

บทที่ 5

สรุปและข้อเสนอแนะ

จากการทดสอบการเรียกใช้ด้วยโปรแกรมภาษาต่าง ๆ ตลอดจนการศึกษาค้นคว้า และวิจัยสามารถสรุปผลได้ดังนี้

1. ผลจากการทดสอบต่าง ๆ การเรียกใช้โดยวิธีที่แตกต่างกัน มีผลต่อโปรแกรมสำเร็จรูปและผู้ใช้งานแตกต่างกันดังนี้คือ

1. การเรียกใช้โดยคำสั่งควบคุม EXEC เป็นวิธีที่ง่ายและสะดวกต่อผู้ใช้งาน และโมดูลที่เข้ามาปฏิบัติการจะเป็นโมดูลที่จำเป็นต้องใช้ในการทำงานของการเรียงลำดับเท่านั้น

2. การเรียกใช้โดยโปรแกรมภาษาโคบอล ผู้ใช้งานเพียงศึกษาวิธีการเขียนโปรแกรมตามหลักที่ภาษาโคบอลกำหนดเท่านั้น ไม่จำเป็นต้องศึกษาคำสั่งควบคุมของการเรียงลำดับ แต่โปรแกรมสำเร็จรูปจะทำงานได้ช้ากว่าวิธีปกติ เนื่องจากการรับข้อมูลที่จะเรียงลำดับนั้น โปรแกรมจะใช้จุดออก E15 โดยมีตัวแปลภาษาโคบอลเป็นผู้เปิดแฟ้มข้อมูล นำเข้าและส่งแต่ละระเบียนให้โปรแกรมสำเร็จรูป

3. การเรียกใช้โดยโปรแกรมภาษาแอสเซมบลี เป็นวิธีการติดต่อโดยตรงและใกล้ชิดกับการทำงานของโปรแกรมสำเร็จรูปมากที่สุด สำหรับผู้ใช้งานที่ทราบถึงรายละเอียดเกี่ยวกับการทำงานของแต่ละโมดูลของโปรแกรมสำเร็จรูปและบริเวณพีพีไอแล้ว จะสามารถปรับปรุงโปรแกรมสำเร็จรูปนี้ได้



4. การพัฒนาการเรียกใช้โดยภาษาฟอร์แทรนนั้น ผู้ใช้งานจะต้องศึกษาคำสั่งควบคุมของการเรียงลำดับ โดยจัดเรียงข้อมูลและคำสั่งควบคุมการเรียงลำดับดังอธิบายในบทที่ 4 ในหัวข้อ 4.1 และ 4.2 และในกรณีที่มีข้อมูลนำเข้าเป็นบัตรเจาะรู จะต้องเสียเวลาในการถ่ายข้อมูลจากบัตรเจาะรูไปเก็บในจานแม่เหล็กก่อนจึงเรียงลำดับ

2. โปรแกรมสำเร็จรูปประกอบด้วยโมดูลต่าง ๆ ทั้งหมด 80 โมดูล และโมดูลใดจะเข้ามาทำงานขึ้นกับลักษณะที่ผู้ใช้งานกำหนด โมดูลทั้งหมดเรียงตามลำดับตัวอักษรคือ

| Object Module | Control Section | ชื่อในซีไอแอล | Object Module | Control Section | ชื่อในซีไอแอล |
|------------------|---------------------|---------------|------------------|---------------------|---------------|
| 1. ILSSASA | ILSSASA | SORTASA | 41.ILSSRCT | ILSSRCT | SORTRCT |
| 2. ILSSASF | ILSSASF | SORTASF | 42.ILSSRDA | ILSSADA, ILSSRDA | SORTRDA |
| 3. ILSSASG | ILSSASG | SORTASG | 43.ILSSRDB | ILSSADB, ILSSRDB | SORTRDB |
| 4. ILSSASH | ILSSASH | SORTASH | 44.ILSSRDC | ILSSADC, ILSSRDC | SORTRDC |
| 5. ILSSASK | ILSSASK | SORTASK | 45.ILSSRGA | ILSSAGA, ILSSRGA | SORTRGA |
| 6. ILSSASL | ILSSASL | SORTASL | 46.ILSSRGB | ILSSAGB, ILSSRGB | SORTRGB |
| 7. ILSSASP | ILSSASP | SORTASP | 47.ILSSRGD | ILSSAGD, ILSSRGD | SORTRGD |
| 8. ILSSPPI | ILSSPPI | - | | | |
| 9. ILSSRAA | ILSSAAA, ILSSRAA | SORTRAA | | | |
| 10. ILSSRAB | ILSSAAB, ILSSRAB | SORTRAB | | | |

| Object Module | Control Section | ชื่อในซีไอแอล | Object Module | Control Section | ชื่อในซีไอแอล |
|------------------|---------------------|---------------|------------------|---------------------|---------------|
| 11.ILSSRAC | ILSSAAC, ILSSRAC | SORTRAC | 48.ILSSRGE | ILSSAGE, ILSSRGE | SORTRGE |
| 12.ILSSRAD | ILSSAAD, ILSSRAD | SORTRAD | 49.ILSSRGF | ILSSAGF, ILSSRGF | SORTRGF |
| 13.ILSSRBA | ILSSABA, ILSSRBA | SORTRBA | 50.ILSSRGG | ILSSAGG, ILSSRGG | SORTRGG |
| 14.ILSSRBB | ILSSABB, ILSSRBB | SORTRBB | 51.ILSSRGH | ILSSAGH, ILSSRGH | SORTRGH |
| 15.ILSSRBC | ILSSABC, ILSSRBC | SORTRBC | 52.ILSSRGI | ILSSAGI, ILSSRGI | SORTRGI |
| 16.ILSSRBD | ILSSABD, ILSSRBD | SORTRBD | 53.ILSSRGJ | ILSSAGJ, ILSSRGJ | SORTRGJ |
| 17.ILSSRBE | ILSSABE, ILSSRBE | SORTRBE | 54.ILSSRMC | ILSSRMC, ILSSRMD | SORTRMC |
| 18.ILSSRBF | ILSSABF, ILSSRBF | SORTRBF | 55.ILSSROA | ILSSAOA, ILSSROA | SORTROA |
| 19.ILSSRBG | ILSSABG, ILSSRBG | SORTRBG | 56.ILSSROB | ILSSAOB, ILSSROB | SORTROB |
| 20.ILSSRBH | ILSSABH, ILSSRGH | SORTRBH | 57.ILSSROC | ILSSAOC, ILSSROC | SORTROC |

| Object Module | Control Section | ชื่อในซีไอแอล | Object Module | Control Section | ชื่อในซีไอแอล |
|------------------|---------------------|---------------|------------------|---------------------|---------------|
| 21.ILSSRBI | ILSSABI, ILSSRBI | SORTRBI | 58.ILSSROD | ILSSAOD, ILSSROD | SORTROD |
| 22.ILSSRBJ | ILSSABJ, ILSSRBJ | SORTRBJ | 59.ILSSROE | ILSSAOE, ILSSROE | SORTROE |
| 23.ILSSRBV | ILSSABV, ILSSRBV | SORTRBV | 60.ILSSROF | ILSSAOF, ILSSROF | SORTROF |
| 24.ILSSRCA | ILSSRCA | SORTRCA | 61.ILSSROG | ILSSAOG, ILSSROG | SORTROG |
| 25.ILSSRCB | ILSSRCB | SORTRCB | 62.ILSSRPA | ILSSAPA, ILSSRPA | SORTRPA |
| 26.ILSSRCC | ILSSRCC, ILSSRCN | SORTRCC | 63.ILSSRPB | ILSSAPB, ILSSRPB | SORTRPB |
| 27.ILSSRCD | ILSSRCD | SORTRCD | 64.ILSSRPC | ILSSAPC, ILSSRPC | SORTRPC |
| 28.ILSSRCE | ILSSRCE | SORTRCE | 65.ILSSRPD | ILSSAPD, ILSSRPD | SORTRPD |
| 29.ILSSRCF | ILSSRCF | SORTRCF | 66.ILSSRPE | ILSSAPE, ILSSRPE | SORTRPE |
| 30.ILSSRCG | ILSSRCG | SORTRCG | 67.ILSSRPF | ILSSAPF, ILSSRPF | SORTRPF |
| 31.ILSSRCH | ILSSRCH | SORTRCH | 68.ILSSRPG | ILSSAPG, ILSSRPE | SORTRPG |
| 32.ILSSRCI | ILSSRCI | SORTRCI | | | |
| 33.ILSSRCJ | ILSSRCJ | SORTRCJ | | | |
| 34.ILSSRCK | ILSSRCK | SORTRCK | | | |
| 35.ILSSRCL | ILSSRCL | SORTRCL | | | |
| 36.ILSSRCM | ILSSRCM | SORTRCM | | | |
| 37.ILSSRCO | ILSSRCO | SORTRCO | | | |
| 38.ILSSRCP | ILSSRCP | SORTRCP | | | |

