หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

การบรรจุโปรแกรมให้ประจำในหน่วยความจำ

ระบบปฏิบัติการเอ็มเอส-ดอส เป็นระบบปฏิบัติการที่มีลักษณะการประมวลผลคำสั่งเป็นแบบงานเดียว (single task) กล่าวคือในขณะใดขณะหนึ่ง จะมีเพียงโปรแกรมเดียวเท่านั้นที่ปฏิบัติการอยู่ในหน่วยความจำ แต่ในความเป็นจริงแล้วระบบเอ็มเอส-ดอสสามารถบรรจุโปรแกรมเพื่อปฏิบัติการในหน่วยความจำได้ครั้งละหลาย ๆ โปรแกรมพร้อมกัน โดยจะมีเพียงหนึ่งคำสั่งของโปรแกรมใดโปรแกรมหนึ่งเท่านั้นที่ถูกประมวลผลในขณะหนึ่ง ๆ การบรรจุโปรแกรมเหล่านี้ให้ประจำอยู่ในหน่วยความจำพร้อมกันอาจทำได้โดย เมื่อเอ็มเอส-ดอสบรรจุโปรแกรมเข้ามาปฏิบัติการในหน่วยความจำ หลังจากโปรแกรมทำงานเสร็จก็จะให้โปรแกรมจบการทำงานและคืนการควบคุมการทำงานให้กับเอ็มเอส-ดอส โดยเอ็มเอส-ดอสจะไม่ดึงเอาโปรแกรมออกไปจากหน่วยความจำ ลักษณะของโปรแกรมเช่นนี้เรียกว่า โปรแกรมที่ประจำอยู่ในหน่วยความจำ (memory resident program) ตัวอย่างที่เห็นได้ชัดได้แก่ ไลบรารีในเวลาดำเนินการโปรแกรม (run-time library) เป็นโปรแกรมซึ่งรวบรวมฟังก์ชันการทำงานที่เป็นประโยชน์เก็บไว้ในหน่วยความจำตลอดเวลา และถูกเรียกใช้งานจากโปรแกรมอื่น ๆ ในเวลาที่มีการปฏิบัติการโปรแกรมนั้นด้วยการเรียก (call) หรืออินเตอร์รัปต์ (interrupt) เป็นต้น

วิธีบรรจุโปรแกรมให้ประจำในหน่วยความจำทำได้หลายวิธี เช่น การบรรจุโปรแกรมจากบรรทัดคำสั่ง (command line) และให้โปรแกรมจบการทำงานด้วยวิธีพิเศษคือใช้ ฟังก์ชันของระบบหมายเลข 31h (Keep Process) หรือใช้อินเตอร์รัปต์หมายเลข 27h (Terminate but Stay Resident) ซึ่งจะทำให้โปรแกรมจบการทำงานแต่ยังคงประจำอยู่ในหน่วยความจำ อีกวิธีหนึ่งคือ การบรรจุโปรแกรมหนึ่งจากบรรทัดคำสั่ง และให้โปรแกรมนั้นทำหน้าที่บรรจุอีกโปรแกรมหนึ่งเข้ามาในหน่วยความจำเพื่อใช้เป็นโปรแกรมซึ่งประจำในหน่วยความจำ โปรแกรมแรกเรียกว่าโปรแกรมพ่อ (parent program) ส่วนโปรแกรมที่ถูกบรรจุเรียกว่าโปรแกรมลูก (child program) วิธีนี้เรียกว่า การบรรจุและปฏิบัติการโปรแกรม (Load and Execute Program) [4] ทำได้โดยการเรียกใช้ฟังก์ชันของระบบหมายเลข 4Bh จากโปรแกรมพ่อ ซึ่งในที่นี้จะกล่าวถึงรายละเอียดของการบรรจุและปฏิบัติโปรแกรมเท่านั้น

3.1 การบรรจุโปรแกรมจากบรรทัดคำสั่ง

เมื่อระบบเอ็มเอส-ดอสบรรจุโปรแกรมเพื่อปฏิบัติการในหน่วยความจำจากบรรทัดคำสั่ง สิ่งแรกที่ระบบต้องทำคือ สร้างบล็อกสภาพแวดล้อม (environment block) เพื่อใช้เก็บเส้นทาง (path) ที่ใช้ค้นหาโปรแกรม COMMAND.COM (COMSPEC=<path>) เส้นทางปัจจุบัน (PATH=<search path>) และพารามิเตอร์อื่น ๆ ที่ถูกกำหนดด้วยคำสั่ง SET ค่าที่เก็บในบล็อกสภาพแวดล้อมจะถูกคัดลอกมาจากสภาพแวดล้อมของระบบ หลังจากนั้นระบบจะจองเนื้อที่ที่เป็นบล็อกของหน่วยความจำ (memory block) ให้กับโปรแกรม โดยสร้างบล็อกข้อมูลพิเศษสำหรับโปรแกรมเรียกว่า พีเอสพี (PSP ย่อมาจาก program segment prefix) มีขนาด 256 ไบต์ ไว้ตรงเนื้อที่ส่วนแรกของบล็อกของหน่วยความจำ ภายในพีเอสพีจะเก็บสารสนเทศที่จำเป็นสำหรับโปรแกรมในขณะที่ปฏิบัติการ จากนั้นจึงอ่านโปรแกรมเข้ามาต่อท้ายพีเอสพีและส่งการทำงานไปยังจุดเริ่มต้นของโปรแกรมเพื่อปฏิบัติการต่อไป

