บทที่ 3 การจัดการหน่วยความจำ

(Memory Management)

T.Kunlaya Charoenmongkonvilai

การจัดการหน่วยความจำ (Memory Management)

หน่วยความจำเป็นส่วนที่สำคัญที่สุดในระบบคอมพิวเตอร์ ถือเป็นศูนย์กลางให้การดำเนินการด้านต่างๆ ในระบบคอมพิวเตอร์เป็นไปอย่างราบรื่นและมีประสิทธิภาพสูงสุด แบ่งออกเป็น 2 ส่วน ได้แก่

1.หน่วยความจำหลัก (Main Memory)

2.หน่วยความจำเสมือน (Virtual Memory)

โดยแต่ละวิธีในการจัดเก็บข้อมูลทั้งสองส่วนมีข้อดีและข้อเสียต่างกันขึ้นอยู่กับซอฟต์แวร์และฮาร์ดแวร์ที่เลือกใช้ว่าสอดคล้องและสนับสนุนการทำงานและวิธีการที่จัดเก็บในหน่วยความจำที่เลือกใช้มากน้อยเพียงใด ดังรูปที่ 2.1

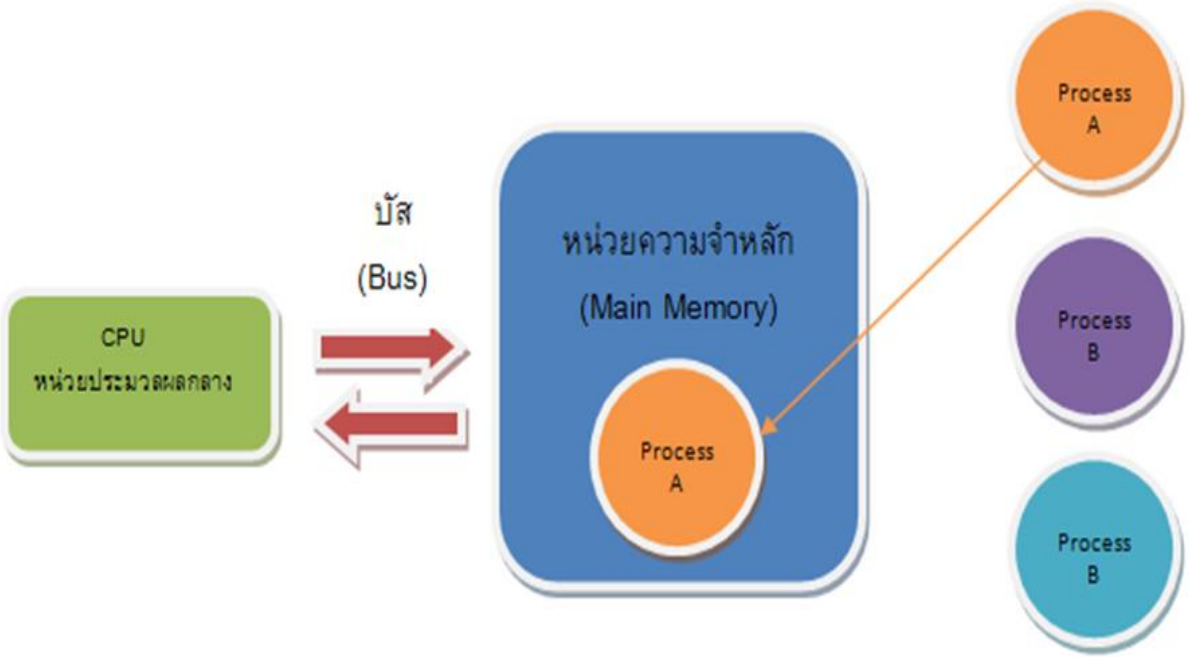
Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

หน่วยความจำหลัก (Main Memory)

ประกอบไปด้วยอารเรย์ขนาดใหญ่ (large array) ซึ่งภายในประกอบไปเวิร์ด (words) และไบต์ (bytes) ซึ่งแต่ละที่จะมีเลขตำแหน่ง (address) เป็นของตัวเอง

นอกจากนี้หน่วยความจำหลักยังทำหน้าที่เก็บชนิดกระบวนการในการประมวลผลคำสั่ง (a typical instruction-execution cycle) เพื่อให้หน่วยประมวลผลกลาง (Central Processing Unit: CPU) นำไปใช้ในการประมวลแล้วจึงส่งผลลัพธ์ของคำสั่งนั้นๆ กลับมาจัดเก็บกลับไว้ในหน่วยความจำหลัก

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.



รูปที่ 2.1.แสดงกระบวนการทำงานของหน่วยความจำหลัก (Main Memory)

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

หน่วยความจำเสมือน (Virtual Memory)

เป็นเทคนิคที่อนุญาตให้โปรเซส (Process) สามารถประมวลผลได้นอกหน่วยความจำหลักโดยไม่ต้องคำนึงถึงขนาดพื้นที่ใช้ในการประมวลผลว่าเพียงพอกับขนาดของโปรแกรมหรือไม่

นอกจากนี้ยังง่ายต่อการแชร์ไฟล์ (Share files) พื้นที่ว่าง (Address Space) และเพิ่มประสิทธิภาพให้โปรเซสทำงานได้เร็วขึ้น เพราะไม่ต้องคอยตรวจสอบขนาดของหน่วยความจำทางกายภาพ (Physical Memory)

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

การเชื่อมโยงตำแหน่ง (Address Binding)

- โดยทั่วไปโปรเซสที่จะถูกนำไปประมวลผลจะขึ้นอยู่กับการจัดการหน่วยความจำที่เลือกใช้ โปรเซสอาจจะถูกเคลื่อนย้ายกลับไปกลับมาระหว่างดิสก์และหน่วยความจำก่อนที่มันจะถูกประมวลผล
- โดยระบบปฏิบัติการจะมีการเก็บโปรเซสที่รอประมวลผลตามลำดับคิว (input queue) ของโปรเซสที่จะนำเข้ามาประมวลผลในหน่วยความจำร่วม ทั้งยังเชื่อมโยงกับค่าเริ่มต้นของตำแหน่ง ระบบคอมพิวเตอร์มักเริ่มต้นค่าตำแหน่งเชื่อมโยงที่ค่า 00000 อีกยังแบ่งค่าที่เชื่อมโยงตำแหน่ง ได้เป็น ค่าจริง (Absolute address) ของโปรเซสที่อยู่ในหน่วยความจำ และค่าตำแหน่งที่สัมพันธ์ (Relative address) หรือชุดคำสั่งต่างๆ ที่ได้รับหลังจากการคอมไฟล์

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

การจำแนกการเชื่อมโยงของคำสั่งและตำแหน่งของข้อมูลในหน่วยความจำสามารถแบ่งแต่ละขั้นตอนตามช่วงเวลาได้ดังนี้

1.ช่วงเวลาคอมไฟล์ (Compile time)

คือ ช่วงเวลาที่คำสั่งหรือข้อมูลถูกแปลคำสั่งโดยการคอมไฟล์โดยตัวคอมไฟเลอร์ (Compiler) โดยตัวคอมไฟเลอร์จะค้นหาตำแหน่งจริง (Absolute Code) ในหน่วยความจำเมื่อพบแล้วจะสร้างโค้ดทำให้ระบบปฏิบัติการสามารถประมวลผลคำสั่งหรือข้อมูลนั้นได้ทันที แต่หากตำแหน่งจริงถูกเปลี่ยนแปลง ตัวคอมไฟเลอร์ก็จะทำการรีคอมไฟล์ (Recompile) โค้ดคำสั่งหรือข้อมูลนั้นใหม่ทุกครั้ง

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

การจำแนกการเชื่อมโยงของคำสั่งและตำแหน่งของข้อมูลในหน่วยความจำสามารถแบ่งแต่ละขั้นตอนตามช่วงเวลาได้ดังนี้ (ต่อ)

2.ช่วงเวลาโหลด (Load time)

ช่วงเวลาที่คำสั่งหรือข้อมูลถูกโหลดเข้าสู่หน่วยความจำโดยลำดับแรกตัวคอมไพเลอร์จะสร้างโค้ดที่สามารถประมวลผลได้ทันที (Relocation code)ขึ้นมาก่อน หลังจากคำสั่งหรือข้อมูลถูกโหลดเข้าสู่หน่วยความจำแล้ว ตัวคอมไพเลอร์จะทำการแปลตำแหน่งที่โหลดเข้ามาให้เป็นตำแหน่งจริง (Absolute Code) เพื่อให้ระบบปฏิบัติการสามารถประมวลผลคำสั่งหรือข้อมูลนั้นได้โดยไม่ต้องเสียเวลาในการคอมไฟล์ใหม่ทุกครั้ง แต่จะเสียเวลาเฉพาะตอนโหลดคำสั่งหรือข้อมูลนั้นเข้ามาในหน่วยความจำ

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

การจำแนกการเชื่อมโยงของคำสั่งและตำแหน่งของข้อมูลในหน่วยความจำสามารถแบ่งแต่ละขั้นตอนตามช่วงเวลาได้ดังนี้ (ต่อ)

3.ช่วงเวลาประมวลผล (Execution time)

ช่วงเวลาที่คำสั่งหรือข้อมูลถูกประมวลผล โดยตัวคอมไพเลอร์จะทำการเชื่อมโยงตำแหน่งและแปลโค้ดคำสั่งหรือข้อมูลของตำแหน่งนั้นๆ เข้าไปเก็บไว้ในหน่วยความจำขณะที่ระบบปฏิบัติการกำลังประมวลผล (Run time) ทำให้ระบบปฏิบัติการต้องเสียเวลาในการแปลตำแหน่งคำสั่งหรือข้อมูลต่างๆ ก่อนถูกนำมาเข้าสู่กระบวนการประมวลผลทุกครั้ง

