บทที่ 2

ระบบการจัดการหน่วยความจำ

2.1 แนวคิดเรื่องหน่วยความจำเสมือน (Virtual Storage Concept)

จากระบบการจัดการหน่วยความจำแบบอื่น ๆ เช่น แบบซิงเกิลอโลเคชัน (Single Allocation)[1] ซึ่งเป็นการจัดให้แต่ละงานที่เข้ามาประมวลผลใช้เนื้อที่ทั้งหมดในหน่วยความจำหลัก (Main Storage) ดังรูปที่ 2.1 แบบกำหนดขนาดและตำแหน่งของพาติชันแบบคงที่ (Fixed Partition)[2] ซึ่งใช้ในระบบการทำงานแบบมัลติโปรแกรมมิง (Multiprogramming) โดยทำการแบ่งเนื้อที่ในหน่วยความจำหลักออกเป็นส่วน ๆ แต่ละส่วนจะถูกใช้โดยงาน 1 งานเท่านั้น และตำแหน่งของพาติชันจะไม่สามารถโยกย้ายไปมาภายในหน่วยความจำหลักได้ ดังรูปที่ 2.2 หรือแบบพาติชันที่สามารถโยกย้ายตำแหน่งไปมาได้ (Relocatable Partition)[3] ซึ่งได้ใช้ในระบบการทำงานแบบมัลติโปรแกรมมิงเช่นเดียวกัน ดังรูปที่ 2.3 ไม่ว่าจะเป็นระบบการจัดการหน่วยความจำแบบใดดังกล่าวมา ก็ยังคงประสบปัญหาบางประการอยู่ ได้แก่[4]

2.1.1 ปัญหาอันเกิดจากพื้นที่ในหน่วยความจำหลักแยกกระจายออกเป็นส่วน ๆ (Fragmentation) ทำให้การใช้งานของหน่วยความจำหลักไม่มีประสิทธิภาพ ดังพื้นที่ที่ไม่ถูกใช้งานในรูปที่ 2.2

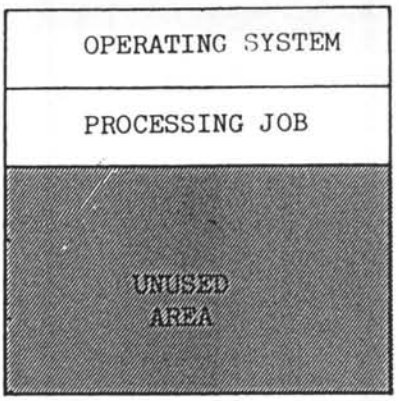
---

[1] Stuart E. Madnicr and John J. Donovan, Operating System (Kogakusha : McGraw-Hill Book Co., Ltd.), p. 107

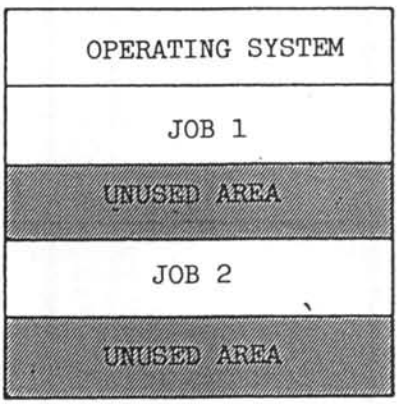
[2] Ibid., p. 115.

[3] Ibid., p. 124.

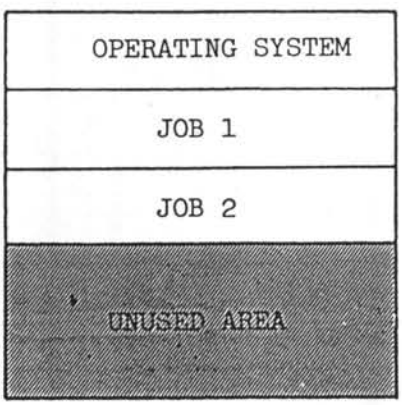
[4] Ibid., p. 129



รูปที่ 2.1 การจัดหน่วยความจำแบบซิงเกิลอโลเคชัน



รูปที่ 2.2 การจัดหน่วยความจำแบบกำหนดขนาดและตำแหน่งของพาติชันแบบคงที่



รูปที่ 2.3 การจัดหน่วยความจำแบบโยกย้ายตำแหน่งไปมาได้

2.1.2 หากทำการรวมพื้นที่แยกกระจายเข้าด้วยกันเพื่อนำมาใช้ประโยชน์ (Compaction) ลักษณะของระบบควบคุมการทำงาน (Operating System) จะยุ่งยากและใช้เวลาในการทำงานมากขึ้น และนอกจากนั้นยังใช้เนื้อที่ในหน่วยความจำหลักมากขึ้น ผลที่ตามมาก็คือเนื้อที่ภายในหน่วยความจำหลักสำหรับงานที่เข้ามาประมวลผลลดลง

2.1.3 ขนาดของงานยังคงถูกจำกัดด้วยขนาดของหน่วยความจำหลักอยู่เช่นเดิม

จากลักษณะของระบบการทำงานแบบมัลติโปรแกรมมิง ทำให้ระบบมีความต้องการใช้หน่วยความจำหลักที่มีขนาดใหญ่มากขึ้นอันเนื่องมาจากขนาดของระบบควบคุมการทำงานใหญ่ขึ้นและความต้องการใช้เนื้อที่ของงานที่เข้ามาประมวลผลในระบบดังกล่าวก็สูงขึ้นด้วย การแก้ไขปัญหาดังกล่าวอาจจะกระทำได้ง่าย ๆ โดยการใช้หน่วยความจำหลักที่มีขนาดใหญ่มากขึ้นเพื่อให้เพียงพอกับการใช้งาน แต่การแก้ไขด้วยวิธีการดังกล่าวเป็นการกระทำที่ไม่เหมาะสม เนื่องจากค่าใช้จ่ายและราคาของหน่วยความจำหลักที่มีขนาดใหญ่นั้นสูงมากจนอาจจะทำให้ผลที่ได้รับไม่คุ้มค่าใช้จ่าย[1] ต่อมาได้มีแนวคิดที่จะพยายามใช้หน่วยความจำหลักที่มีอยู่ให้มีประสิทธิภาพมากขึ้นกว่าเดิมโดยการสร้างภาพเสมือนของหน่วยความจำหลักขึ้นมา ภาพเสมือนของหน่วยความจำหลักนี้จะมีขนาดใหญ่เพียงพอกับการใช้งาน และขนาดของหน่วยความจำหลักที่แท้จริงจะยังคงมีขนาดคงเดิมและเล็กกว่าภาพเสมือนดังกล่าวมาก ภาพเสมือนของหน่วยความจำหลักนี้เรียกว่า "หน่วยความจำเสมือน" (Virtual Storage)

---

[1]Stuart E. Madnick and John J. Donovan, Operating System (Kogakusha : McGraw-Hill Book Co., Ltd.), p. 139

ข้อมูลต่าง ๆ ที่อยู่ในหน่วยความจำเสมือนจะถูกเก็บไว้ในหน่วยความจำสำรอง (Auxiliary Storage) เช่น จานแม่เหล็ก กระบอกแม่เหล็ก เป็นต้น เนื่องจากในการประมวลผลข้อมูลที่กระทำโดยหน่วยประมวลผลกลาง (Central Processing Unit) ข้อมูลที่ถูกประมวลผลจะต้องอยู่ในหน่วยความจำหลักเท่านั้น ดังนั้นเมื่อต้องการประมวลผลข้อมูลส่วนใดส่วนหนึ่งของงานก็จะทำการโยกย้ายข้อมูลนั้นจากหน่วยความจำสำรองมาสู่หน่วยความจำหลัก ด้วยการกระทำดังกล่าวจะเห็นได้ว่าในการประมวลผลงานใดงานหนึ่งข้อมูลและโปรแกรมทั้งหมดของงานนั้นไม่จำเป็นจะต้องอยู่ในหน่วยความจำหลักตลอดเวลา จึงทำให้การใช้งานหน่วยความจำของระบบมีประสิทธิภาพสูงขึ้นเป็นอันมาก

เนื่องจากตำแหน่งของข้อมูลที่ถูกอ้างถึงนั้นเป็นตำแหน่งที่มองโดยผู้ใช้เอง ตำแหน่งนี้เรียกว่า "ตำแหน่งเสมือน" (Virtual Address) ดังนั้นเมื่อข้อมูลดังกล่าวถูกย้ายมาอยู่ในหน่วยความจำหลักเพื่อทำการประมวลผล ตำแหน่งเสมือนจะต้องถูกแปลงมาเป็นตำแหน่งในหน่วยความจำหลักที่หน่วยประมวลผลสามารถใช้งานได้โดยถูกต้องเสียก่อน เนื่องจากตำแหน่งของข้อมูลที่อยู่ในหน่วยความจำเสมือนอาจจะไม่ตรงกับตำแหน่งในหน่วยความจำหลักก็ได้ ตำแหน่งของข้อมูลที่อยู่ในหน่วยความจำหลักนี้เรียกว่า "ตำแหน่งจริง" (Real Address) กรรมวิธีที่ใช้ในการแปลงตำแหน่งเสมือนมาเป็นตำแหน่งจริง คือ การแปลงตำแหน่งข้อมูลแบบพลศาสตร์ (Dynamic Address Translation) ซึ่งเป็นส่วนของเครื่องที่ทำงานร่วมกับระบบควบคุมการทำงาน

มีเทคนิคที่ใช้ในการจัดการหน่วยความจำที่สำคัญอยู่ 2 แบบคือ[1] การจัดการหน่วยความจำแบบดีมานด์เพจ (Demand-Paged Menory Management) และการจัด

---

[1]IBM, Introduction to Virtual Storage in System/370, Form GR 20-4260-1, (Bangkok : IBM Co., Ltd. (Thailand)), p. 8.

การหน่วยความจำแบบเซกเมนต์ (Segmented Memory Management) ส่วนระบบการจัดการหน่วยความจำที่ใช้กับเครื่องคอมพิวเตอร์ไอบีเอ็มระบบ 370 นั้น ได้รวมเอาเทคนิคของทั้งสองแบบดังกล่าวเข้าด้วยกันเรียกว่า ระบบการจัดการหน่วยความจำแบบเซกเมนต์ และ ดีมานด์เพจ (Segmented and Demand-Paged Memory Management)[1] ซึ่งจะกล่าวโดยละเอียดในส่วนต่อไป แต่ก่อนที่จะกล่าวถึงระบบการจัดการหน่วยความจำแบบเซกเมนต์ และ ดีมานด์เพจ จะกล่าวถึงระบบการจัดการหน่วยความจำแบบดีมานด์เพจ และแบบเซกเมนต์ เสียก่อนเพื่อเป็นแนวทางในการทำความเข้าใจระบบการจัดการหน่วยความจำแบบเซกเมนต์และดีมานด์เพจต่อไป

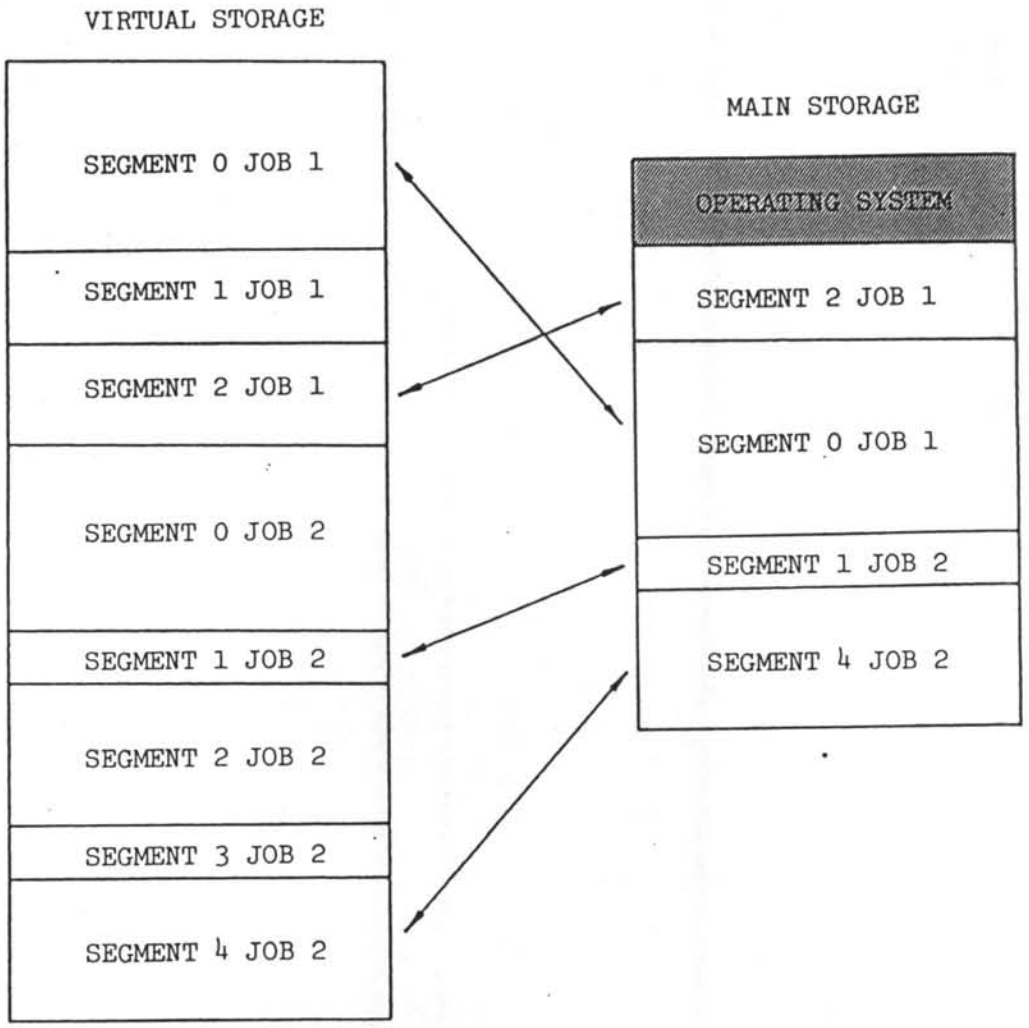
2.2 ระบบการจัดการหน่วยความจำแบบเซกเมนต์

2.2.1 การจัดแบ่งเนื้อที่ภายในหน่วยความจำ

สำหรับในระบบการจัดการหน่วยความจำแบบเซกเมนต์นี้ หน่วยความจำเล่มือนจะถูกแบ่งออกเป็นส่วน ๆ ตามขนาดและลักษณะของงานที่ส่งเข้ามาเพื่อทำการประมวลผล แต่ละส่วนของงานที่ถูกแบ่งจะเรียกว่า "เซกเมนต์" ข้อมูลที่อยู่ภายในแต่ละเซกเมนต์จะมีตำแหน่งอยู่ติดกันหมดไม่ว่าจะเป็นทางตรรกภาพ (Logical View) หรือกายภาพ (Physical View) แต่ถ้าหากมีการพิจารณาทั้งงานแล้ว แต่ละเซกเมนต์ในแต่ละโปรแกรมจะมีตำแหน่งติดกันทางตรรกภาพ แต่ถ้าหากทางกายภาพแล้ว เมื่อมีการนำเอาโปรแกรมนี้ลงในหน่วยความจำหลักเพื่อทำการประมวลผล แต่ละเซกเมนต์อาจจะอยู่ติดกันหรือไม่ติดกันก็ได้และนอกจากนั้นในการประมวลผลแต่ละงาน ทุก ๆ เซกเมนต์ของงานนั้นไม่จำเป็นจะต้องอยู่ในหน่วยความจำหลักตลอดเวลา ดังแสดงในรูปที่ 2.4

---

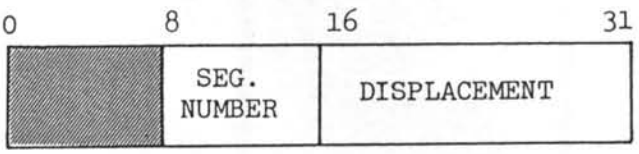
[1] IBM, Introduction to Virtual Storage in System/370, From GR 20-4260-1, (Bangkok : IBM Co., Ltd. (Thailand)), p. 15



รูปที่ 2.4 การจัดแบ่งเนื้อที่ภายในหน่วยความจำสำหรับ-
ระบบการจัดการหน่วยความจำแบบเซกเมนต์

2.2.2 การจัดลักษณะของตำแหน่งที่ใช้ในการอ้างถึงข้อมูล

เนื่องจากลักษณะของการจัดเนื้อที่ของหน่วยความจำได้กระทำเป็นเซกเมนต์ ดังนั้นการอ้างถึงตำแหน่งของข้อมูลใด ๆ จะต้องประกอบด้วยส่วนสำคัญ 2 ส่วน ได้แก่ หมายเลขของเซกเมนต์ (Segment Number) และระยะขจัด (Displacement) เมื่อเทียบกับจุดเริ่มต้นของเซกเมนต์นั้น การจัดตำแหน่งแบบนี้เป็นเทคนิคการจัดตำแหน่งแบบหนึ่งเรียกว่า "การจัดตำแหน่งข้อมูลแบบเชิงเส้น" (Linear Addressing) ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 การจัดตำแหน่งข้อมูลแบบเชิงเส้น

จากรูปที่ 2.5 เป็นการจัดตำแหน่งข้อมูลแบบเชิงเส้นที่ใช้กับเครื่องคอมพิวเตอร์ไอบีเอ็มระบบ 370 ซึ่งมีระบบการจัดการหน่วยความจำแบบเซกเมนต์ จะเห็นได้ว่าเมื่อมีการจัดตำแหน่งข้อมูลแบบนี้จะทำให้สามารถอ้างอิงข้อมูลได้ 256 เซกเมนต์ โดยที่แต่ละเซกเมนต์มีขนาด 64 กิโลไบท์

2.2.3 องค์ประกอบที่ใช้ในการแปลงตำแหน่งข้อมูล

เนื่องจากตำแหน่งข้อมูลที่ได้มาจากโปรแกรมนั้น เป็นตำแหน่งเสมือน

ดังนั้นก่อนที่จะมีการประมวลผลข้อมูลที่ถูกอ้างถึงดังกล่าว จะต้องมีการแปลงตำแหน่งเสมือนมาเป็นตำแหน่งจริงเสียก่อน ในการแปลงเป็นตำแหน่งจริงนั้นจะต้องใช้ข้อมูลต่าง ๆ มาประกอบได้แก่

- ตารางงาน (Job Table)
- รีจิสเตอร์บอกตำแหน่งเริ่มต้นของตารางเซกเมนต์ (Segment-Table Origin Register)
- ตารางเซกเมนต์ (Segment Table)