3.2 การบรรจุและปฏิบัติการโปรแกรมด้วยฟังก์ชันของระบบหมายเลข 4Bh

การบรรจุและปฏิบัติการโปรแกรม ด้วยฟังก์ชันของระบบหมายเลข 4Bh จะทำงานได้ 2 ลักษณะคือ

3.2.1 การปฏิบัติการโปรแกรม (Execute) เป็นการบรรจุโปรแกรมลูกเข้ามาในหน่วยความจำแล้วให้โปรแกรมลูกปฏิบัติงานโดยปราศจากการควบคุมจากโปรแกรมพ่อ มีข้อกำหนดคือ กำหนดค่าของรีจิสเตอร์ AL เท่ากับ 0 และกำหนดพารามิเตอร์อื่น ๆ ให้เหมาะสม

3.2.2 การบรรจุซ้อนทับ (Load Overlay) เป็นการบรรจุโปรแกรมลูกเข้ามาในหน่วยความจำแล้วคืนการควบคุมการทำงานให้กับโปรแกรมพ่อ มีข้อกำหนดคือ กำหนดค่าของรีจิสเตอร์ AL เท่ากับ 3 และกำหนดพารามิเตอร์อื่น ๆ ให้เหมาะสม

ในที่นี้จะกล่าวถึงเฉพาะรายละเอียดของการปฏิบัติการโปรแกรมเท่านั้น ได้แก่ ขั้นตอนการทำงาน และการกำหนดพารามิเตอร์ที่สำคัญ

3.3 ขั้นตอนการบรรจุและปฏิบัติการโปรแกรม

3.3.1 เมื่อระบบเอ็มเอส-ดอสบรรจุโปรแกรมพ่อจากบรรทัดคำสั่งในหน่วยความจำ ระบบจะจองเนื้อที่ชั่วคราวในหน่วยความจำทั้งหมดให้กับโปรแกรมนั้น ทำให้ไม่มีเนื้อที่ในหน่วยความจำเหลือพอที่จะบรรจุโปรแกรมอื่นได้อีก ดังนั้นก่อนที่โปรแกรมพ่อจะบรรจุโปรแกรมลูกได้จะต้องมีการกำหนดขนาดของเนื้อที่ที่จองให้กับโปรแกรมพ่อใหม่ให้มีขนาดเท่ากับที่โปรแกรมพ่อ

จำเป็นต้องใช้งานจริง ๆ โดยเรียกใช้ฟังก์ชันของระบบหมายเลข 4Ah (Modify Allocated Memory) เพื่อลดขนาดของเนื้อที่หน่วยความจำให้มีขนาดเท่าที่โปรแกรมต้องใช้งาน มีรูปแบบการใช้คำสั่งซึ่งเขียนเป็นภาษาแอสเซมบลีได้ดังนี้

```
mov    es, block
mov    bx, newsize
mov    ah, 4Ah
int    21h
```

โดย block คือค่าเซกเมนต์ของหน่วยความจำที่จองให้กับโปรแกรม

newsize คือขนาดของหน่วยความจำใหม่ มีหน่วยเป็นพารากราฟ (1 พารากราฟ เท่ากับ 16 ไบต์)

3.3.2 สร้างบล็อกของพารามิเตอร์เพื่อใช้ปฏิบัติการ (EXEC parameter block) บล็อกนี้เป็นส่วนสำคัญซึ่งฟังก์ชันจะนำไปสร้างพีเอสพีของโปรแกรมลูก มีขนาด 14 ไบต์ และมีรายละเอียดดังนี้

ไบต์ที่ 1-2 เก็บค่าเซกเมนต์ของบล็อกสภาพแวดล้อมของโปรแกรมลูก ซึ่งสร้างไว้ที่โปรแกรมพ่อ ถ้ากำหนดเป็น 0 โปรแกรมลูกจะใช้บล็อกสภาพแวดล้อมของโปรแกรมพ่อ

ไบต์ที่ 3-6 เก็บตำแหน่งที่อยู่ของส่วนที่ใช้รับข้อความจากบรรทัดคำสั่งสำหรับโปรแกรมลูก ซึ่งกำหนดไว้ที่โปรแกรมพ่อ

ไบต์ที่ 7-10 เก็บตำแหน่งที่อยู่ของเอฟซีบี (FCB ย่อมาจาก file control block) หมายเลข 1 ของโปรแกรมพ่อ

ไบต์ที่ 11-14 เก็บตำแหน่งที่อยู่ของเอฟซีบี หมายเลข 2 ของโปรแกรมพ่อ

3.3.3 กำหนดชื่อเต็มของโปรแกรมลูกประกอบด้วยชื่อเส้นทาง และชื่อโปรแกรมตามด้วยนามสกุลเป็น .EXE หรือ .COM ชื่อโปรแกรมลูกจะต้องเป็นตัวอักษรตัวใหญ่ทั้งหมดและปิดท้ายชื่อด้วยนัลซึ่งมีค่า 00h (เรียกว่า ASCIIZ)

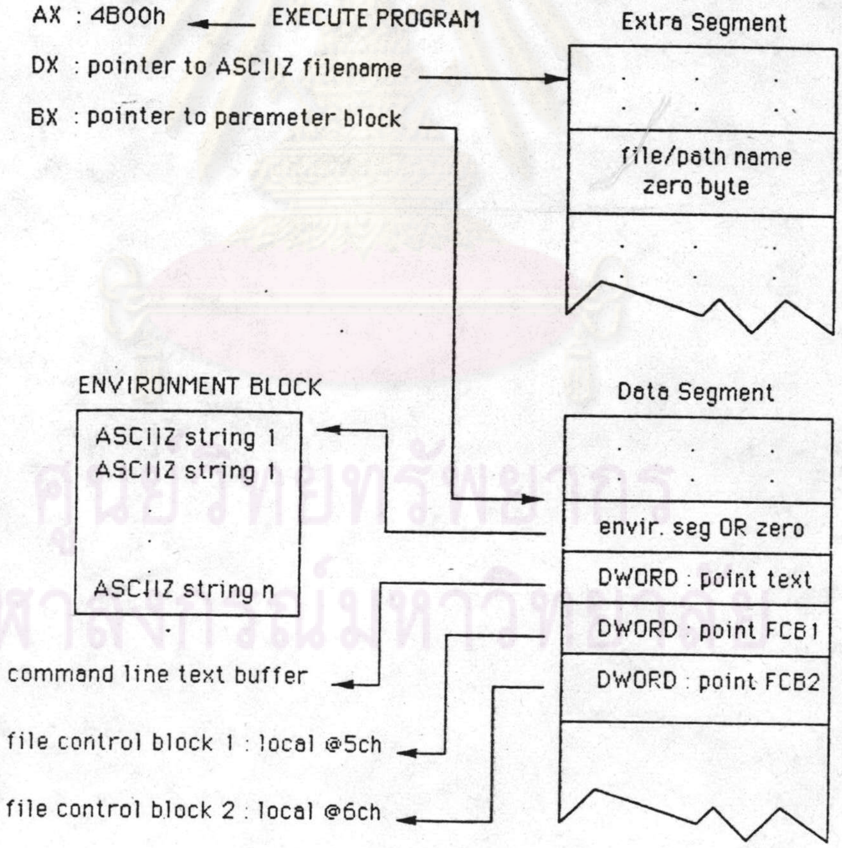
3.3.4 เก็บบันทึกสภาวะเดิมของรีจิสเตอร์ทุกตัวที่ใช้ไว้ในสแตกของโปรแกรมพ่อและเก็บค่าของสแตกเซกเมนต์ (SS) และตัวชี้สแตก (SP) ไว้ในเนื้อที่ที่กำหนดไว้ในเซกเมนต์ของโปรแกรมพ่อ

3.3.5 เรียกใช้ฟังก์ชัน 4Bh เพื่อบรรจุและปฏิบัติการโปรแกรม โดยมีการกำหนดค่าให้กับรีจิสเตอร์ซึ่งเขียนเป็นภาษาแอสเซมบลีได้ดังนี้

```
mov    ah, 4Bh
mov    al, 0h
mov    dx, ASCIIZ file name
mov    bx, EXEC parameter block
int    21h
```

โดย ASCIIZ file name คือชื่อโปรแกรมที่จะถูกบรรจุและปฏิบัติการ
EXEC parameter block คือค่าเซกเมนต์ของบล็อกของพารามิเตอร์

