมายถึง ชื่อระดับ

levelnumber หมายถึง หมายเลขต่าง ๆ ของระดับนี้

แปรปรวน

```
COMPUTE STD = SQRvarname / COUNTvarname - (varname /
COUNTvarname)
ON SIZE ERROR DISPLAY "SIZE OF varname ERROR"
GO TO CLOSE-FILES-RTN.
```

โดยที่ varname หมายถึง ชื่อตัวแปร

(ข) การคำนวณอื่น ๆ

```
MOVE varname TO varname:levelname:levelnumber.
```

โดยที่ varname หมายถึง ชื่อตัวแปร

levelname หมายถึง ชื่อระดับ

levelnumber หมายถึง หมายเลขต่าง ๆ ของระดับนี้

4) สร้างส่วนในการพิมพ์รายงาน

```
WRITE OUT-REC FROM levelname:levelnumber.
```

โดยที่ levelname หมายถึง ชื่อระดับ

levelnumber หมายถึง หมายเลขต่าง ๆ ของระดับนี้

5) ถ้ามีการสั่งคำนวณต่าง ๆ ในระดับ
นี้จะต้องมีการกำหนดค่าของตัวแปรต่าง ๆ ให้กลายเป็น 0

(ก) การคำนวณโดยใช้ฟังก์ชัน

- การคำนวณหา ค่าผลรวม

และค่าตัวนับ

MOVE 0 TO varname.

MOVE 0 TO varname:levelname:levelnumber.

โดยที่ varname หมายถึง ชื่อตัวแปร

levelname หมายถึง ชื่อระดับ

levelnumber หมายถึง หมายเลขต่าง ๆ ของระดับนี้

- การคำนวณหาค่าเฉลี่ย

MOVE 0 TO varname.

MOVE 0 TO COUNTvarname.

MOVE 0 TO varname:levelname:levelnumber.

โดยที่ varname หมายถึง ชื่อตัวแปร

levelname หมายถึง ชื่อระดับ

levelnumber หมายถึง หมายเลขต่าง ๆ ของระดับนี้

จุฬาลงกรณ์มหาวิทยาลัย

แปรปรวน และ ความเบี่ยงเบนมาตรฐาน

MOVE 0 TO varname.

MOVE 0 TO COUNTvarname.

MOVE 0 TO SQRvarname.

MOVE 0 TO varname:levelname:levelnumber.

โดยที่ varname หมายถึง ชื่อตัวแปร

levelname หมายถึง ชื่อระดับ

levelnumber หมายถึง หมายเลขต่าง ๆ ของระดับนี้

(ข) การคำนวณอื่น ๆ

MOVE 0 TO varname.

โดยที่ varname หมายถึง ชื่อตัวแปร

6) สำหรับรายการข้อมูล หัวรายงานย่อย

ท้ายรายงานย่อย จะสร้างคำสั่งต่อไปนี้

ADD linenumber TO LINENO.

โดยที่ linenumber หมายถึง จำนวนบรรทัดที่ถูกพิมพ์ในระดับนี้

7) ถ้ารายการข้อมูล และหัวรายงานย่อย
อยู่ในบรรทัดเดียวกัน จะสร้างคำสั่งต่อไปนี้

```
MOVE 1 TO CHANDDE.
```

ฉ) การสร้างส่วน CTLBRK-RTN รุกั้นจะสร้าง
รูกั้นนี้ก็ต่อเมื่อมีการกำหนดเขตข้อมูลคอนโทรลเบรค

1) สร้างส่วนในการเคลื่อนย้ายเขตข้อมูล
ของแฟ้มข้อมูลหลักไปยังเขตข้อมูลคอนโทรลเบรค จะสร้างคำสั่งต่อไปนี้ตามจำนวน
เขตข้อมูลคอนโทรลเบรคทั้งหมด

```
MOVE fieldnameINO TO fieldnameCT.
```

โดยที่ fieldname หมายถึง ชื่อเขตข้อมูล

2) ถ้ามีการสั่งพิมพ์หัวรายงานย่อย จะ
สร้างคำสั่งต่อไปนี้ตามจำนวนแฟ้มข้อมูลทั้งหมดที่ถูกกำหนดในคำสั่ง ch

```
MOVE variablename:filenumber TO variablenameCH.
```

โดยที่ variablename หมายถึง ชื่อตัวแปร

filenumber หมายถึง หมายเลขแฟ้มข้อมูล

๕) การสร้างส่วน CTLBRKCF-RTN รุกิน จะสร้างรูกินนี้ก็ต่อเมื่อมีการสั่งพิมพ์ท้ายรายงานย่อย จะสร้างคำสั่งต่อไปนี้ตามจำนวนแฟ้มข้อมูลทั้งหมดที่ถูกกำหนดในคำสั่ง cf

```
MOVE variablename:filenumber TO variablenameCF.
```