ตารางงาน

เป็นตารางที่ใช้เก็บตำแหน่งของตารางเซกเมนต์ของแต่ละงาน จำนวนของข้อมูลที่อยู่ภายในตารางงาน (Job Table Entry) จะเท่ากับจำนวนสูงสุดของงานที่สามารถทำการประมวลผลพร้อม ๆ กันได้ในระบบการประมวลผลแบบมัลติโปรแกรมมิง จากรูปที่ 2.6 เป็นตัวอย่างของตารางงานที่ใช้สำหรับระบบที่สามารถประมวลผลงานได้พร้อมกันทั้งหมด 5 งาน ตำแหน่งของตารางเซกเมนต์ของแต่ละงานนั้นเป็นตำแหน่งจริง ดังนั้นการอ้างถึงตำแหน่งดังกล่าวจึงไม่มีการแปลงตำแหน่งอีก

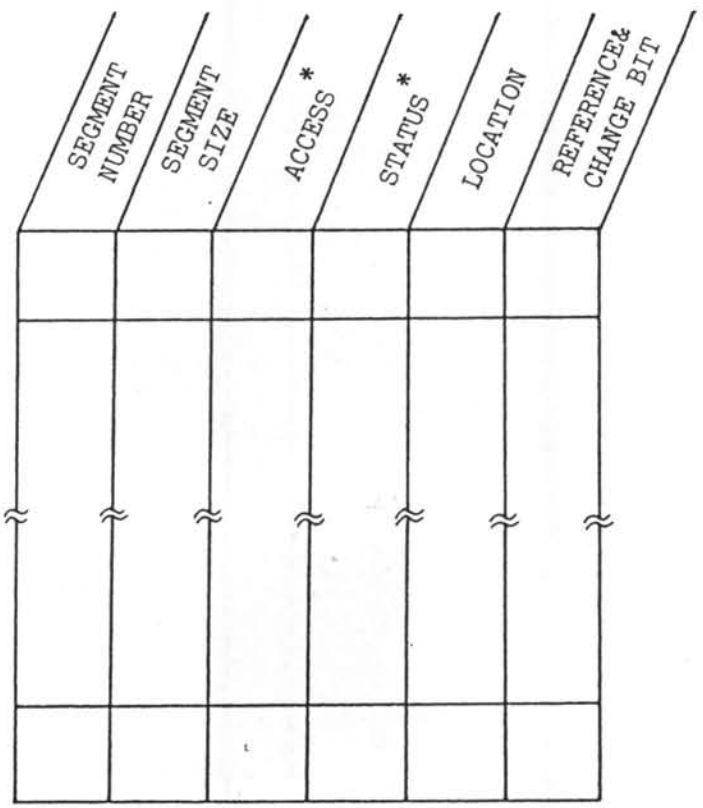
ตารางเซกเมนต์

ใช้ในการอ้างถึงตำแหน่งของเซกเมนต์ที่อยู่ภายในหน่วยความจำหลัก (Segment Address) ขนาดของเซกเมนต์ (Segment Size) ซึ่งบ่งถึงความยาวของเซกเมนต์ สถานะของเซกเมนต์ (Status) ซึ่งบ่งว่าเซกเมนต์นี้อยู่ในหน่วยความจำหลักในขณะที่ทำการอ้างถึงนี้หรือไม่ ลักษณะการเข้าถึงข้อมูล (Access) ว่าข้อมูลภายในเซกเมนต์นี้จะถูกอ่าน เขียนหรือทั้งอ่านและเขียนบิทอ้างอิง (Reference Bit) ซึ่งบ่งว่าเซกเมนต์นี้ถูกอ้างอิงถึงบ้างแล้วหรือยังตั้งแต่ถูกนำมาลงในหน่วยความจำหลัก และบิทแสดงการเปลี่ยนแปลงข้อมูล (Change Bit) ภายในแต่ละเซกเมนต์นั้น จำนวนข้อ

JOB NO.	SIZE	LOCATION OF ST.	STATUS*
1			
2			
3			
4			
5			

*STATUS
- ALLOCATED
- EMPTY ENTRY

รูปที่ 2.6 ตารางงาน



*ACCESS - E EXECUTED ALLOWED.
- R READ ALLOWED.
- W WRITE ALLOWED.

*STATUS - Y IN MAIN STORAGE.
- N NOT IN MAIN STORAGE.

รูปที่ 2.7 ตารางเซกเมนต์

มูลภายในตารางเซกเมนต์ (Segment Table Entry) ขึ้นอยู่กับจำนวนของเซกเมนต์ในแต่ละงาน ในแต่ละงานจะมีตารางเซกเมนต์ของตนเอง และตารางเซกเมนต์นี้จะอยู่ภายในหน่วยความจำหลักตลอดเวลา การปรับปรุงแก้ไขและใช้งานตารางเซกเมนต์จะกระทำโดยระบบควบคุมการทำงาน ลักษณะของตารางเซกเมนต์จะแสดงดังรูปที่ 2.7

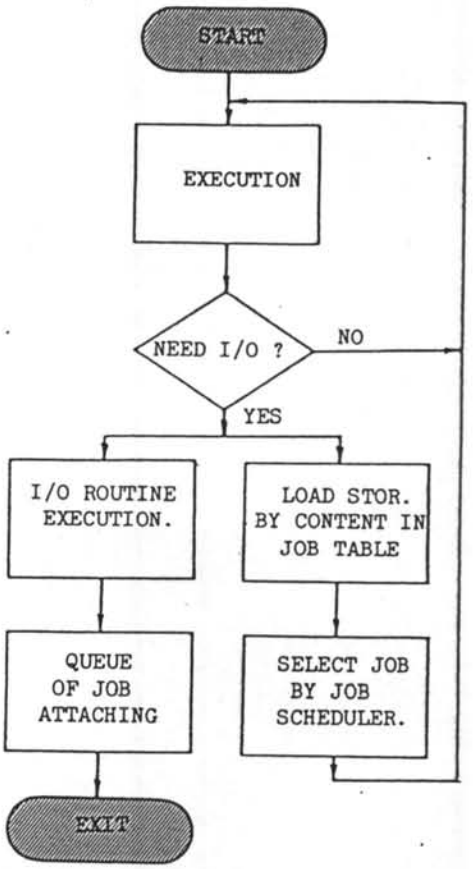
<u>รีจิสเตอร์บอกตำแหน่งเริ่มต้นของตารางเซกเมนต์</u>

ระบบควบคุมการทำงานจะใช้รีจิสเตอร์ควบคุมตัวใดตัวหนึ่งมาเป็นรีจิสเตอร์บอกตำแหน่งเริ่มต้นของตารางเซกเมนต์ ข้อมูลภายในรีจิสเตอร์นี้จะเป็นตำแหน่งเริ่มต้นของตารางเซกเมนต์ของงานที่จะใช้ในการประมวลผล รีจิสเตอร์ตัวนี้จะถูกใช้ในการแปลงตำแหน่งเสมือนมาเป็นตำแหน่งจริงระหว่างที่มีการประมวลผลข้อมูลนั่นเอง

2.2.4 การแปลงตำแหน่งข้อมูล

เนื่องจากการประมวลผลของระบบเป็นแบบมัลติโปรแกรมมิง ดังนั้นในการแปลงตำแหน่งเสมือนมาเป็นตำแหน่งจริงเพื่อทำการประมวลผลนั้น ข้อมูลที่อยู่ภายในรีจิสเตอร์บอกตำแหน่งเริ่มต้นของตารางเซกเมนต์จะต้องเป็นตำแหน่งของตารางเซกเมนต์ของงานที่จะประมวลผลเสมอ เมื่องานที่กำลังทำการประมวลผลอยู่มีความต้องการใช้อุปกรณ์การนำข้อมูลเข้าหรือออก ระบบการจัดงาน (Job Scheduler) ซึ่งเป็นส่วนหนึ่งของระบบควบคุมการทำงานจะทำการเลือกงานต่อไปมาทำการประมวลผลต่อเพราะในช่วงนี้หน่วยประมวลผลกลางจะว่าง เมื่อทำการเลือกงานนั้นได้แล้ว ก็จะทำการนำค่าของตำแหน่งของตารางเซกเมนต์ของงานนั้น (นำมาจากตารางงาน) ไปเก็บที่รีจิสเตอร์บอกตำแหน่งเริ่มต้นของตารางเซกเมนต์ก่อนจึงจะทำการประมวลผลต่อไป ดังแสดงในผังงานที่ 2.1

การแปลงตำแหน่งข้อมูลที่โปรแกรมอ้างถึงในขณะที่ทำการประมวลผลนั้น จะได้นำมาอธิบายในลักษณะของผังงานที่ 2.2 และรูปที่ 2.8 ประกอบคำอธิบายดังต่อไปนี้



ผังงานที่ 2.1 ลักษณะการเปลี่ยนงานที่จะทำการประมวลผล

1. ทำการหาข้อมูลภายในตารางเซกเมนต์ โดยการใช้ค่าภายในรีจิสเตอร์บอกตำแหน่งเริ่มต้นของเซกเมนต์กับค่าของหมายเลขของเซกเมนต์ที่ได้จากตำแหน่งเสมือน

2. ทำการตรวจสอบสถานะของเซกเมนต์นั้นว่าอยู่ภายในหน่วยความจำหลักในขณะที่มีการอ้างถึงหรือไม่ หากไม่อยู่จะเกิดการขัดจังหวะ (Interrupt) ในการประมวลผลข้อมูลที่ตำแหน่งนั้นเนื่องจากข้อผิดพลาดของสถานะเซกเมนต์ (Segment Exception) การทำงานจะถูกส่งกลับไปยังระบบควบคุมการทำงานเพื่อทำการแก้ไขต่อไป (ในกรณีเช่นนี้จะเรียกว่า "เซกเมนต์ฟอลท์" ซึ่งจะกล่าวถึงหลักการทำงานของระบบควบคุมการทำงานเมื่อเกิดกรณีเช่นนี้ต่อไป)

3. ทำการตรวจสอบว่าระยะขจัดที่อ้างถึงนั้นเกินขนาดของเซกเมนต์นั้นหรือไม่ โดยทำการตรวจสอบกับค่าความยาวของเซกเมนต์ที่อยู่ภายในข้อมูลที่ได้จากตารางเซกเมนต์ หากมีการอ้างเกินจะเกิดการขัดจังหวะเนื่องจากการอ้างถึงตำแหน่งข้อมูลนอกเซกเมนต์นั้น และการทำงานจะถูกส่งกลับมายังระบบควบคุมการทำงานต่อไป

4. ทำการตรวจสอบลักษณะของการเข้าถึงข้อมูลภายในเซกเมนต์ว่าตรงตามที่กำหนดภายในข้อมูลที่ได้จากตารางเซกเมนต์หรือไม่ หากไม่ตรงจะเกิดการขัดจังหวะเนื่องจากการใช้ข้อมูลผิดลักษณะและการทำงานจะถูกส่งกลับมายังระบบควบคุมการทำงานต่อไป

5. กำหนดค่าบิทอ้างอิงภายในข้อมูลที่ได้จากตารางเซกเมนต์นั้นให้เป็น 1

6. หากข้อมูลภายในเซกเมนต์นั้นมีการเปลี่ยนแปลงในระหว่างที่อยู่ภายในหน่วยความจำหลัก (มีการเขียนข้อมูลลงในเซกเมนต์) จะทำการกำหนดค่าภายในบิทแสดงการเปลี่ยนแปลงข้อมูลให้เป็น 1

7. ทำการหาตำแหน่งจริงโดยการนำค่าของตำแหน่งของเซกเมนต์ที่ได้จากข้อมูลภายในตารางเซกเมนต์บวกกับระยะขจัด

8. ทำการประมวลผลข้อมูลนั้นโดยใช้ตำแหน่งจริงที่ได้จากการแปลงตำแหน่ง

ในกรณีที่เกิดเซกเมนต์ฟอลท์ขึ้นระบบควบคุมการทำงานจะทำการจัดการดังต่อไปนี้

1. ทำการตรวจสอบว่าเนื้อที่ที่ไม่ถูกใช้งานโดยเซกเมนต์ใด ๆ ภายในหน่วยความจำหลักรวมกันแล้วมากกว่าหรือเท่ากับเนื้อที่ของเซกเมนต์ที่เกิดเซกเมนต์ฟอลท์หรือไม่ หากเนื้อที่มากกว่าหรือเท่ากับแล้วยังเป็นเนื้อที่ที่มีตำแหน่งติดต่อกันทั้งหมดอีกด้วยก็จะทำการโยกย้ายเซกเมนต์ที่ก่อให้เกิดเซกเมนต์ฟอลท์ซึ่งเก็บไว้ภายในหน่วยความจำสำรองมาลงในเนื้อที่ภายในหน่วยความจำหลักนั้น แต่ถ้าหากเนื้อที่ดังกล่าวมีตำแหน่งแยกกระจายออกเป็นหลาย ๆ ส่วน ดังนั้นก่อนที่จะมีการโยกย้ายเซกเมนต์ต้องทำการโยกย้ายตำแหน่งของเซกเมนต์ต่าง ๆ ที่อยู่ภายในหน่วยความจำหลักเสียใหม่จนได้เนื้อที่ที่มีตำแหน่งติดกันและมีขนาดพอเพียงเสียก่อน (ข้อมูลภายในตารางเซกเมนต์จะต้องเปลี่ยนแปลงไปด้วย) ดังผังงานที่ 2.3

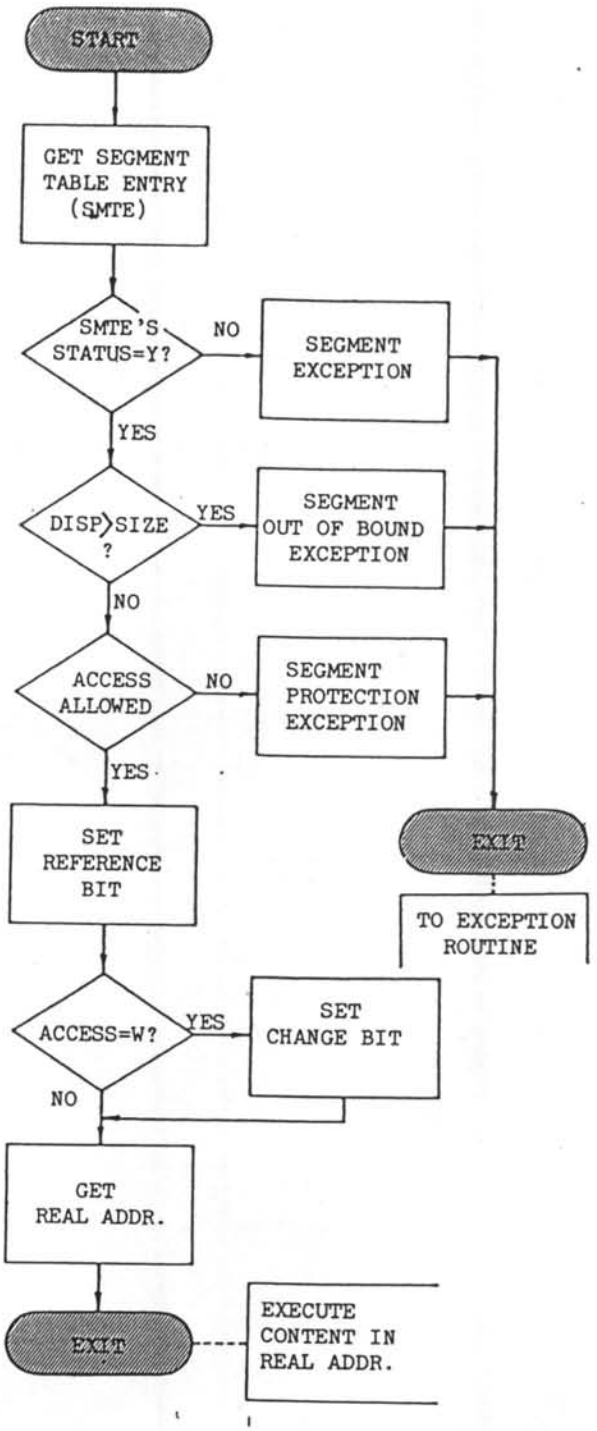
2. หากเนื้อที่นั้นรวมกันแล้วมีขนาดเล็กกว่าขนาดเซกเมนต์ที่ก่อให้เกิดเซกเมนต์ฟอลท์ จะต้องมีการเลือกเอาบางเซกเมนต์ที่อยู่ภายในหน่วยความจำหลักออกไปเสียบ้างโดยการพิจารณาถึงขนาดค่าภายในบิทอ้างอิงและบิทแสดงการเปลี่ยนแปลงข้อมูลประกอบในการเลือก แล้วจึงจะทำการโยกย้ายตำแหน่งของเซกเมนต์เพื่อรวมเนื้อที่หากมีความจำเป็น ดังผังงานที่ 2.3

2.2.5 หน้าที่ของระบบควบคุมการทำงาน

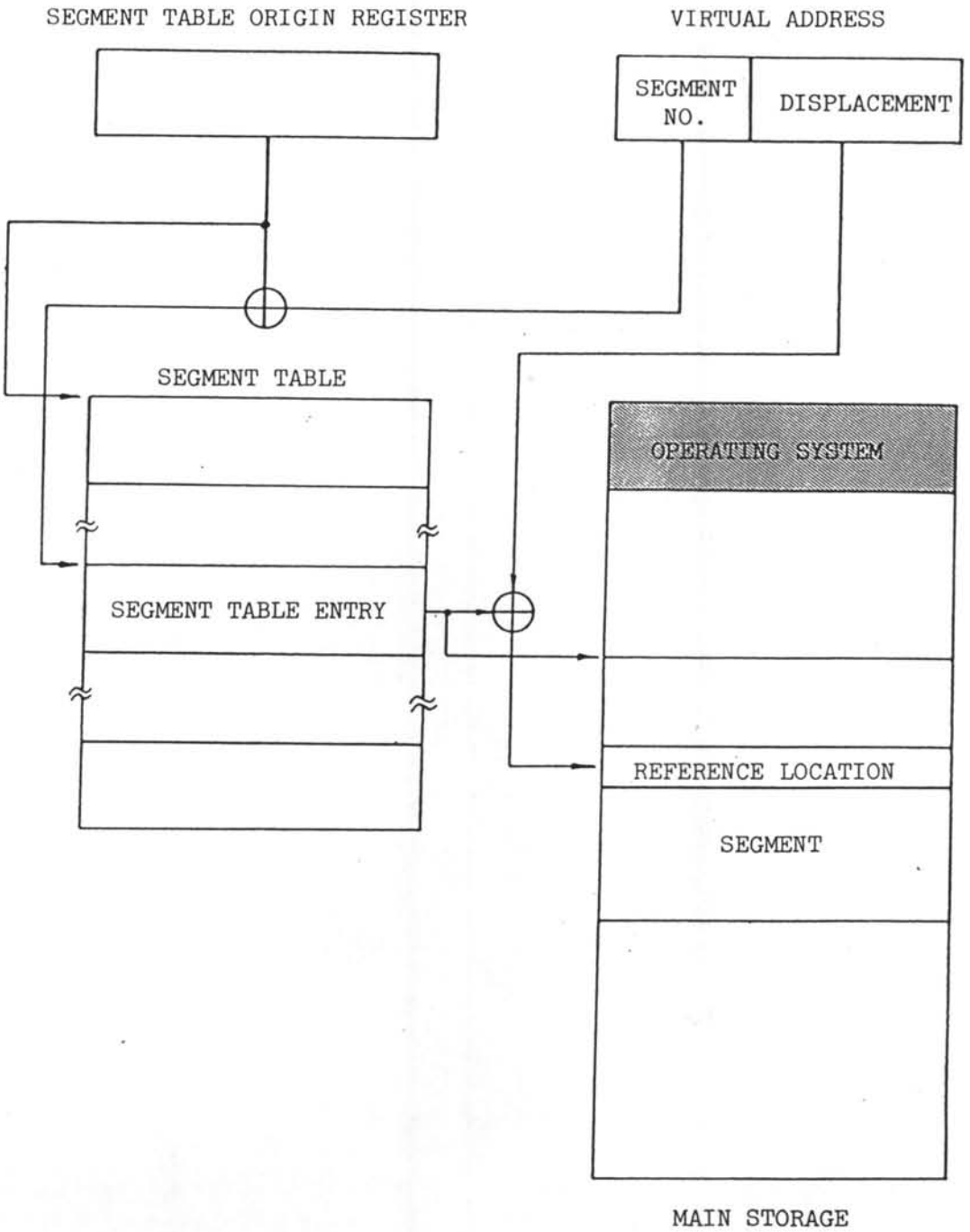
สรุปหน้าที่ของระบบควบคุมการทำงานสำหรับระบบการจัดการหน่วยความจำแบบเซกเมนต์ได้ดังนี้

1. ทำการตรวจสอบและปรับปรุงแก้ไขข้อมูลภายในตารางเซกเมนต์รีจิสเตอร์บอกตำแหน่งของตารางเซกเมนต์

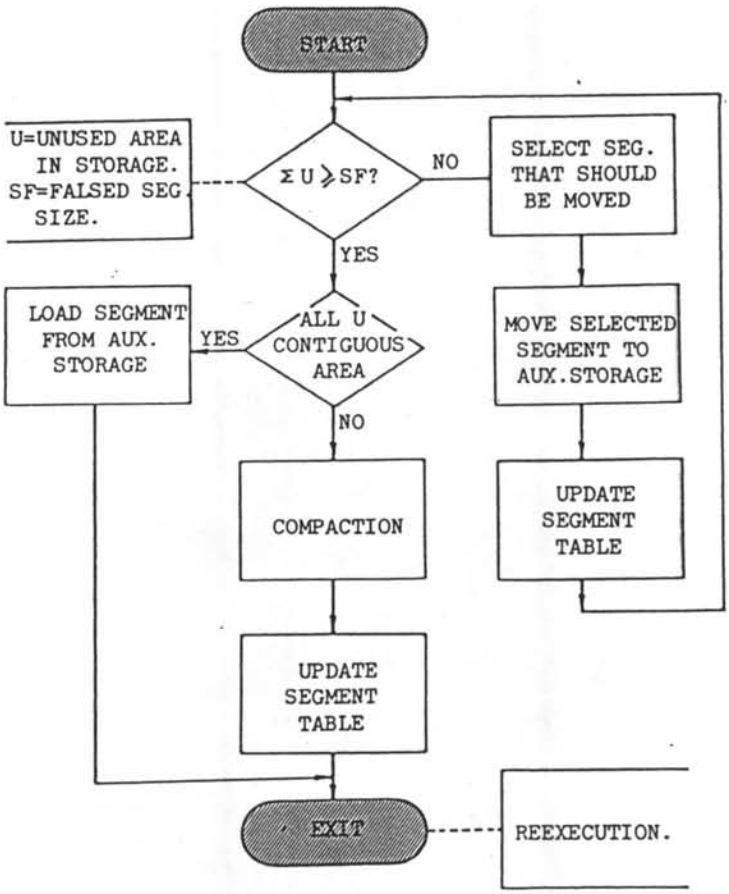
007325



ฝังงานที่ 2.2 การแปลงตำแหน่งข้อมูลสำหรับระบบการจัดการ-หน่วยความจำแบบเซกเมนต์



รูปที่ 2.8 แผนผังความสัมพันธ์ในการแปลงตำแหน่งข้อมูล



ผังงานที่ 2.3 ผังการทำงานกรณีเกิดเซกเมนต์ฟอลท์

2. ทำการเลือกงานที่จะทำการประมวลผลในระบบการประมวลผลแบบมัลติโปรแกรมมิง

3. จัดการหน่วยความจำให้เป็นแบบเซกเมนต์ โดยอาศัยหลักการของเซกเมนต์ดีมานด์

4. ทำการโยกย้ายตำแหน่งของเซกเมนต์เพื่อจะได้พื้นที่ภายในหน่วยความจำหลักที่มีตำแหน่งติดกัน

5. ทำการโยกย้ายข้อมูลระหว่างหน่วยความจำหลักและหน่วยความจำสำรองในกรณีที่เกิดเซกเมนต์ฟอลท์

6. เมื่องานที่ทำการประมวลผลเสร็จสิ้นลง จะกำหนดพื้นที่ภายในหน่วยความจำเสมือนและหน่วยความจำหลักที่งานนั้นครอบครองอยู่ให้ว่าง เพื่อให้สามารถทำงานอื่นต่อไป

2.2.6 ข้อดีข้อเสียของระบบการจัดการหน่วยความจำแบบเซกเมนต์[1]

ข้อดี

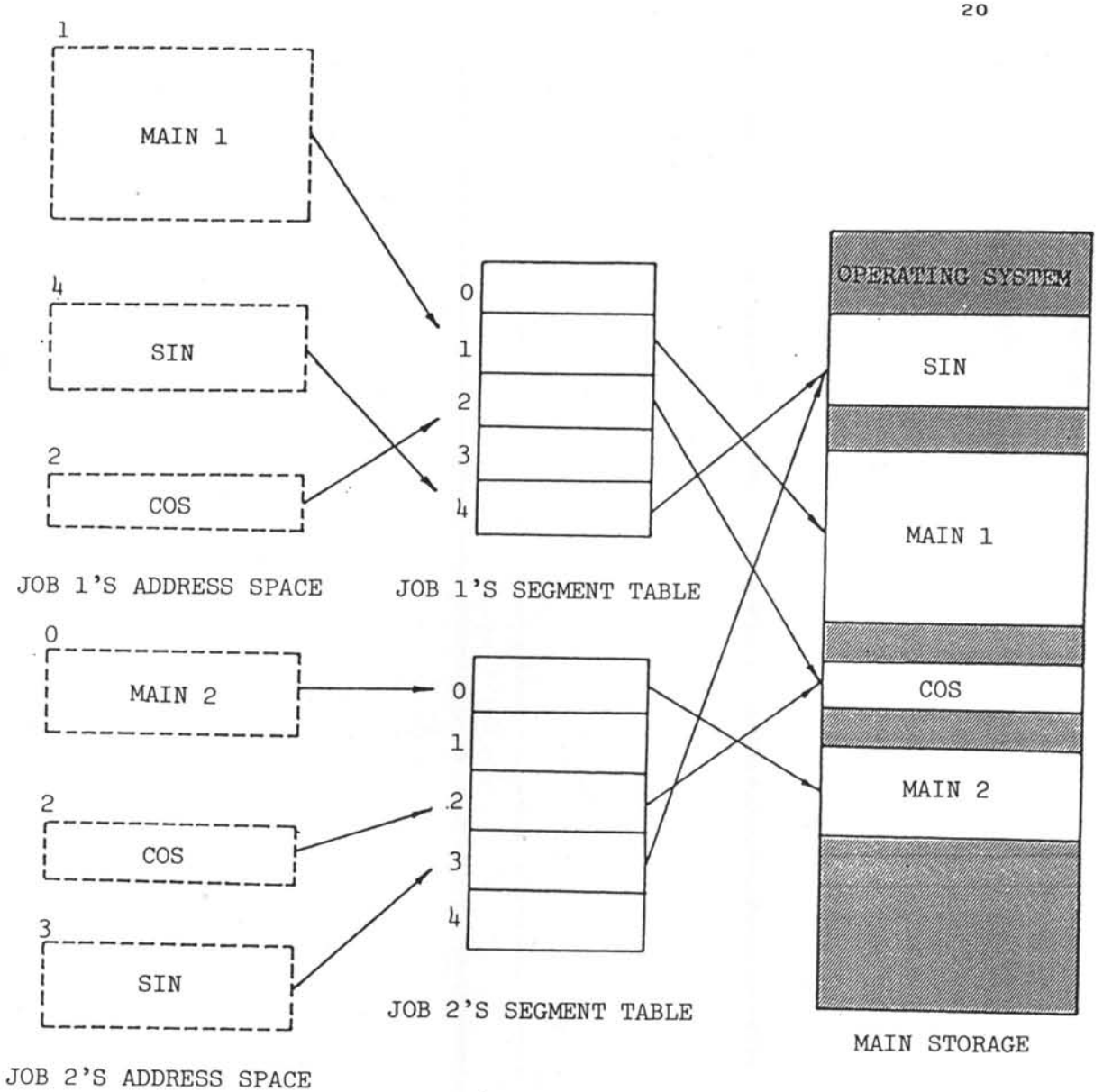
1. ลดการแยกกระจายของเนื้อที่ภายในหน่วยความจำหลัก ช่วยให้การใช้งานหน่วยความจำหลักมีประสิทธิภาพ

2. งานมากกว่าสองงานสามารถใช้บางเซกเมนต์ร่วมกันได้ จึงช่วยลดความซ้ำซ้อนของข้อมูลภายในหน่วยความจำหลักลงได้มาก ผลที่ได้ก็คือประหยัดเนื้อที่ในหน่วยความจำหลักดังแสดงในรูปที่ 2.9

3. สามารถควบคุมการเข้าถึงข้อมูลของแต่ละเซกเมนต์ ช่วยป้องกันความผิดพลาดในการใช้งานได้มากยิ่งขึ้น

---

[1]Stuart E. Madnick and John J. Donovan, Operating System (Kogakusha : McGraw-Hill Book Co., Ltd.), pp. 180-181.



