

บทที่ 4

การออกแบบโปรแกรมหลักและโปรแกรมย่อย

ในบทที่ 3 ได้กล่าวถึงวิธีการออกแบบระบบในเชิงหน้าที่หลักและการแบ่งส่วนของระบบ ตลอดจนวิธีการจัดการข้อมูลต่าง ๆ ที่จำเป็นทั้งในรูปแบบของผังข้อมูล และตารางต่าง ๆ ในบทนี้จะมาแนะนำส่วนในส่วนของโปรแกรมหลัก และโปรแกรมย่อยที่ใช้ในแต่ละส่วน เพื่อแสดงถึงหน้าที่ของโปรแกรมหลักที่นี้ รวมทั้งตรวจวิธีที่น่าสนใจในการออกแบบและนำมาใช้ในการสร้างตัวตรวจสอบไวยากรณ์โปรแกรมภาษาโคบอล ซึ่งการนำเสนอ จะแบ่งออกเป็น 3 เฟสเช่นเดียวกับที่กับการนำเสนอในบทที่ 3

เฟสที่ 1 การวิเคราะห์เลขชี้ตัว

ในเฟสการวิเคราะห์เลขชี้ตัวนี้ มีส่วนประกอบของ โปรแกรม ดังนี้

1. โปรแกรมหลัก มีชื่อว่า SCAN เป็นโปรแกรมหลักในเฟสการวิเคราะห์เลขชี้ตัว โดย SCAN จะทำหน้าที่ต่าง ๆ ดังนี้
 - 1.1 จัดการเรื่องแสดงชื่อโปรแกรม (Program Local Stack) เพื่อใช้ในคำสั่ง PUSH และคำสั่ง POP ค่าของรีจิสเตอร์ต่าง ๆ ในขณะดำเนินการของโปรแกรม (Execution)
 - 1.2 รับผิดชอบแก้ไขข้อมูลโปรแกรมภาษาโคบอลที่จะทำการตรวจสอบ โดยการรับที่สามารถ ทำได้ 2 ทาง คือ
 - 1.2.1 รับเข้าสู่โปรแกรม ในลักษณะส่วนท้ายของคำสั่ง (Command Tail) คือ ป้อนชื่อผังข้อมูลต่อท้ายการเรียกใช้งานตัวตรวจสอบไวยากรณ์โปรแกรมภาษาโคบอล
 - 1.2.2 ถ้าไม่มีชื่อผังข้อมูลต่อท้ายการเรียกใช้ โปรแกรมจะจัดการขึ้นหน้าจอเพื่อถามชื่อผังข้อมูลที่จะตรวจสอบ จากผู้ใช้งาน

1.3 เปิดแฟ้มข้อมูล โปรแกรมตีพิมพ์ภาษา โดบออล ที่บันทึกอยู่ในแฟ้มงานแม่เหล็ก ในรูปแบบ IBM-3470

1.4 ตรวจสอบแฟ้มข้อมูลในแฟ้มงานแม่เหล็ก ที่ใช้ทำงานว่าแฟ้มข้อมูลที่ตรวจ (CBLSK1.PRN, CBLSK3.ERR, CBLSK4.TKN) มีอยู่แล้วในสำเนาของแฟ้มงานแม่เหล็กหรือไม่ ถ้ามีอยู่แล้ว ก็จะลบออก แล้วสร้างที่ใหม่

2. โปรแกรมย่อย ส่วนของโปรแกรมย่อยในแฟ้มที่ 1 สามารถจะแสดงได้ดังนี้

2.1 READI ทำหน้าที่อ่านข้อมูล โปรแกรมตีพิมพ์ภาษา โดบออล ที่ถูกเก็บบันทึกอยู่ในแฟ้มงานแม่เหล็ก รูปแบบ IBM-3470 โดยไปเรียกใช้สำเนาโปรแกรมของระบบบันทึกข้อมูล "โดยทำ" ที่ชื่อ DSMRD ทำหน้าที่อ่านข้อมูลขึ้นมาทีละ 1 เซกเตอร์ ซึ่งมีความยาว 128 ไบต์

2.2 COPYPRN ทำหน้าที่คัดลอก โปรแกรมตีพิมพ์จากแฟ้มข้อมูลตั้งต้นที่อยู่ในรูปแบบ IBM-3470 มาไปลงบันทึกในแฟ้มข้อมูล ของระบบที่พิมพ์แล้ว ที่ชื่อ CBLSK1.PRN โดยยังคงรูปแบบการบันทึก คล้ายคลึงกับ IBM-3470 คือ บันทึก 1 ระเบียบที่อ่านได้จากแฟ้มข้อมูลของ IBM ลงไปแฟ้มข้อมูล CBLSK1.PRN ระเบียบ 1 ระเบียบเหมือนกัน โดยใช้พื้นที่ในแฟ้มงานแม่เหล็ก 1 เซกเตอร์ ระเบียบเดียวกัน โดยใช้พื้นที่ 128 ไบต์ เพื่อบันทึกคำสั่งภาษาโดบออล 1 บรรทัด

ในการถ่ายถอดจากแฟ้มข้อมูล IBM ลงไปยัง CBLSK1.PRN ต้องมีการเรียกใช้ โปรแกรมสร้างเลขลำดับเพื่อการอ้างอิงด้วย (SEQGEN)

2.3 SEQGEN เป็นโปรแกรมที่เมื่อถูกเรียกแต่ละครั้งจะมีผลทำให้ตัวนับบรรทัดเพิ่มค่าขึ้นทีละ 1 และในกรณีที่ตัวนับบรรทัดเพิ่มค่า จะเปลี่ยนค่าในตัวนับค่าซึ่งถูกเก็บในรูปแบบเลขฐานสอง ให้เป็นค่าเลขฐานสิบที่สามารถแสดงได้ด้วยรหัส ASCII

2.4 COL7 เป็นโปรแกรมย่อยที่ทำหน้าที่ 2 ประการ ได้แก่

2.4.1 ตรวจสอบเลขลำดับ ที่ผู้เขียนโปรแกรมอาจจะเขียนเข้ามาในคอลัมน์ 1-6 ว่าการเรียงลำดับของแต่ละบรรทัดถูกต้องหรือไม่ โดยการเก็บเลขลำดับบรรทัดที่อ่านพบในขณะนี้ เปรียบเทียบกับเลขลำดับบรรทัดก่อนหน้าเพื่อดูว่าเพิ่มขึ้นหรือไม่ พังทึน ถ้าหากไม่เพิ่มขึ้น ก็จะมีว่าเกิดความผิดพลาดทึน

2.4.2 ตรวจสอบตัวอักษรที่อยู่ในคอลัมน์ 7 ว่าเป็นตัวอักษรที่ถูกต้องหรือไม่ โดยตัวอักษรที่ถูกต้องเป็นไปได้ คือ

- + เป็นบรรทัดหมายเหตุ
- - เป็นบรรทัดต่อค่าคงที่
- / เป็นการบอกตัวแปรโปรแกรม ให้สร้างรหัสใช้ควบคุมเครื่องพิมพ์ เพื่อให้ทันสมัยกว่ากระดาษใหม่เมื่อพิมพ์ผล
- D บอกว่าคำสั่งในบรรทัดได้ใช้ช่วยในการค้นหาที่ผิดพลาดที่ละบรรทัด (Debugging Line) โดยใช้ควบคู่กับรหัส WITH DEBUGGING MODE ในย่อหน้า OBJECT-COMPUTER

2.5 ERROR เป็นโปรแกรมทำงานที่บันทึกหรือส่งความผิดพลาดพร้อมทั้งหมายเลขบรรทัดที่ผิดพลาดลงในแฟ้มข้อมูล CBLSK3.ERR โดย 1 ระเบียบแถวแฟ้มข้อมูลจะสามารถเก็บรหัสแสดงความผิดพลาดได้ 8 สมาชิก โดยที่แต่ละสมาชิก ใช้เนื้อที่ในการเก็บ 16 ไบต์

2.6 LEXIC เป็นโปรแกรมที่ทำงานที่วิเคราะห์เลขซีเตล เพื่อจัดสร้างโทเกนขึ้นโดยใช้ตัวบ่งจุดสิ้นสุดโทเกน (Delimiter) ได้แก่ ช่องว่าง, เครื่องหมายวรรคตอน (Punctuation) เป็นต้น ทศจุดสิ้นสุดของโทเกน แล้วจะส่งเอาโทเกน ที่ได้ส่ง ไปเก็บที่แฟ้มข้อมูล CBLSK4.TKN

- 2.7 WRITEB เป็นโปรแกรมย่อยที่ถูกเรียกใช้โดย LEXIC ทำหน้าที่บันทึกข้อมูลลงในแฟ้มข้อมูล (BLSK4.TKN) โดยก่อนการบันทึก ต้องมีการเติมค่า OFFH ลงในฟิลด์ของรีเฟเฟอร์ก่อน แล้วจึงสามารถนำข้อมูลที่อยู่ในรีเฟเฟอร์นั้น ไปบันทึกลงในแฟ้มตามแรมเหล็ก
- 2.8 CHKBUF เป็นโปรแกรมย่อยที่ถูกเรียกใช้โดย LEXIC ทำหน้าที่ตรวจสอบว่ามีฟิลด์ของ BLSK4.TKN (BUFFER) ยังคงเหลือที่ว่างก่อนที่จะนำเลขไบนารีใหม่เก็บลงไปได้หรือไม่ ถ้าไม่ได้ ก็จะเรียก WRITEB ให้ทำการบันทึกข้อมูลในแฟ้มฟิลด์ลงในแฟ้มข้อมูล BLSK4.TKN แล้วจึงเคลื่อนย้ายไบนารีใหม่ที่ส่งลงสู่รีเฟเฟอร์
- 2.9 CLBUF3 เป็นโปรแกรมย่อยที่ถูกเรียกใช้โดย SCAN ในตอนสิ้นสุดของเฟสการวิเคราะห์หลักถัดไป เพื่อจัดรีเฟเฟอร์ให้เรียบร้อยก่อนนำข้อมูลในรีเฟเฟอร์สุดท้ายที่เก็บมาทำการวัดความผิดพลาดที่ส่งเก็บมาบันทึกในแฟ้มข้อมูล โดยก่อนการเก็บ ถ้าหากมีรีเฟเฟอร์ที่ยังคงเหลือที่ว่าง จะมีการเติมตัวอักษร "+" ลงในฟิลด์นั้นๆ ให้เต็ม เสียก่อน
- 2.10 CLBUF4 เป็นโปรแกรมย่อยที่ถูกเรียกใช้โดย SCAN ในตอนสิ้นสุดของเฟสการวิเคราะห์หลักถัดไป เพื่อจัดรีเฟเฟอร์ให้เรียบร้อยก่อนนำข้อมูลในรีเฟเฟอร์สุดท้ายที่เก็บมาทำการวัดความผิดพลาดในแฟ้มข้อมูล BLSK4 โดยก่อนการเก็บมาทำการวัด จะตรวจสอบว่าในรีเฟเฟอร์ยังมีที่ว่างหรือไม่ ถ้าหากยังมีที่ว่าง จะเติมค่า OFFH ให้เต็มก่อนแล้วจึงจัดการย้ายเก็บรีเฟเฟอร์นั้นๆ เก็บลงในแฟ้มข้อมูล BLSK4.TKN

เฟสที่ 2 การวิเคราะห์ความถูกต้องของไวยากรณ์ภาษา

ในเฟสนี้ ประกอบด้วยโปรแกรมต่าง ๆ ดังต่อไปนี้

1. CHECK เป็นส่วนโปรแกรมเริ่มต้นของเฟสที่ 2 โดยทำหน้าที่ต่าง ๆ ได้แก่
 - 1.1 จัดการเรียกและแสดงผลโปรแกรม เพื่อให้ใบคำขึง PUSH และคำสั่ง POP ของเครื่องแต่ละตัวต่าง ๆ ในขณะที่ดำเนินการโปรแกรมในเฟสที่ 2 นี้
 - 1.2 เปิดแฟ้มข้อมูลที่ใช้ในเฟสที่ 2 คือ CBLSK4.TKN
 - 1.3 ตรวจสอบว่าโปรแกรมมีข้อมูล CBLSK3.ERR อยู่หรือไม่ ถ้าหากมีอยู่ก็จะทำการลบออก แล้วสร้างแฟ้มใหม่
 - 1.4 ตรวจสอบว่าโปรแกรมที่ได้รับเข้าสู่เฟสที่ 2 นี้ เป็นโปรแกรม "ID" หรือ "IDENTIFICATION" หรือไม่ ถ้าใช่ก็จะส่งต่อการทำงานให้โปรแกรม IDDIV ต่อไป แต่ถ้าไม่ใช่ก็จะสร้างรหัสความผิดพลาด แล้วจึงส่งต่อการทำงานให้โปรแกรม IDDIV ซึ่งการกระทำดังกล่าว เปรียบเสมือนเป็นการสมมติ (Assume) ว่ามีชื่อ DIVISION นี้อยู่
2. IDDIV เป็นส่วนย่อยที่ทำหน้าที่ตรวจสอบลำดับและความถูกต้องที่ควรจะเป็นของโปรแกรมที่จะต้องเป็นตามลำดับโครงสร้างใน IDENTIFICATION DIVISION
3. ENVIRO เป็นส่วนย่อยที่ทำหน้าที่ตรวจสอบลำดับและความถูกต้องที่ควรจะเป็นของโปรแกรมที่จะต้องเป็นตามลำดับโครงสร้างภายใน ENVIRONMENT DIVISION
1. DATADV เป็นส่วนของโปรแกรมที่ทำงานในส่วนนี้เกี่ยวกับ DATA DIVISION โดยแบ่งเป็น
 - 4.1 ตรวจสอบความถูกต้องที่ควรจะเป็นของลำดับโปรแกรม ที่ต้องเป็นตามลำดับของโครงสร้างโปรแกรมภายในส่วนของ DATA DIVISION
 - 4.2 เก็บโปรแกรมที่เป็นสัญลักษณ์เข้าสู่ตารางสัญลักษณ์
5. PROCDE เป็นโปรแกรมหลักที่โปรแกรมของเฟสที่ 2 ทำหน้าที่รับรู้โปรแกรมเข้ามาแล้วตรวจสอบว่าเป็นโปรแกรมที่เป็นคำสั่งประเภทคำสั่งหรือไม่ ถ้าใช่ก็จะเรียกส่วนของโปรแกรมย่อยต่าง ๆ เพื่อตรวจสอบความถูกต้อง ของการเขียนคำสั่ง

แต่ละคำสั่งนั้น เช่นเมื่อพบโทเคน "ADD" ก็จะเรียกส่วนของโปรแกรมย่อยให้
ตรวจความถูกต้องของคำสั่ง ADD

6. FRNOTE เป็นส่วนโปรแกรมย่อย ที่ทำหน้าที่รับรู้โทเคน แล้วจะส่งรหัสกลับมาให้
แก่โปรแกรมที่เรียกใช้ โดยส่งสิ่งต่าง ๆ เหล่านี้

6.1 รหัสคำสั่ง ส่งกลับค่าตัวแปร NOTE โดยถ้าไม่ใช่คำสั่งจะส่งค่า
OFEH OOH กลับมา

6.2 เนื้อที่โทเคนตั้งอยู่ โดยส่งกลับค่าตัวแปร AREA

6.3 ความยาวของโทเคน โดยส่งกลับค่าตัวแปร LENGTH

6.4 สัญลักษณ์ว่าเป็นคำสั่งหรือไม่ สัญลักษณ์ดังกล่าว เพื่อทดสอบว่าโทเคนที่
พบนั้นเป็นคำสั่งประเภทคำสั่งหรือไม่ว่าง โดยถ้าใช่จะส่งค่า "T" กลับมา
ในตำแหน่งที่ VERB

โปรแกรมย่อย FRNOTE จะเป็นส่วนที่ควบคุมข้อมูลโทเคนในบัฟเฟอร์ (DMA)
ด้วย ถ้าหากข้อมูลในบัฟเฟอร์มีไม่พอ ก็จะตรวจสอบดูว่า มีข้อมูลระเบียบใหม่ใน
บัฟเฟอร์ของโปรแกรมหรือไม่ (BUFFER) ถ้ามีอยู่ ก็จะเคลื่อนย้ายข้อมูลจาก
BUFFER มายัง DMA แต่ถ้าไม่มีก็จะอ่านข้อมูลระเบียบใหม่จากแผ่นแม่เหล็ก
เข้ามาถึง DMA โดยตรงเลย นอกจากนั้นแล้ว ทุกครั้งที่มีการเรียกใช้ FRNOTE
ตัวชี้จะถูกลื่อนไปชี้ถึงโทเคนตัวถัดไปทันที

7. AHEAD เป็นส่วนโปรแกรมย่อยที่ทำงานที่เหมือน FRNOTE ทุกประการ เพียงแต่
ไม่มีตัวชี้ตัวชี้ (Pointer) ไปยังโทเคนถัดไป

8. DINAME เป็นส่วนโปรแกรมย่อยที่ทำงานที่ตรวจสอบว่าโทเคนนั้นถูกต้องตามกฎเกณฑ์
การตั้งชื่อตัวแปร โทเคนหาได้หรือไม่ โดยกฎเกณฑ์ต่าง ๆ คือ

8.1 ความยาวของชื่อตัวแปร ต้องยาวไม่เกิน 30 ตัวอักษร

8.2 ต้องมีตัวอักษร A - Z อย่างน้อย 1 ตัวอักษร

8.3 มีตัวอักษรพิเศษได้เพียง - อย่างเดียวเท่านั้น และตัวอักษรพิเศษ

ตั้งกลุ่ม ให้เมเป็นตัวเลขหรือตัวสุดท้ายของชื่อตัวแปร

8.4 ให้เมเป็นตัวเลข

9. DATNAM เป็นส่วนโปรแกรมย่อยที่ทำหน้าที่ค้นหาตัวแปรในตารางสัญลักษณ์ หลังจากที่ได้เก็บสัญลักษณ์เข้าสู่ตารางแล้ว เพื่อใช้ในการตรวจสอบความถูกต้องในส่วนของ PROCEDURE DIVISION ถ้าพบการค้นหาในตารางสัญลักษณ์ที่พบว่าไม่มีสัญลักษณ์ที่ต้องการอยู่ ก็หมายความว่า ตัวแปรนั้น ๆ ได้มีการสร้างหรือมีการกำหนดขึ้นในส่วนของ DATA DIVISION เรียบร้อยแล้ว ก่อนมีการเรียกใช้ในส่วนของ PROCEDURE DIVISION ผลลัพธ์ที่ได้จากการค้นหาในส่วนของโปรแกรมย่อยนี้ จะถูกส่งกลับมาในเรจิสเตอร์ A
10. KPTSYM เป็นส่วนโปรแกรมย่อยที่ถูกเรียกใช้ในส่วนของ DATADV เพื่อเก็บค่าโหนดที่เป็นสัญลักษณ์ ที่เกี่ยวข้องโปรแกรมกำหนดขึ้น เข้าสู่ตารางสัญลักษณ์ ในการเก็บค่าโหนดเข้าสู่ตารางสัญลักษณ์ จะมีการเก็บข่าวสารต่าง ๆ เข้าสู่ตารางสัญลักษณ์ด้วย ตัวอย่างของข่าวสารดังกล่าว คือ ความยาว, ประเภทของสัญลักษณ์ และ แอดเดรสของโหนด
11. PICTUR เป็นส่วนโปรแกรมย่อยที่ถูกเรียกใช้ในส่วนของ DATADV เพื่อตรวจสอบข่าวสารจากการใช้ชื่อ PICTURE ถูกต้องหรือไม่ โดยถ้าถูกต้อง จะส่งค่ากลับไปยังโปรแกรมที่เรียกใช้ คือ ประเภทของตัวแปร และความยาวของค่าที่ตัวแปรนั้นสามารถรับได้ (Field Length)
12. CONVRT เป็นส่วนโปรแกรมย่อยที่ใช้เปลี่ยนค่าจากเลขฐานสิบ ให้กลายเป็นค่าตัวเลขฐานสอง จะถูกเรียกใช้โดยส่วนโปรแกรม PICTUR เพื่อทราบถึงความยาวของค่าที่ตัวแปรนั้น จะเป็นไปได้
13. TRACK เป็นส่วนโปรแกรมย่อยที่ถูกเรียกใช้ในส่วน DATADV เพื่อเก็บแอดเดรส

ของต้นแปรใบเตยารางสัญญาลักษณตามหมายเลขลำดับ เฉพาะตัวที่จับคู่กันเกิดที่ใน
ขณะนั้น โดยเก็บแอดเดรสดังกล่าวเข้าสู่ตารางเก็บตัวที่ เพื่อนำไปบันทึกเป็น
ตัวที่ซึ่งเป็นส่วนหนึ่งของสมาชิกที่เก็บอยู่ในเตยารางสัญญาลักษณ

แปลที่ 3 การสังเคราะห์ข่าวสารความถี่ขนาด

ในการสังเคราะห์ข่าวสารความถี่ขนาดนั้น สามารถแบ่งส่วนของโปรแกรมออกเป็น
ส่วนประกอบต่าง ๆ ได้ดังนี้

1. MSSGE เป็นส่วนโปรแกรมหลัก ที่ทำหน้าที่เริ่มต้น ก่อนการเข้าสู่กระบวนการ
สังเคราะห์ข่าวสารความถี่ขนาด โดยหน้าที่เหล่านี้ ได้แก่

1.1 เปิดแฟ้มข้อมูล CBLSK1.PRN

1.2 ค้นหาในสารบัญชาระบบของแอดเดรสแม่เหล็ก ว่ามีแฟ้มข้อมูล CBLSK9.PRN
อยู่ในแอดเดรสแม่เหล็กหรือไม่ ถ้าหากมีอยู่ ก็จะลบออกแล้วสร้างที่ใหม่

2. READ เป็นส่วนโปรแกรมย่อย ที่ทำหน้าที่อ่านข้อมูลโปรแกรมดิบ จากแฟ้มข้อมูล
CBLSK1.PRN และในเสี้ยวนี้จะทำการตัดลอกข้อมูล จากแฟ้มข้อมูลดังกล่าวเข้าสู่
แฟ้มข้อมูล CBLSK9.PRN โดยจะต้องมีการจัดรูปแบบข้อมูล ที่อ่านได้จากแฟ้ม
ข้อมูล CBLSK1.PRN เสียใหม่ เนื่องจากข้อมูลใน CBLSK1 อยู่ในลักษณะของ
ความยาวคงที่ 128 ไบต์ โดยใช้จริงเพียง 80 ไบต์ และเหลือทิ้ง 48 ไบต์
ดังนั้นจะต้องละเลยไบบิตของช่องว่างที่อยู่ท้ายระเบียบนอกไปทั้งหมด ซึ่งเรา
จะทำให้เหลือที่ว่างของระเบียบข้อมูลได้ยาวไม่ถึง 80 ไบต์ เพื่อที่จะประหยัด
เนื้อที่ในแอดเดรสแม่เหล็ก และเมื่อทำการดังกล่าวแล้วจะต้องเติมค่า ODH และ
OAH เพื่อบอกจุดจบของระเบียบข้อมูลด้วย ตามลักษณะการบันทึกระเบียบข้อมูลของ
ระบบที่อื่น