โดยที่ variablename หมายถึง ชื่อตัวแปร

filenumber หมายถึง หมายเลขแฟ้มข้อมูล

๖) การสร้างส่วน NEWPAGE-RTN รุกิน

1) ถ้ามีการสั่งพิมพ์ท้ายรายงาน จะสร้าง

คำสั่งต่อไปนี้

```
PERFORM PF-RTN.
```

```
PERFORM NEWLINE-RTN VARYING LINECOUNT
```

```
FROM 1 BY 1 UNTIL LINECOUNT > lineno.
```

โดยที่ lineno หมายถึงจำนวนบรรทัดต่อหน้า-บรรทัดสุดท้ายของท้ายรายงาน

2) ถ้าไม่มีการสั่งพิมพ์ท้ายรายงาน จะ

สร้างคำสั่งต่อไปนี้

```
COMPUTE LINENO = PAGELONG - LINENO - lineno.
```

```
PERFORM NEWLINE-RTN VARYING LINECOUNT
```

```
FROM 1 BY 1 UNTIL LINECOUNT > LINENO.
```

โดยที่ lineno หมายถึง บรรทัดแรกของระดับรายการข้อมูล

3) ถ้ามีการสั่งพิมพ์หัวรายงาน หัวรายการ
หัวรายงานย่อย จะสร้างคำสั่งต่อไปนี้

```
PERFORM levelname-RTN.
```

โดยที่ levelname หมายถึง ชื่อของระดับต่าง ๆ

4) ถ้ามีการสั่งพิมพ์หมายเลขหน้าจะสร้าง
คำสั่งต่อไปนี้

```
ADD 1 TO PAGENO.
```

5) ถ้าระยะห่าง ระหว่างรายการข้อมูล
รายการแรก และ ระดับก่อนหน้ามากกว่า 1 จะสร้างคำสั่งต่อไปนี้

```
PERFORM NEWLINE-RTN VARYING LINECOUNT
```

```
FROM 1 BY 1 UNTIL LINECOUNT > lineno
```

โดยที่ lineno หมายถึงระยะห่างระหว่างรายการข้อมูลรายการแรก และ
ระดับก่อนหน้า

6) สร้างส่วนในการกำหนด ค่าตัวแปร
LINENO ให้เป็น 0 จะสร้างคำสั่งต่อไปนี้

```
MOVE 0 TO LINENO.
```

๓) การสร้างส่วน SQRT-RTN รุกัน เป็นรุกัน
สำหรับการคำนวณหาค่ารากกำลังที่ 2 จะสร้างคำสั่งต่อไปนี้

SQRT-RTN.

ADD 1 TO NOSTD.

IF NOSTD > 50

MOVE 0 TO FLAGSTD.

COMPUTE MIDSTD = (MAXSTD+MINSTD)/2

ON SIZE ERROR DISPLAY "SIZE OF MIDSTD ERROR"

GO TO CLOSE-FILES-RTN.

COMPUTE DIVSTD = STD/MIDSTD

ON SIZE ERROR DISPLAY "SIZE OF DIVSTD ERROR"

GO TO CLOSE-FILES-RTN.

IF MIDSTD > DIVSTD

MOVE MIDSTD TO MAXSTD

MOVE DIVSTD TO MINSTD

ELSE

IF MIDSTD < DIVSTD

MOVE MIDSTD TO MINSTD

MOVE DIVSTD TO MAXSTD

ELSE

MOVE 0 TO FLAGSTD.

ณ) การสร้างส่วน NEWLINE-RTN รุกิน เป็น
รุกินสำหรับการขึ้นบรรทัดใหม่ จะสร้างคำสั่งต่อไปนี้

NEWLINE-RTN.

WRITE OUT-REC FROM NEW-LINE.