รูปที่ 2.9 ลักษณะการใช้เซกเมนต์ร่วมกัน

ข้อเสีย

1. ต้องใช้เวลาในการโยกย้ายตำแหน่งของเซกเมนต์ เพราะต้องมีการโยกย้ายข้อมูลและปรับปรุงข้อมูลภายในตารางเซกเมนต์

2. มีความยุ่งยากในการจัดเก็บโปรแกรมซึ่งมีขนาดของเซกเมนต์ไม่เท่ากันลงในหน่วยความจำสำรอง

3. ขนาดสูงสุดของเซกเมนต์ถูกจำกัดด้วยขนาดของหน่วยความจำหลัก

4. มีความยุ่งยากในการป้องกันการเกิดเซกเมนต์แทรชชิ่ง (Segment Thrashing) ซึ่งหมายถึงการโยกย้ายเซกเมนต์ใด ๆ ระหว่างหน่วยความจำหลักและหน่วยความจำสำรองบ่อย ๆ ครั้ง

2.3 ระบบการจัดการหน่วยความจำแบบดีมานด์เพจ

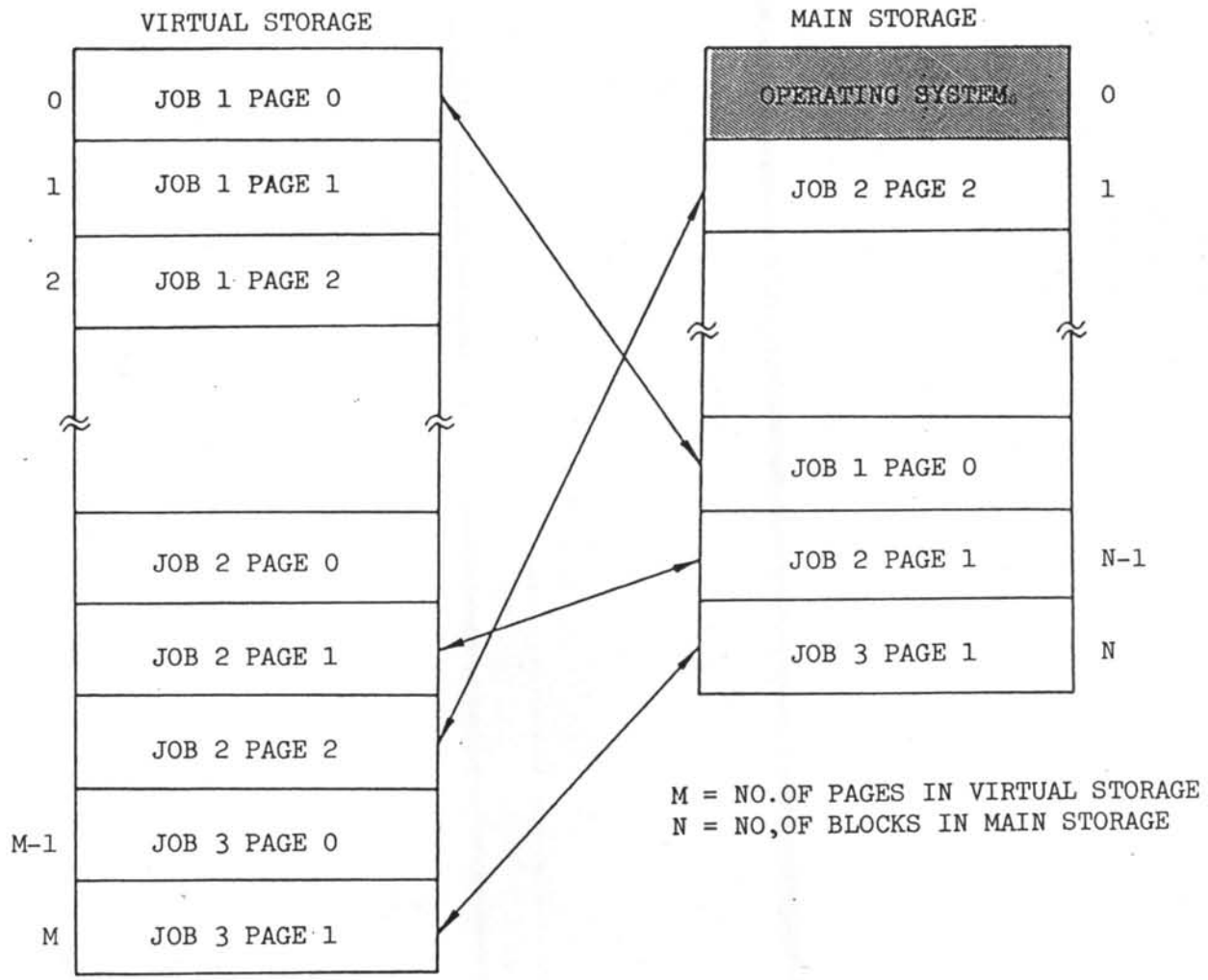
2.3.1 การจัดแบ่งเนื้อที่ภายในหน่วยความจำ

สำหรับในระบบการจัดการหน่วยความจำแบบดีมานด์เพจนี้ บริเวณตำแหน่งที่โปรแกรมอ้างถึง (Address Space) ที่อยู่ภายในหน่วยความจำเสมือน จะถูกแบ่งออกเป็นส่วน ๆ โดยที่แต่ละส่วนจะมีขนาดความยาวเท่ากันและตำแหน่งข้อมูลภายในส่วนนี้จะอยู่ติดกันทั้งทางตรรกภาพและกายภาพ ส่วนเหล่านี้เรียกว่า "เพจ" (Page)[1] ส่วนเนื้อที่ภายในหน่วยความจำหลักนั้นก็จะถูกแบ่ง ออกเป็นส่วน ๆ ด้วยเช่นกัน และตำแหน่งข้อมูลภายในส่วนเหล่านี้จะอยู่ติดกันทั้งทางตรรกภาพและกายภาพเช่นเดียวกัน ส่วนเหล่านี้เรียกว่า "บลอค" (Block)[2] หรือ "เพจเฟรม" (Page Frame) และขนาด

---

[1]Stuart E. Madnick and John J. Donovan, Operating System (Kogakusha : McGraw-Hill Book Co., Ltd.) p. 130

[2]Ibid.



รูปที่ 2.10 การจัดแบ่งเนื้อที่ภายในหน่วยความจำของระบบการจัดการ-หน่วยความจำแบบดีมานด์เพจ

ของเพจกับบลอคและเพจเฟรมจะเท่ากัน ดังนั้นในการประมวลผลข้อมูลเพจจะถูกนำมาลงในบลอคภายในหน่วยความจำหลักพอดี และบลอคที่บรรจุเพจทั้งหมดของโปรแกรมหนึ่งอาจจะไม่อยู่ติดกันทางกายภาพทั้งหมดได้ดังแสดงในรูปที่ 2.10 ขนาดของเพจที่ใช้ส่วนมากจะเป็นกำลังของ 2 เช่น 2 กิโลไบท์ หรือ 4 กิโลไบท์ เป็นต้น ซึ่งการกำหนดขนาดของเพจนี้นับว่าสำคัญมาก เพราะหากขนาดของเพจเล็กเกินไปก็จะทำให้เกิดแทรช์ยิ่ง แต่ถ้าหากใหญ่เกินไปก็จะทำให้เกิดการแยกกระจายของพื้นที่ที่ไม่ได้นำมาใช้งานทำให้ประสิทธิภาพในการใช้หน่วยความจำหลักลดลง

2.3.2 การจัดลักษณะของตำแหน่งที่ใช้ในการอ้างถึงข้อมูล

ลักษณะของการจัดตำแหน่งข้อมูลที่ใช้กับระบบการจัดการหน่วยความจำแบบดีมานด์เพจนี้จะคล้ายกันกับของระบบการจัดการหน่วยความจำแบบเซกเมนต์ คือจะใช้เทคนิคการจัดตำแหน่งข้อมูลแบบเชิงเส้นเช่นเดียวกัน ดังแสดงในรูปที่ 2.11

PAGE NUMBER	DISPLACEMENT

รูปที่ 2.11 ตำแหน่งข้อมูลที่ใช้ในระบบการจัดการหน่วยความจำแบบดีมานด์-เพจ

จากรูปที่ 2.11 จะเห็นได้ว่าขนาดของเซตพิกัดข้อมูลแต่ละเซตนั้นจะขึ้นอยู่กับขนาดของเพจเป็นสำคัญ อย่างเช่นของเครื่องไอบีเอ็ม ระบบ 370 ซึ่งจะได้อธิบายโดยละเอียดในส่วนต่อไป

2.3.3 องค์ประกอบที่ใช้ในการแปลงตำแหน่งข้อมูล

สำหรับระบบการจัดการหน่วยความจำแบบดีมานด์เพจนั้น องค์ประกอบที่ใช้ในการแปลงตำแหน่งข้อมูลได้แก่