| Object Module | Control Section | ชื่อในซีไอแอล | Object Module | Control Section | ชื่อในซีไอแอล |
|------------------|--------------------|---------------|------------------|--------------------|---------------|
| 39.ILSSRCQ | ILSSRCQ | SORTRCQ | 69.ILSSRSD | ILSSRSD | SORTRSD |
| 40.ILSSRCS | ILSSRCS | SORTRCS | 70.ILSSRSE | ILSSRSE | SORTRSE |
| | | | 71.ILSSRSG | ILSSRSG | SORTRSG |
| | | | 72.ILSSRSH | ILSSRSH | SORTRSH |
| | | | 73.ILSSRSI | ILSSRSI | SORTRSI |
| | | | 74.ILSSRSJ | ILSSRSJ | SORTRSJ |
| | | | 75.ILSSRSM | ILSSRSM | SORTRSM |
| | | | 76.ILSSRSN | ILSSRSN | SORTRSN |
| | | | 77.ILSSRSP | ILSSRSP | SORTRSP |
| | | | 78.ILSSRSQ | ILSSRSQ | SORTRSQ |
| | | | 79.ILSSRSR | ILSSRSR | SORTRSR |
| | | | 80.ILSSRTMG | ILSSRTMG | SORT |

3. ลักษณะที่ดีของโปรแกรมสำเร็จรูปคือ

1. วิธีการเรียงลำดับที่โปรแกรมสำเร็จรูปใช้เป็นวิธีการเรียงลำดับภายนอกหน่วยความจำ อาศัยอุปกรณ์อื่นเป็นที่เก็บข้อมูลชั่วคราว เหมาะกับงานที่มีข้อมูล ซึ่งต้องการเรียงลำดับเป็นจำนวนมาก

2. ในแต่ละเฟสของโปรแกรมสำเร็จรูปแบ่งออกเป็นโมดูล ซึ่งแต่ละโมดูลทำหน้าที่แตกต่างกัน ทำให้สามารถตัดแปลงโมดูลเหล่านั้นไปใช้ในงานอื่นภายหลังได้ (วิธีการนำโมดูลที่ต้องการจากซีไอแอลดั่งแสดงในการทดสอบที่ 5)

3. การเลือกใช้วิธีการเรียงลำดับแบบเลือกแทนที่นี้ ทำให้สามารถทำงานได้พร้อมกัน ในการอ่าน เรียงลำดับภายในหน่วยความจำ และการเขียนข้อมูลผลลัพธ์ นอกจากนี้โปรแกรมสำเร็จรูปยังใช้กลวิธีในการทำนายล่วงหน้าว่าบัฟเฟอร์ข้อมูลนำเข้าใดจะหมดก่อน โดยเตรียมข้อมูลในบัฟเฟอร์อีกบัฟเฟอร์หนึ่งสำรองไว้ให้ ดังนั้น จึงไม่เกิดเหตุการณ์ที่ข้อมูลในบัฟเฟอร์ข้อมูลนำเข้าหมดพร้อม ๆ กัน ซึ่งถ้าเกิดเหตุการณ์นี้แล้ว จะทำให้การทำงานต้องชะงักลง เนื่องจากต้องรอให้โมดูลซึ่งทำหน้าที่ในการอ่านข้อมูล อ่านข้อมูลให้เต็มบัฟเฟอร์เสียก่อน จึงจะเริ่มงานต่อไปได้ (การดูว่าข้อมูลในบัฟเฟอร์ใดจะหมดก่อน จะดูจากคีย์ตัวสุดท้ายในบัฟเฟอร์นั้น ๆ)