4.4.7 การสร้างส่วนจบโปรแกรมจะทำการปิดแฟ้มข้อมูล

ทุกประเภท



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

5. โปรแกรมจัดการแฟ้มข้อมูล

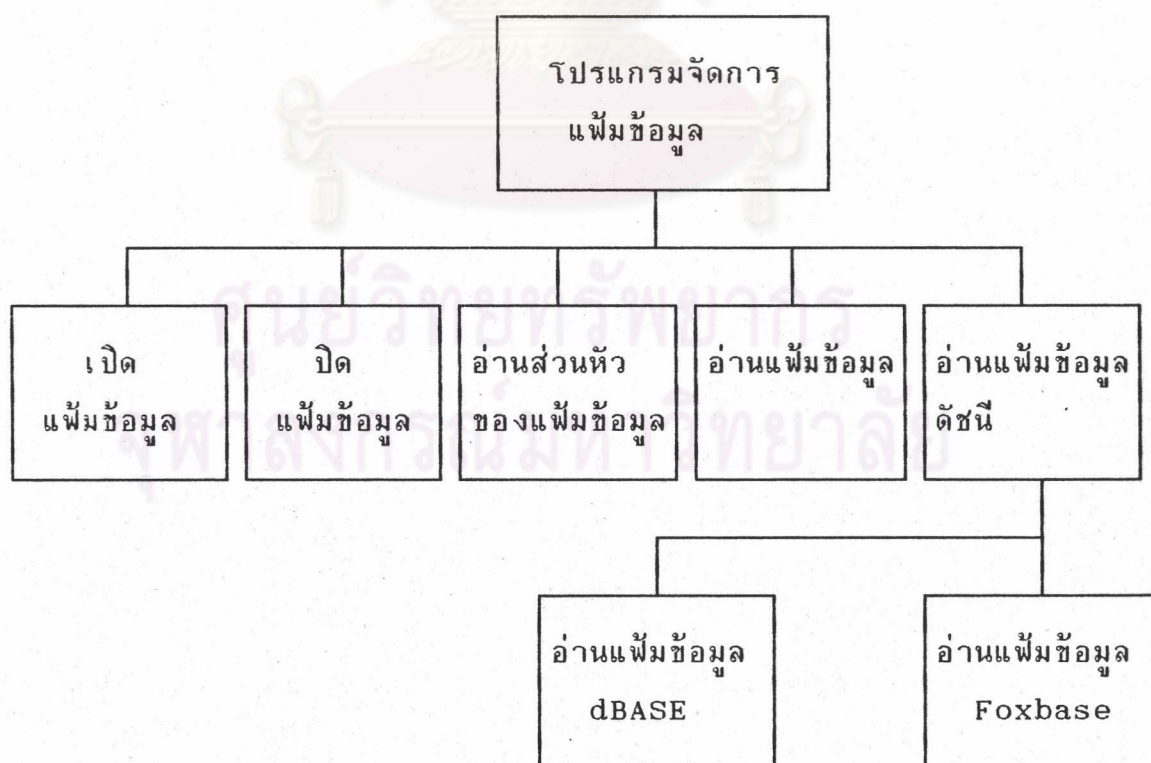
เนื่องจากภาษาโคบอลไม่สามารถอ่านข้อมูลจากแฟ้มข้อมูล dBASE Foxbase และ Text File บางประเภทได้เนื่องจากการจัดเก็บข้อมูลของแฟ้มข้อมูลต่าง ๆ เหล่านี้มีสัญลักษณ์บางตัวที่มีผลกระทบกับคำสั่ง READ ของภาษาโคบอล ดังนั้นจึงจำเป็นต้องทำการพัฒนาโปรแกรมจัดการแฟ้มข้อมูลขึ้นมา เพื่อให้โปรแกรมภาษาโคบอลที่สร้างขึ้นมานั้นสามารถเรียกใช้ข้อมูลจากแฟ้มข้อมูลประเภทต่าง ๆ ดังกล่าวข้างต้นได้ ซึ่งโปรแกรมจัดการแฟ้มข้อมูลนี้พัฒนาขึ้นโดยใช้ภาษาแอสเซมบลี 8088/8086 จากนั้นนำไปผ่านตัวแอสเซมเบลอร์ผลลัพธ์ที่ได้คือโปรแกรมภาษาโคบอลที่สามารถเชื่อมโยง (Link) เข้ากับภาษาโคบอลที่ถูกสร้างขึ้นได้

5.1 หน้าทีของโปรแกรมจัดการกับแฟ้มข้อมูล dBASE Foxbase และ Text File บางประเภทเพื่อให้สามารถอ่านข้อมูลขึ้นมาทำรายงานได้

5.2 ข้อมูลเข้า พารามิเตอร์จากโปรแกรมภาษาโคบอล

5.3 ข้อมูลออก ข้อมูลต่าง ๆ จากแฟ้มข้อมูลเพื่อส่งกลับไปให้โปรแกรมภาษาโคบอล

5.4 ขั้นตอนการประมวลผล การประมวลผลประกอบไปด้วยโปรแกรมย่อยต่าง ๆ แสดงได้ดังรูปที่ 4.5



รูปที่ 4.5 ผังโครงสร้างโปรแกรมจัดการแฟ้มข้อมูล

- 2.5.4.1 เปิดแฟ้มข้อมูล (Open.obj)
- 2.5.4.2 ปิดแฟ้มข้อมูล (Close.obj)
- 2.5.4.3 อ่านส่วนหัวของแฟ้มข้อมูล
(Readhead.obj)
- 2.5.4.4 อ่านแฟ้มข้อมูล (Read.obj)
- 2.5.4.5 อ่านแฟ้มข้อมูลดัชนีของ dBASE
(Readndx.obj)
- 2.5.4.5 อ่านแฟ้มข้อมูลดัชนีของ Foxbase
(Readidx.obj)



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย