**การจัดการหน่วยความจำหลัก**

**แบงออกเป็น 4 แบบคือ**

1) ระบบโปรแกรมเดียว (Monoprogramming)

2) ระบบหลายโปรแกรมที่กำหนดขนาดของพาร์ติชันคงที่ (Multiprogramming with Fixed Partition)

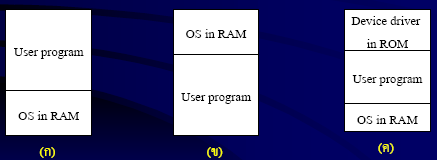
3) ระบบหลายโปรแกรมที่กำหนดขนาดของพาร์ติชันไม่คงที่ (Dynamic Partition)

4) การจัดการแบบระบบบัดดี้ (Buddy System)

1. **ระบบโปรแกรมเดียว (Monoprogramming)**

ในการจัดการหน่วยความจำแบบ monoprogramming นั้น หน่วยความจำเกือบทั้งหมดจะถูกจัดสรรให้กับโปรแกรมเพียงโปรแกรมเดียว ส่วนที่เหลือจะใช้เก็บตัวระบบปฏิบัติการเอง โปรแกรมทั้งโปรแกรมจะถูก load เข้าไปเก็บอยู่ในหน่วยความจำเพื่อรอการ execute เมื่อโปรแกรม run เสร็จหรือจบการทำงาน โปรแกรมใหม่ก็จะ load เข้าไปแทนที่โปรแกรมเดิม

จะมีโปรแกรมเพียง 1 โปรแกรมทำงาน ณ เวลาขณะใดขณะหนึ่ง ดังนั้นการใช้หน่วยความจำจะมีเพียงโปรแกรมนั้นกับระบบปฎิบัติการเท่านั้นซึ่งสามารถแบ่งได้เป็น 3 แบบ ดังรูปที่ 1



รูปที่ 1

**การจัดการหน่วยความจำแบบ**

**ก.** จะพบมากในเครื่องคอมพิวเตอร์ขนาดใหญ่ในสมัยก่อน  
แบบ

**ข.** จะใช้เครื่องคอมพิวเตอร์ขนาดเล็ก ได้แก่ Plam PDA  
และการจัดการหน่วยความจำแบบ

**ค.** จะพบในเครื่อง PC ในสมัยการใช้ DOS

**หน้าที่ของ memory manager แบบ monoprogramming**

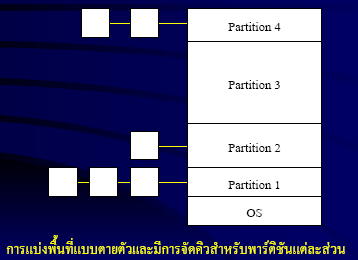
คือ (1) ทำการ load โปรแกรมเข้าไปในหน่วยความจำ (2) run โปรแกรมที่ load เข้ามา (3) แทนที่โปรแกรมที่ run เสร็จแล้วด้วยโปรแกรมต่อไป อย่างไรก็ดี การจัดการหน่วยความจำแบบนี้ก็มีปัญหาหลายประการดังนี้  
 1. โปรแกรมจะต้องมีขนาดที่สามารถ load เข้าไปอยู่ในหน่วย ความจำได้ทั้งหมด ถ้าหน่วยความจำมีขนาดเล็กกว่าขนาดของโปรแกรม ก็จะไม่สามารถ run โปรแกรมนั้นได้  
 2. เมื่อมีโปรแกรมหนึ่งกำลัง run อยู่ โปรแกรมอื่นๆจะต้องรอ ในระหว่างที่โปรแกรมหนึ่งกำลังถูก execute นั้นโดยปกติจะมีการรับข้อมูลเข้ามาจากอุปกรณ์นำเข้า และส่งผลลัพธ์ออกทางอุปกรณ์แสดงผล อุปกรณ์ I/O เหล่านี้ทำงานช้ากว่า CPU มาก ดังนั้นระหว่างที่อุปกรณ์ I/O กำลังทำงาน CPU ก็จะว่าง ไม่อาจทำ งานอย่างอื่นได้เนื่องจากไม่มีโปรแกรมอื่นอยู่ในหน่วยความจำ วิธี การแบบนี้นับเป็นการใช้หน่วยความจำและ CPU ไม่สู้จะมีประ สิทธิภาพเท่าที่ควร

1. **ระบบหลายโปรแกรมที่กำหนดขนาดพาร์ติชันคงที่ (Multiprogramming with Fixed Partition)**