4. โปรแกรมสำเร็จรูปพยายามใช้เนื้อที่ให้น้อยที่สุดในหน่วยความจำสำหรับเก็บโปรแกรม โดยการทำ overlay ทั้งคำสั่งและส่วนโปรแกรม ตลอดจนการเขียนส่วนโปรแกรมที่ทำหน้าที่อย่างเดียวกันหลายโปรแกรม เช่นประกอบด้วยโมดูลที่รวมแฟ้มข้อมูล 1-5 แฟ้มข้อมูล และโมดูลที่รวมแฟ้มข้อมูลตั้งแต่ 1-8 แฟ้มข้อมูลเพื่อขยายบริเวณที่ใช้ในการเรียงลำดับ

4. สิ่งที่ควรแก้ไขเพื่อเพิ่มประสิทธิภาพของโปรแกรมสำเร็จรูปคือ

1. จากการศึกษาจากโปรแกรมดิบ (Source Program) ของโปรแกรมสำเร็จรูปพบว่ามีการใช้บริเวณที่เก็บข้อมูลชั่วคราวมากเกินไป คือจะต้องใช้ประมาณ ๒ เท่าของขนาดของแฟ้มข้อมูลนำเข้า เพื่อจะอ่านข้อมูลจากส่วนหนึ่งไปบันทึกลงบนอีกส่วนหนึ่ง ซึ่งในกรณีที่มีบริเวณที่เก็บข้อมูลน้อยกว่าที่โปรแกรมต้องการเพียงเล็กน้อย ก็ไม่สามารถเรียงลำดับได้

2. โปรแกรมสำเร็จรูปต้องใช้แฟ้มข้อมูล ซึ่งเป็นที่เก็บข้อมูลชั่วคราวเสมอ¹ แม้ว่าแฟ้มข้อมูลนำเข้าและหน่วยความจำจะมีขนาดใดก็ตาม ซึ่งในกรณีที่มีข้อมูลน้อย และมีเนื้อที่ในหน่วยความจำเพียงพอ ควรเรียงลำดับในหน่วยความจำ เพราะสามารถทำงานได้เร็วกว่า

3. ผู้ใช้งานไม่สามารถเรียงลำดับแฟ้มข้อมูล ซึ่งอยู่บนอุปกรณ์ต่างประเทศได้² แม้ว่าจะแฟ้มข้อมูลชนิดเดียวกัน ทำให้ต้องเสียเวลาในการถ่ายข้อมูล

4. เขตข้อมูลที่เป็นคีย์ ซ้อนกันไม่ได้ ทำให้ไม่สะดวกในการทำงานเท่าที่ควร

5. ข้อเสนอแนะสำหรับผู้ใช้งาน

1. เมื่อมีอุปกรณ์น้อยจะสามารถใช้วิธีต่อไปนี้ช่วยในการเรียงลำดับคือ

ก. พูลลิง (Pooling) หมายถึงการใช้เนื้อที่ของอุปกรณ์ที่มีอยู่ให้ทำหน้าที่มากกว่าหนึ่งหน้าที่ เช่นเป็นทั้งบริเวณที่เป็นที่เก็บข้อมูลนำเข้าและเป็นทั้งบริเวณที่ใช้เป็นที่เก็บข้อมูลชั่วคราวด้วย เป็นต้น ซึ่งผู้ใช้งานจะต้องปฏิบัติดังนี้คือ

¹
IBM Nordic Laboratory, "DOS/VS Sort/Merge Programmer's Guide" (IBM Thailand, 1973), p. 13.

²
Ibid., p. 16.

| บริเวณที่จะใช้ร่วมกันของงานแม่เหล็ก | เงื่อนไขในการปฏิบัติ |
|--|---|
| 1. ใช้เป็นที่เก็บแฟ้มข้อมูลนำเข้า แฟ้มข้อมูลผลลัพธ์ และที่เก็บข้อมูลชั่วคราว | ไม่สามารถทำได้ |
| 2. ใช้เป็นที่เก็บแฟ้มข้อมูลนำเข้าและแฟ้มข้อมูลผลลัพธ์ | ต้องกำหนดให้เป็นบริเวณเดียวกันและกำหนด "file ID" ในคำสั่ง DLBL เหมือนกัน |
| 3. แฟ้มข้อมูลนำเข้าและที่เก็บข้อมูลชั่วคราว | ไม่สามารถทำได้ |
| 4. แฟ้มข้อมูลผลลัพธ์และที่เก็บข้อมูลชั่วคราว | <ol style="list-style-type: none">1. ให้แฟ้มข้อมูลผลลัพธ์เป็นบริเวณเดียวกับบริเวณที่เก็บข้อมูลชั่วคราวบริเวณแรก และจะต้องเริ่มและจบลงที่แทรคและไซลินเดอร์เดียวกันบนอุปกรณ์เดียวกัน2. ไม่มีการกำหนด EXIT ในคำสั่งควบคุม OUTFIL3. ต้องมีบริเวณที่เก็บข้อมูลชั่วคราวอย่างน้อย 2 บริเวณ4. ต้องมี "file ID" ในคำสั่ง DLBL เหมือนกันกับบริเวณที่ใช้ร่วมกัน |

| บริเวณที่ใช้ร่วมกันของเทปแม่เหล็ก | เงื่อนไขในการปฏิบัติ |
|---|--|
| 1. ใช้เป็นที่เก็บแฟ้มข้อมูลนำเข้า แฟ้มข้อมูลผลลัพธ์และที่เก็บข้อมูลชั่วคราว | ไม่สามารถทำได้ |
| 2. ใช้เป็นที่เก็บแฟ้มข้อมูลนำเข้าและแฟ้มข้อมูลผลลัพธ์ | ไม่มีการกำหนด EXIT ในคำสั่ง INPFIL และ OUTFIL |
| 3. ใช้เป็นที่เก็บแฟ้มข้อมูลนำเข้าและที่เก็บข้อมูลชั่วคราว | 1. ให้ไปใช้ร่วมกับบริเวณที่ใช้เป็นที่เก็บข้อมูลชั่วคราวมันสุดท้าย 2. ไม่มีการกำหนด EXIT ในคำสั่ง INPFIL |
| 4. ใช้เป็นที่เก็บแฟ้มข้อมูลของผลลัพธ์และใช้เป็นที่เก็บข้อมูลชั่วคราว | 1. ให้ไปใช้ร่วมกับเทปมันที่เป็นที่เก็บข้อมูลชั่วคราวมันแรก 2. ไม่มีการกำหนด EXIT ในคำสั่ง OUTFIL |

ข. การใช้ที่ของอุปกรณ์ร่วมกัน นั่นคือ หลังจากทีอ่านแฟ้มข้อมูลแรกจบแล้วก็นำเทปมันนั้นออกไป แล้วใส่เทปมันใหม่ตรงที่ที่เคยใส่เทปมันเดิม

2. การเพิ่มจำนวนข้อมูลที่ต้องการจะเรียงลำดับของเทป สามารถเพิ่มจาก 1 มัน เป็น 2 มันได้ โดยการระบุ ALTKW ในคำสั่งควบคุม OPTION