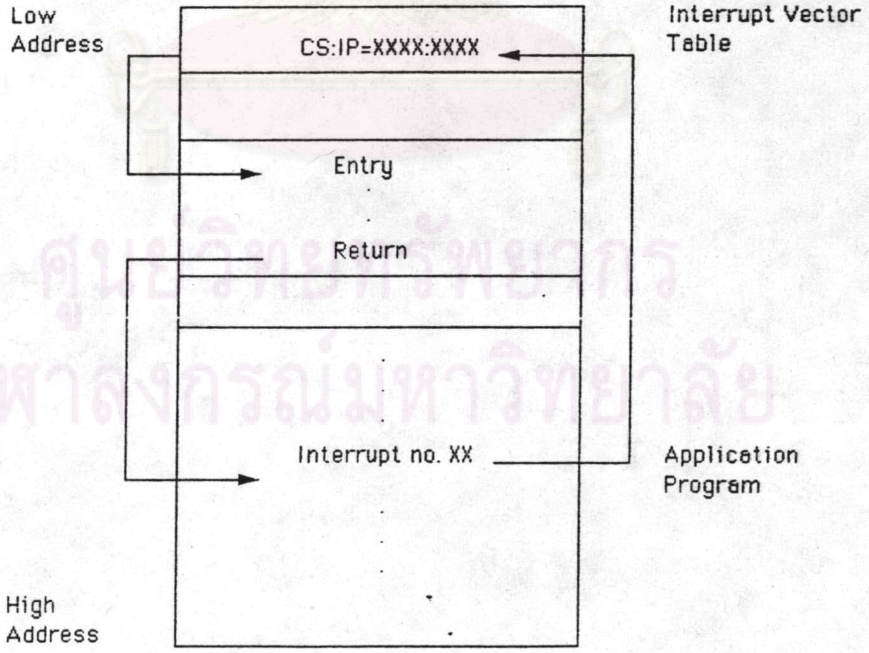
3.3.6 หลังจากนั้นโปรแกรมลูกจะถูกปฏิบัติการต่อไป โดยปราศจากการควบคุมใด ๆ จากโปรแกรมพ่อ และเมื่อโปรแกรมลูกทำงานเสร็จ ก็จะคืนการควบคุมให้กับโปรแกรมพ่อต่อไป



ภาพที่ 3.1 การกำหนดรีจิสเตอร์และบล็อกของพารามิเตอร์ของฟังก์ชันของระบบหมายเลข 4Bh

3.4 การเรียกใช้ฟังก์ชันของโปรแกรมที่ประจำในหน่วยความจำ

เมื่อโปรแกรมถูกติดตั้งให้ประจำในหน่วยความจำ จะต้องมีวิธีการจัดการให้โปรแกรมอื่นที่กำลังปฏิบัติการอยู่ในหน่วยความจำพร้อมกันสามารถเรียกใช้ฟังก์ชันใด ๆ ของโปรแกรมนั้นได้ วิธีการหนึ่งคือจะต้องทราบตำแหน่งของหน่วยความจำที่เป็นจุดเริ่มต้นการทำงานของฟังก์ชันเหล่านั้น และเก็บตำแหน่งนั้นไว้ในเนื้อที่หน่วยความจำส่วนหนึ่งซึ่งโปรแกรมทั่วไปสามารถเข้าถึงได้ และไม่ส่งผลกระทบกับการทำงานของระบบ เนื้อที่หน่วยความจำนี้คือ ตารางอินเตอร์รัปต์เวคเตอร์ (Interrupt Vector Table) เป็นตารางขนาด 1024 ไบต์ อยู่ที่ตำแหน่งแรกสุดของหน่วยความจำ (0000:0000) แบ่งเป็น 256 ช่อง ช่องละ 4 ไบต์ แต่ละช่องของตารางใช้อ้างอิงการทำงานต่าง ๆ ของระบบได้จากค่าเซกเมนต์และออฟเซตที่เก็บอยู่ ทำให้สามารถอ้างหมายเลขการอินเตอร์รัปต์ได้ 256 หมายเลข โดย 64 หมายเลขแรกคือ 00h ถึง 3Fh ใช้สำหรับการอินเตอร์รัปต์การทำงานของฮาร์ดแวร์ ส่วนหมายเลขที่เหลือคือ 40h ถึง FFh เป็นบริเวณที่ว่างที่ผู้เขียนโปรแกรมสามารถใช้เป็นเนื้อที่สำหรับเก็บตำแหน่งเริ่มต้นการทำงานของฟังก์ชันได้ และให้โปรแกรมอื่นเรียกใช้ฟังก์ชันที่ประจำในหน่วยความจำได้ด้วยการอินเตอร์รัปต์หมายเลขของตารางอินเตอร์รัปต์เวคเตอร์ที่ใช้เก็บตำแหน่งเริ่มต้นการทำงานของฟังก์ชันนั้น



ภาพที่ 3.2 การเรียกใช้ฟังก์ชันของโปรแกรมที่ประจำในหน่วยความจำด้วยการอินเตอร์รัปต์