การทำงานจะกระทำซ้ำนี้เรื่อยไป จนกว่าจะหมดข้อมูลใน CBLSK1.PRN แล้ว
จึงจะทำการปิดแฟ้มข้อมูลดังกล่าว ก่อนจะส่งต่อการทำงานให้กับส่วนโปรแกรมที่
อยู่ถัดไป



3. READ1 คือส่วนโปรแกรมย่อยที่ทำหน้าที่ต่าง ๆ ดังนี้
 - 3.1 เปิดแฟ้มข้อมูล CBLSK3.ERR ซึ่งเป็นแฟ้มข้อมูลที่มีเท็กรหัสความผิดพลาดจากเฟสการวิเคราะห์ผลเชิงตัวเลข
 - 3.2 อ่านข้อมูลรหัสความผิดพลาดเข้าสู่เนื้อที่การทำงานเพื่อเตรียมเรียงลำดับ (DATA) จนกว่าจะหมดข้อมูลใหม่แฟ้มข้อมูล CBLSK3.ERR
 - 3.3 ในระยะเปรียบเทียบค่าของ CBLSK3.ERR อาจจะมีการเติมตัวอักษร "+" ในท้ายระยะเปรียบเทียบเดิม ดังที่มีในโปรแกรมส่วนนี้ ต้องละเลยเครื่องหมายดังกล่าว เพื่อระกักับตัวที่แสดงเตรสไปยังสมาชิกรหัสความผิดพลาดที่เป็นตัวสุดท้ายที่แท้จริง
 - 3.4 ปิดแฟ้มข้อมูล CBLSK3.ERR แล้วส่งการควบคุมการทำงานไปยังส่วนของโปรแกรมที่อยู่ถัดไป

1. READ2 เป็นส่วนโปรแกรมย่อยที่มีหน้าที่เหมือนกับ READ1 ทุกประการ แต่จะทำการที่จะจัดการกับแฟ้มข้อมูลที่มีเท็กรหัสความผิดพลาด CBLSK5.ERR พร้อมทั้งเก็บตัวที่แสดงเตรสของสมาชิกรหัสความผิดพลาดตัวสุดท้าย เพื่อนำไปใช้ในารจัดเรียงลำดับข้อมูลรหัสความผิดพลาดต่อไป

5. SORT เป็นส่วนโปรแกรมย่อยที่ทำารจัดเรียงลำดับข้อมูลรหัสความผิดพลาด แต่ละละละตามลำดับสอง หมายเลขตามบรรทัดล่างลง โดยวิธีการจัดเรียงลำดับนั้นใช้วิธีการเปลี่ยนที่ (Interchange Sort) ซึ่งข้อมูลรหัสความผิดพลาดที่จะจัดเรียงนั้น อยู่ในเนื้อที่ DATA ที่อยู่ต่อจากส่วนของโปรแกรม การจัดเรียงลำดับนี้ใช้หลักการวิธีฟองอากาศ (Bubble Sort) โดยมีส่วนโปรแกรมย่อยที่ชื่อ SWAP ทำหน้าที่ที่สลับค่าของข้อมูล 2 สมาชิก ที่จะต้องเปลี่ยนที่กัน เมื่อทำการจัดเรียงลำดับแล้ว จะส่งค่าการทำงานไปยังส่วนของโปรแกรมที่อยู่ถัดไป