3. การกำหนดขนาดของ L_5 ซึ่งเป็นความยาวเฉลี่ยในคำสั่ง Record (ความยาวเฉลี่ยของข้อมูลส่วนใหญ่ของระเบียนที่มีความยาวไม่คงที่) ถ้าไม่ทราบขนาดที่แท้จริงควรประมาณให้ต่ำกว่าความเป็นจริง และควรประมาณค่าของ L_4 ซึ่งเป็นความยาวต่ำสุดด้วย

4. การที่ผู้ใช้งานกำหนดจุดออกให้แก่โปรแกรมสำเร็จรูป จะมีผลต่อเวลาที่ใช้ในการทำงานของโปรแกรมสำเร็จรูป โดยปกติการประมวลผลส่วนโปรแกรมของผู้ใช้งานในแต่ละส่วนจะไม่เสียเวลามากนัก แต่ถ้าเป็นจุดออก E15, E25, E32 และ E35 ซึ่งทำหน้าที่ในการอ่านข้อมูลแต่ละระเบียนจากแฟ้มข้อมูลและถ้าแฟ้มข้อมูลมีขนาดใหญ่แล้ว จะต้องเสียเวลามาก นอกจากนี้โปรแกรมของผู้ใช้งานยังทำให้เสียเนื้อที่ในหน่วยความจำ ซึ่งอาจนำไปใช้ประโยชน์ในการเรียงลำดับ

5. การกำหนดคำสั่งต่าง ๆ ที่ทำให้จำนวนข้อมูลลดลงมีผลดีต่อการทำงานคือ

ก. การกำหนดคำสั่ง ADDROUT จะทำให้ลดเวลาในการเรียงลำดับและการเขียนข้อมูลลงบนแฟ้มข้อมูลผลลัพธ์

ข. คำสั่ง INCLUDE/OMIT จะทำให้เวลาที่ใช้ในการเคลื่อนข้อมูลและเวลาของ CPU ลดลง

ค. คำสั่ง SUM จะทำให้ลดจำนวนระเบียนที่จะต้องเรียงลำดับ ทำให้เวลาที่ใช้ในการเคลื่อนข้อมูลและเวลาของ CPU ลดลง

ง. คำสั่ง OUTREC จะทำให้ขนาดของระเบียนลดลง ทำให้เวลาในการเคลื่อนข้อมูลลดลง

6. การเรียกโมดูลต่าง ๆ ที่ทำหน้าที่ตรงกับความต้องการมาใช้งาน ควรศึกษาอย่างละเอียด และดัดแปลงให้เข้ากับโปรแกรมผู้ใช้งาน โดยการไลดบริเวณ ฟิฟไอก่อน ตรวจสอบเขตข้อมูลใน ฟิฟไอที่โมดูลนั้นจำเป็นต้องใช้งาน นำข้อมูลไปเก็บก่อน ต่อจากนั้นจึงไลดโมดูลที่ต้องการเข้ามาและให้โมดูลนั้นทำงาน

7. ถ้าต้องการเรียกใช้โปรแกรมสำเร็จรูปโดยโปรแกรมภาษาฟอร์แทรน หรือ แอลเซมบลิ ควรกำหนดบริเวณที่จะใช้เรียงลำดับให้อยู่ส่วนล่างของหน่วยความจำหลัก โดยการแยกออกเป็นเฟสสุดท้าย เพื่อไม่ให้โปรแกรมสำเร็จรูปใช้เนื้อที่บริเวณที่เตรียมไว้สำหรับโมดูลที่ทำหน้าที่เกี่ยวกับการพิมพ์ข้อมูล

8. การเรียกใช้โปรแกรมสำเร็จรูปโดยภาษาโคบอล ทำให้เสียเวลาในการทำงานมาก เนื่องจากตัวแปลของภาษาโคบอล กำหนดให้โปรแกรมออก ณ จุดออก E15 เพื่อรับข้อมูลเข้า ดังนั้น ทุกครั้งที่ต้องการข้อมูลในการเรียงลำดับโปรแกรมสำเร็จรูป จะต้องส่งการควบคุมออกนอกโปรแกรมเสมอ