- ตารางงาน
- รีจิสเตอร์แสดงตำแหน่งตารางเพจ (Page Map Table Address Register)
- ตารางเพจ (Page Table)
- ตารางแฟ้มตำแหน่งข้อมูล (File Map Table)
- ตารางบลอคหน่วยความจำ (Memory Block Table)

ตารางงาน

ลักษณะของตารางงานที่ใช้กับระบบการจัดการหน่วยความจำแบบดีมานด์เพจนี้จะเหมือนกันกับตารางงานของระบบการจัดการหน่วยความจำแบบเซกเมนต์

รีจิสเตอร์แสดงตำแหน่งตารางเพจ

ลักษณะการใช้งานจะเหมือนกับรีจิสเตอร์บอกตำแหน่งเริ่มต้นของตารางเซกเมนต์ที่ใช้กับระบบการจัดการหน่วยความจำแบบเซกเมนต์

ตารางเพจ

ตารางเพจที่ใช้กับระบบการจัดการหน่วยความจำแบบดีมานด์เพจจะมีหน้าที่ในการกำหนดว่าเพจนั้น ๆ อยู่ภายในบลอคภายในหน่วยความจำหลักบลอคใด และอยู่ภายในหน่วยความจำหลักยังอ้างถึงหรือไม่ซึ่งเป็นสถานะของเพจนั่นเอง ลักษณะของตารางเพจนั้นอาจจะใช้ชุดของรีจิสเตอร์หลาย ๆ ตัวมาประกอบเป็นตารางเพจ หรืออาจจะใช้เนื้อที่ส่วนหนึ่งภายในหน่วยความจำหลักได้เช่นกัน แม้ว่าแบบแรกจะทำให้การแปลงข้อมูลกระทำได้ด้วยความรวดเร็ว แต่ค่าใช้จ่ายของระบบที่ใช้ตารางเพจแบบนี้จะสูงมาก ดังนั้นจึงนิยมใช้ตารางเพจในลักษณะหลังดังแสดงในรูปที่ 2.12

PAGE NUMBER	BLOCK NUMBER	STATUS

รูปที่ 2.12 ลักษณะเขตข้อมูลภายในตารางเพจ

ตารางแฟ้มตำแหน่งข้อมูล

เนื่องจากข้อมูลที่อยู่ภายในหน่วยความจำเล่มือนนั้นถูกเก็บไว้ภายในหน่วยความจำสำรอง ดังนั้นในการโยกย้ายข้อมูลระหว่างหน่วยความจำหลักและหน่วยความจำสำรองจะต้องทราบถึงตำแหน่งข้อมูลภายในหน่วยความจำสำรองซึ่งเพจนั้นบรรจุอยู่ ตำแหน่งข้อมูลของเพจแต่ละเพจจะนำมาประกอบได้สองลักษณะคือ นำไปต่อกับทุกข้อมูลภายในตารางเพจดังรูปที่ 2.13 หรืออาจจะแยกออกมาเป็นตารางภายนอกดังรูปที่ 2.14

BLOCK NUMBER	STATUS	FILE ADDRESS

รูปที่ 2.13 แฟ้มตำแหน่งข้อมูลร่วมกับข้อมูลภายในตารางเพจ

PAGE NUMBER	FILE ADDRESS

รูปที่ 2.14 ลักษณะข้อมูลภายในตารางแฟ้มข้อมูล

ตารางบลอคหน่วยความจำ

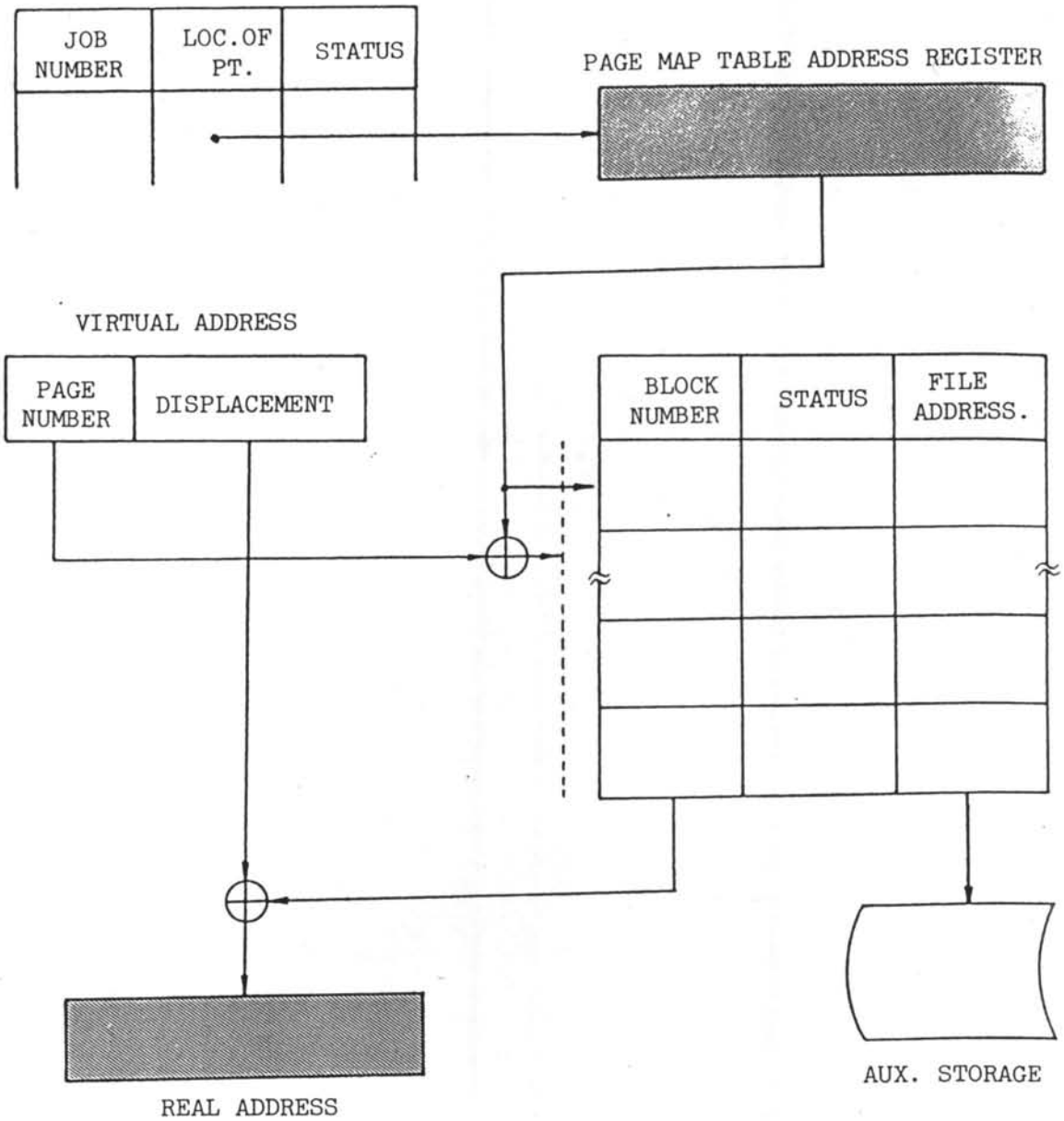
ข้อมูลภายในตารางบลอคหน่วยความจำจะบอกสถานะของแต่ละบลอคข้อมูลของหน่วยความจำหลัก ข้อมูลนี้จะนำมาใช้ในระบบควบคุมการทำงานที่ใช้ในการจัดการหน่วยความจำ เช่น การเลือกบลอคที่จะนำข้อมูลกลับไปยังหน่วยความจำสำรองที่เรียกว่าการย้ายเพจออก

2.3.4 การแปลงตำแหน่งข้อมูล

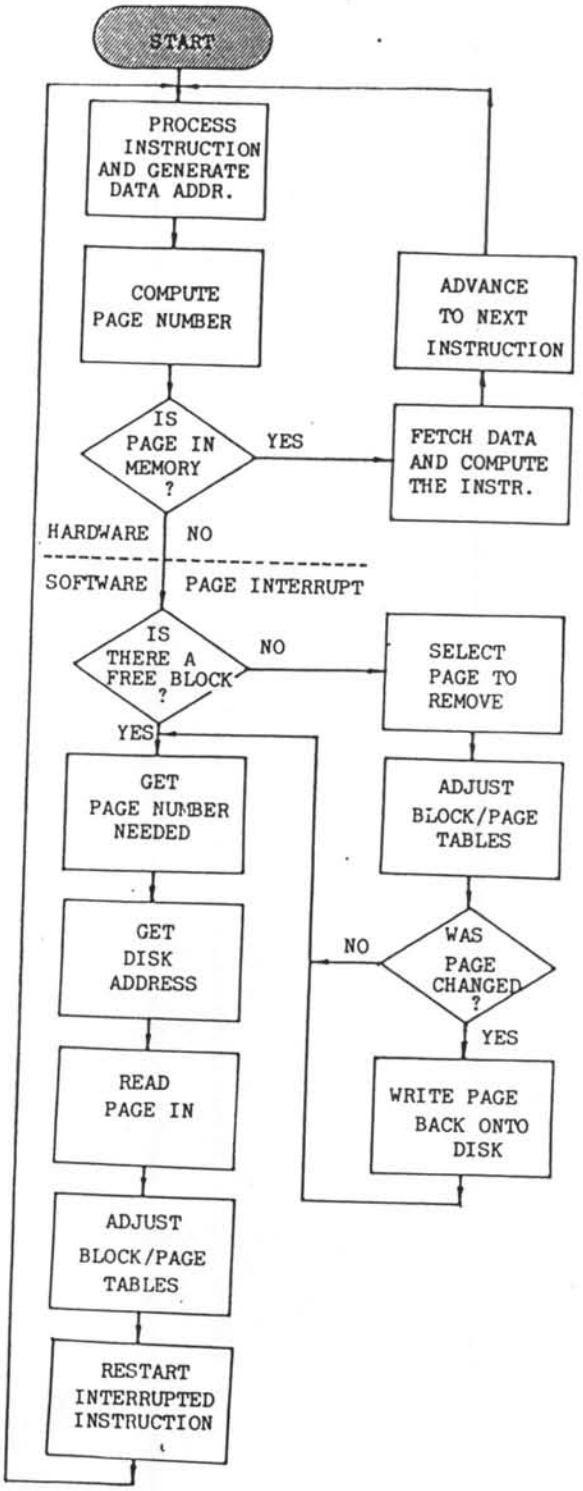
การแปลงตำแหน่งข้อมูลของระบบการจัดการหน่วยความจำแบบดีมานด์เพจนี้จะใช้หลักการคล้าย ๆ กันกับระบบการจัดการหน่วยความจำแบบเซกเมนต์ โดยการใช้ข้อมูลภายในรีจิสเตอร์บอกตำแหน่งตารางเพจ เมื่อได้ตารางเพจก็จะใช้ส่วนของหมายเลยเพจภายในตำแหน่งเสมือนเป็นข้อมูลในการหาข้อมูลภายในตารางเพจ จะสามารถสรุปได้ดังรูปที่ 2.15 และผังงานที่ 2.4