6. GENER คือส่วนโปรแกรมย่อยที่ทำหน้าที่ต่าง ๆ ดังนี้

- 6.1 อ่านข้อมูลรหัสความถี่ของเวลาที่เรียงลำดับแล้วเข้ามาทีละ 1 สมาชิก
- 6.2 นำรหัสความถี่ของเวลาที่อ่านเข้ามา ได้ที่มี ไปค้นหา ในตารางข่าวสารความถี่ของเวลา
- 6.3 การค้นหา ที่มี เมื่อพบแล้ว จะอ่านข้อมูลความยาวของข่าวสารความถี่ของเวลาที่เข้ามาได้ 1 ไบต์ เพื่อค้นหาความยาวของข่าวสารความถี่ของเวลาที่เข้ามาได้ 1 ไบต์ จะต้องอ่านจากตารางข่าวสารความถี่ของเวลา ใช้สื่อบันทึกที่เตรียมไว้ก่อนที่จะย้ายลงแฟลปเฟอ์ เมื่อเตรียมบันทึกลงบนแผ่นแม่เหล็ก โนแฟปข้อมูล CBLSK9.PRN ดังไป
7. WRITE เป็นส่วนของโปรแกรมย่อยที่ถูกเรียกใช้โดย READ และ GENER เพื่อจัดการแฟลปเฟอ์ของแฟ้มข้อมูล CBLSK9.PRN โดยจะตรวจดูว่า แฟลปเฟอ์ของแฟ้มข้อมูลดังกล่าวเต็ม 128 ไบต์หรือยัง ถ้าเต็มแล้ว จะบันทึกข้อมูลที่อยู่ในแฟลปเฟอ์ลงสู่แผ่นแม่เหล็ก ก่อนจะเคลื่อนย้ายข้อมูลที่ต้องการลงสู่แฟลปเฟอ์
8. SCHEAD เป็นส่วนของโปรแกรมย่อยที่สร้างหัวของรายงาน โดยตัวอักษรตัวแรกของระเบียบหัวรายงาน จะต้องเติมตัวอักษร OBH ซึ่งจะนำไปเครื่องพิมพ์ที่เห็นการกระทำชนิดใหม่ (Form Feed) นอกจากนั้นในส่วนของ SCHEAD จะทำหน้าที่นับเลขประจำหน้า นรวมทั้งเปิดหน้าต่างของเลขดังกล่าว ซึ่งเก็บในลักษณะของตัวเลขฐานสอง ให้กลายเป็นรูปแบบที่สามารถแสดงได้ตามปรกติของเลขฐานสิบ

นอกจากส่วนของโปรแกรมต่าง ๆ ที่กล่าวมาแล้ว การเรียกใช้อุปกรณ์รับส่งข้อมูล (Input-Output Device) โปรแกรมจะเรียกใช้ส่วนจัดการอุปกรณ์รับส่งข้อมูลของระบบซีพีเอ็ม คือส่วนของ บิตอส ที่ถูกจัดสรรไว้ตั้งแต่แอดเดรส 05H ในหน่วยความจำ

การเรียกใช้แผ่นแม่เหล็กในระบบ IBM-3470 โปรแกรมเรียกใช้ส่วนจัดการอุปกรณ์รับส่งข้อมูล โดยเฉพาะแผ่นแม่เหล็ก สำหรับระบบการอ่านของระบบเตรียมข้อมูล "ไทยท่า" ซึ่งเป็นส่วนโปรแกรมย่อยชื่อ DSMRD

ส่วนของโปรแกรมทั้งหมดที่กล่าวมาแล้วนั้น เป็นส่วนหลัก ๆ ที่อาจจะแบ่งนำมาแสดงได้อย่างชัดเจนเท่าที่สม ใในทางปฏิบัติ ยังมี โปรแกรมย่อยอีกบาง โปรแกรมที่มี นำมาแสดงไว้ในที่นี้ เนื่องจากเป็นส่วนที่เล็กมาก และถูกจัดเป็นส่วนย่อย ๆ ของโปรแกรมย่อยต่าง ๆ ที่กล่าวมาแล้วทั้งหมด