เมื่อระบบเริ่มต้นทำงาน ระบบจะแบ่งพื้นที่หน่วยความจำออกเป็น n พาร์ติชั่น เมื่อมีโปรเซสเข้ามาทำงานระบบจะต้องจัดการนำโปรเซสเหล่านั้นเข้าแถวเพื่อนำไปใส่ในพาร์ติชันส่วนที่เล็กที่สุดที่เพียงพอให้โปรเซสนั้นทำงานได้

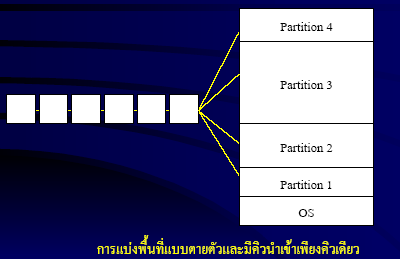
แต่เนื่องจากมีการกำหนดให้ขนาดของพาร์ติชันตายตัว ระบบก็จะสูญเสียพื้นที่ที่ไม่ได้ใช้งานในพาร์ติชันนั้นไป ซึ่งเรียกว่า การสูญเปล่าของพื้นที่ย่อยภายใน (Internal Fragmentation)

1. **Multiple input queue**



รูปที่ 2

1. **Single input queue**



รูปที่ 3

การแบ่งพื้นที่แบบตายตัวและมีคิวนำเช้าเพียงคิวเดียว เป็นการแก้ปัญหาการที่คิวขนาดเล็กจะเต็มอยู่ตลอดเวลาในขณะที่คิวของพาร์ติชันขนาดใหญ่ยังคงว่างอยู่ และจะเลือกงานที่มีขนาดใหญ่ที่สุดในคิวมาใช้พาร์ติชันที่ใหญ่ที่สุดในขณะนั้น เพื่อให้ระบบใช้งานหน่วยความจำได้เต็มประสิทธิภาพ

แต่วิธีการดังกล่าวยังคัดค้านกับทฤษฎีไม่ให้งานที่มีขนาดเล็กใช้พาร์ติชันที่มีขนาดใหญ่ ซึ่งจะจัดการกับปัญหาโดย

1) ให้สร้างพาร์ติชันที่มีขนาดเล็กสักกลุ่มหนึ่งที่จะให้งานขนาดเล็กทำงาน

2) กำหนดขึ้นมาว่างานที่เลือกเช้ามาใช้งานนั้นจะถูกมองข้ามได้ไม่เกินจำนวนครั้งที่กำหนด

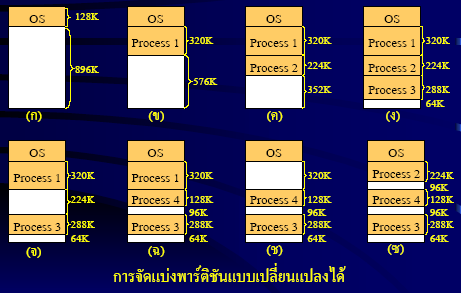
ข้อเสียของการจัดพาร์ติชันแบบตายตัว

1) พาร์ติชันที่กำหนดให้มีขนาดและจำนวนที่ตายตัวเมื่อระบบเริ่มทำงานนั้น จะเป็นตัวจำกัดจำนวนของโปรเซสที่สามารถทำงานในระบบ

2)เนื่องจากระบบมีการกำหนดพาร์ติชันที่ตายตัว ดังนั้นงานที่มีขนาดเล็กจะไม่สามารถใช้พื้นที่ของพาร์ติชันได้อย่างเต็มประสิทธิภาพ

**3. ระบบที่กำหนดขนาดของพาร์ติชันให้เปลี่ยนแปลงได้** **(Dynamic Partition)**

ระบบจะจัดการให้พาร์ติชันสามารถเปลี่ยนแปลงได้ทั้งขนาดและจำนวนของพาร์ติชั่น เมื่อมีโปรเซสเข้ามาใช้งาน หน่วยความจำหลักจะถูกแบ่งเป็นพาร์ติชันให้พอดีกับขนาดของโปรแกรม ซึ่งเป็นการแก้ปัญหาวิธีการจัดแบ่งหน่วยความจำแบบขนาดพาร์ติชันคงที่

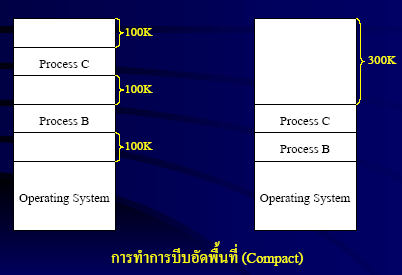


รูปที่ 4

**ปัญหา**

เมื่อระบบทำงานไปได้สักระยะหนึ่ง จะพบว่ามีช่องว่างเกิดขึ้นอย่างมากมายในหน่วยความจำหลัก ทำให้การใช้งานมีประสิทธิภาพน้อยลงไปเรียกช่องเล็กๆ เหล่านี่ว่า การสูญเปล่าของพื้นที่ย่อยภายนอก (External Fragmentation)

วิธีป้องกันปัญหาการสูญเปล่าพื้นที่ย่อยภายนอก คือ การบีบอัดพื้นที่สูญเปล่า (Compact) เหล่านั้น โดยระบบปฎิบัติการสามารถเลื่อนพื้นที่ของโปรเซสต่างๆ ที่ใช้งานให้มาชิดกัน ซึ่งจะทำให้พื้นที่ว่างเปล่าเหลือเพียงแค่ 1 พาร์ติชันใหญ่เท่านั้น



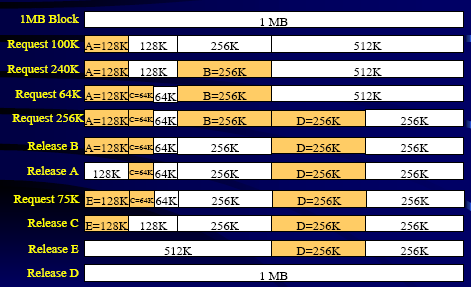
รูปที่ 5

การบีบอัดพื้นที่จะมีความยุ่งยากในการจัดการ ใช้เวลาในการทำงานมาก และเปลืองเวลาทำงานของ Process พอสมควร เพราะมีการย้ายตำแหน่งข้อมูล (Relocation) ซึ่งจะมีความสามารถในการย้ายโปรแกรมจากพื้นที่หนึ่งไปยังพื้นที่หนึ่งในหน่วยความจำหลัก โดยไม่ทำให้จุดบอกตำแหน่งการเข้าถึงข้อมูลนั้นผิดพลาดไปด้วย

1. **การจัดการแบบระบบบัดดี้ (Buddy System)**

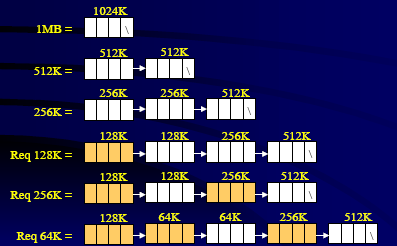
ระบบบัดดี้ จะเป็นการจัดการหน่วยความจำแบบผสมผสานระหว่างระบบการแบ่งพาร์ติชั่นแบบตายตัวและแบบเปลี่ยนแปลงขนาดพาร์ติชันได้

เมื่อมีโปรเซสเข้ามาในระบบ พื้นที่หน่วยความจำจะแบ่งออกเป็น 2 ส่วนเท่าๆกัน แล้วนำขนาดของโปรเซสมาเปรียบเทียบกับขนาดของหน่วยความจำแต่ละส่วน หากขนาดโปรเซสตัวกว่าขนาดของพื้นที่หน่วยความจำก็จะวางโปรเซสนั้นลงในหน่วยความจำทันที แต่หากขนาดของโปรเซสมีขนาดเล็กกวาขนาดของหน่วยความจำก็จะทำการแบ่งหน่วยความจำส่วนนั้นออกเป็น 2 ส่วนอีก แล้วจะเปรียบเทียบขนาดของโปรเซสอีก หากโปรเซสมีขนาดใหญ่กว่าขนาดหน่วยความจำก็จะวางโปรเซสลงในพื้นที่หน่วยคามจำส่วนนั้น และหากขนาดของโปรเซสเล็กกว่าขนาดของส่วนหน่วยความจำที่แบ่งได้ก็จะทำการแบ่งต่อไป และจัดวางโปรเซสเมื่อพบว่าขนาดของส่วนหน่วยความจำที่แบ่งได้ในขณะนั้นเล็กกว่าขนาดของโปรเซส



รูปที่ 6

ระบบบัดดี้จะจัดการและตรวจสอบเนื้อที่หน่วยความจำโดยการใช้วิธีการจัดการแบบ Linked Lists



รูปที่ 7