ในการทำงานของการแปลงตำแหน่ง ฮาร์ดแวร์ที่ทำงานด้านนี้ จะนำเอาหมายเลยเพจไปตรวจสอบกับตารางเพจเพื่อตรวจดูว่าเพจนั้นอยู่ในหน่วยความจำหลักแล้วหรือไม่ โดยตรวจสอบจากสถานะของเพจนั้น (สถานะเท่ากับ Y แสดงว่าเพจนั้นอยู่ในหน่วยความจำหลักและสถานะเท่ากับ N แสดงว่าเพจนั้นไม่ได้อยู่ในหน่วยความจำหลักขณะที่ถูกอ้างถึง) หากเพจนั้นอยู่ในหน่วยความจำหลักแล้ว ก็จะดึงเอาหมายเลยบลอคซึ่งอยู่ภายในข้อมูลในตารางเพจที่สัมพันธ์กับเพจนั้นออกมา ซึ่งหมายเลยบลอคจะชี้ถึงตำแหน่งของบลอคที่เพจนั้นบรรจุอยู่ ส่วนหมายเลยไบท์จะเป็นตัวบ่งถึงระยะขจัดของข้อมูลเมื่อเทียบกับบลอคนั้น เมื่อนำตำแหน่งของบลอคและหมายเลยไบท์มาบวกกันก็จะได้ตำแหน่งจริงตามต้องการ

แต่ถ้าหากทำการตรวจสอบข้อมูลภายในตารางเพจแล้วพบว่าเพจนั้นไม่อยู่ในหน่วยความจำหลัก (สถานะของเพจเป็น N) ก็จะเกิดการขัดจังหวะขึ้นเรียกว่า "เพจฟอลท์" สัญญาณการขัดจังหวะนี้จะถูกส่งไปยังระบบควบคุมการทำงานให้ทำการนำ



รูปที่ 2.15 การแปลงตำแหน่งข้อมูล



ฝังงานที่ 2.4 ความสัมพันธ์ระหว่างฮาร์ดแวร์และซอฟท์แวร์ที่ใช้ในการแปลง-
ตำแหน่งข้อมูล

เพจดังกล่าวมาลงในหน่วยความจำหลัก

ขั้นแรกระบบควบคุมการทำงานจะทำการตรวจสอบตารางบลอคว่ามีบลอคใดที่ว่างอยู่บ้าง หากมีบลอคว่างก็จะทำการกำหนดหมายเลขบลอคนั้นแก่ข้อมูลในตารางเพจของเพจที่ทำให้เกิดเพจฟอลท์และเปลี่ยนสถานะของเพจนั้นให้เป็น Y ส่วนข้อมูลในตารางบลอคที่พบว่าว่างก็จะถูกเปลี่ยนให้เป็นสถานะที่แสดงถึงว่ากำลังถูกใช้งานอยู่ และกลับไปประมวลผลข้อมูลที่ทำให้เกิดเพจฟอลท์ใหม่อีกครั้งหนึ่ง

แต่ถ้าหากหลังจากทำการตรวจสอบตารางบลอคแล้วไม่มีบลอคใดว่างเลย ก็ต้องมีการเลือกเพจใดเพจหนึ่งออกจากบลอคที่เพจนั้นครอบครองอยู่เพื่อให้บลอคนั้นว่าง กรรมวิธีนี้เรียกว่า "การโยกย้ายเพจ" (Page Removal) ส่วนวิธีในการเลือกเพจ (Page Selection) นั้นมีอยู่หลายวิธี ซึ่งจะกล่าวถึงในบทต่อไป เมื่อได้เพจที่จะถูกเลือกออกโดยใช้วิธีการเลือกเพจแบบใดแบบหนึ่งแล้วจะทำการเปลี่ยนแปลงแก้ไขข้อมูลในตารางเพจและตารางบลอคที่สอดคล้องกัน หลังจากนั้นจะทำการตรวจสอบดูว่าข้อมูลในเพจที่อยู่ในบลอคนั้นมีการเปลี่ยนแปลงหรือไม่ (โดยในแต่ละบลอคจะมีคีย์ (Key) ที่บ่งให้ทราบว่าข้อมูลในบลอคนั้นมีการถูกอ้างอิงถึงหรือมีการเปลี่ยนแปลงหรือไม่) หากมีการเปลี่ยนแปลงก็จะทำการโยกย้ายข้อมูลในบลอคนั้นกลับมาลงยังหน่วยความจำเสมือนก่อนซึ่งเรียกว่า "การย้ายเพจออก" (Page out) แล้ว จึงทำการย้ายข้อมูลในเพจที่เกิดเพจฟอลท์มาลงในบลอคนั้นเรียกว่า "การย้ายเพจเข้า" (Page in) แล้วจึงทำการประมวลผลข้อมูลนั้นใหม่

2.3.5 หน้าที่ของระบบควบคุมการทำงาน

สรุปหน้าที่ของระบบควบคุมการทำงานสำหรับระบบการจัดการหน่วยความจำแบบดีมานด์เพจได้ดังนี้

1. ทำการตรวจสอบและปรับปรุงแก้ไขข้อมูลภายในตารางงาน ตารางบลอคหน่วยความจำ ตารางเพจ ตารางแฟ้มข้อมูลและรีจิสเตอร์บอกตำแหน่งตารางเพจ

2. ทำการโยกย้ายเพจในกรณีที่เกิดเพจฟอลท์

3. ทำการเลือกเพจที่จะต้องถูกนำออกเพื่อนำเพจที่ต้องการประมวลผลเข้ามาแทนที่ในกรณีที่เกิดเพจฟอลท์

4. ทำการตรวจสอบอัตราการโยกย้ายเพจเพื่อป้องกันการเกิดแทรชชิง

2.3.6 ข้อดีข้อเสียของระบบการจัดการหน่วยความจำแบบดีมานด์เพจ[1]

ข้อดี

1. ลดยืดจำกัดของขนาดของงานลงโดยที่ขนาดของงานสามารถใหญ่กว่าหน่วยความจำหลักแต่เล็กกว่าหน่วยความจำเสมือน

2. การใช้งานของหน่วยความจำหลักมีประสิทธิภาพมากขึ้นเพราะพื้นที่ทุกส่วนถูกนำมาใช้อย่างเต็มที่

3. เพิ่มยืดความสามารถของระบบประมวลผลแบบมัลติโปรแกรมมิง

ข้อเสีย

1. ต้องมีส่วนที่เป็นฮาร์ดแวร์และซอฟท์แวร์มาช่วยในการจัดการหน่วยความจำและจำต้องใช้เวลา พื้นที่และค่าใช้จ่ายในการทำงานบ้าง

2. หากมีการจัดการด้านการโยกย้ายเพจไม่ดีพอ อาจก่อให้เกิดภาวะที่เรียกว่าแทรชชิงขึ้น ทำให้ประสิทธิภาพในการทำงานของระบบลดลง

3. หากอัตราส่วนของหน่วยความจำหลักและหน่วยความจำเสมือนมีค่าน้อยเกินไป อาจจะก่อให้เกิดแทรชชิงได้เช่นเดียวกัน

---

[1]Stuart E. Madnick and John J. Donovan, Operating System. (Kogakusha : McGraw-Hill Book Co., Ltd.) pp. 164-165.

4. การกำหนดขนาดของเพจสำคัญต่อประสิทธิภาพในการทำงานของระบบเป็นอันมาก กล่าวคือ หากขนาดของเพจเล็กเกินไป ก็จะต้องมีการโยกย้ายเพจบ่อย ๆ ทำให้เสียเวลาในการประมวลผล หากมีขนาดใหญ่ไปก็อาจจะก่อให้เกิดเนื้อที่ที่ไม่ใช่งานภายในเพจมากขึ้น เมื่อนำมาลงในหน่วยความจำหลัก ก็จะทำให้เกิดเนื้อที่ที่ไม่ถูกใช้งานขึ้นภายใน ทำให้การใช้เนื้อที่ภายในหน่วยความจำมีประสิทธิภาพลดลง