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

การเชื่อมโยงระหว่างพื้นที่ทางกายภาพกับพื้นที่ทางตรรกะ (Logical- Versus Physical-Address Space)

- จะมีการอ้างอิงถึงตำแหน่งที่เกี่ยวข้องอยู่ 2 ประเภทคือ

1.ตำแหน่งพื้นที่ทางตรรกะ (Logical Address Space) หรือเรียกอีกชื่อหนึ่งว่า ตำแหน่งเสมือน (Virtual Address) ซึ่งถูกสร้างขึ้น (Generate) โดยหน่วยประมวลผลกลาง (CPU) เพื่อใช้ในการแลกเปลี่ยนข้อมูล โดยกลุ่มของตำแหน่งพื้นที่ทางตรรกะ (Logical Address Space) ทั้งหมดถูกสร้างโดยโปรแกรม

2.ตำแหน่งพื้นที่ทางกายภาพ (Physical Address Space)

คือตำแหน่งที่อยู่ในหน่วยความจำหลัก (Memory Unit) และทำงานโดยตอบสนอง (Corresponding) กับตำแหน่งทางตรระกะ (Logical Address) เสมอ

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

การโหลดแบบพลวัต (Dynamic Loading)

- เนื่องจากพื้นที่ที่ใช้ในการประมวลผลในหน่วยความจำทางกายภาพ (Physical Memory) มีการจำกัดขนาดข้อมูล ดังนั้นโปรแกรมคำสั่งและข้อมูลทั้งหมดของกระบวนการ (Process) จะต้องมีขนาดเล็กกว่าหน่วยความจำทางกายภาพ (Physical Memory) ในขณะที่โหลดข้อมูลทั้งหมดเข้าสู่พื้นที่ว่างในหน่วยความจำ

- ในแต่ละครั้งจะต้องมีการตรวจสอบขนาดของข้อมูล เพื่อจะช่วยให้การประมวลผลโปรแกรมคำสั่ง และการโหลดข้อมูลเข้าหน่วยความจำทำได้รวดเร็วขึ้น ลักษณะการทำงานดังกล่าวนี้เรียกว่า การโหลดแบบพลวัต (Dynamic Loading) ซึ่งวิธีการทำงานจะไม่โหลดโปรแกรมย่อย (Routine) จนกว่าจะมีการเรียกใช้งาน (Call) ทำให้ไม่สิ้นเปลืองเนื้อที่ในหน่วยความจำ

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

การแบ่งส่วน (Overlay)

- วิธีนี้ใช้ในกรณีที่โปรเซสมีขนาดใหญ่กว่าหน่วยความจำที่จัดเก็บ จำเป็นที่จะต้องจัดสรรหน่วยความจำให้เหมาะสมโดยการแบ่งส่วน (Overlay) หลักการทำงานของการแบ่งส่วนโดยที่คำสั่งและข้อมูลจะถูกเก็บอยู่ในหน่วยความจำตามขนาดของหน่วยความจำที่มี
- ซึ่งจะต้องอาศัยการทำงานของตัวขับในการแบ่งส่วน (Overlay Driver) เพื่อโหลดคำสั่งที่ต้องการใช้งานหน่วยความจำเข้ามาใช้งานจนเสร็จก่อนแล้วจึงสลับให้คำสั่งอื่นที่ต้องการใช้งานหน่วยความจำเข้ามาทำใช้งานหน่วยความจำต่อไป

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

การสับเปลี่ยน (Swapping)

- เป็นวิธีที่ในการสลับโปรเซสที่ต้องการจะเข้าหรือออกเพื่อประมวลผลในหน่วยความจำ โดยหลักการจะต้องสลับโปรเซสเก่าออกมาก่อนแล้วจึงนำเข้าโปรเซสใหม่เข้าไปใช้งานหน่วยความจำโดยนำเข้าไปเก็บไว้ในระบบสนับสนุนการจัดเก็บ (Backing Store)
- โดยใช้ Priority-base Scheduling Algorithms ถ้าโปรเซสใดมี Higher-priority process เข้ามาถึงและต้องการจะใช้บริการหน่วยความจำ ตัวจัดการหน่วยความจำสามารถที่จะทำการสลับเอา Lower-priority process ออกไปก่อนเพื่อให้โปรเซสใดมี Higher-priority process ทำงานให้เสร็จก่อน โปรเซสที่มี Lower-priority process จึงจะสลับกลับมา (Swap back) เพื่อทำงานในหน่วยความจำต่อไปได้

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

การจัดสรรหน่วยความจำที่ต่อเนื่องกัน

(Continues Memory Allocation)

- หน่วยความจำหลักจำเป็นจะต้องอำนวยความสะดวกให้กับระบบปฏิบัติการและความหลากของผู้ใช้งานโปรเซส ดังนั้นในแต่ละโปรเซส (Process) จึงมีความต้องการใช้พื้นที่ในหน่วยความจำอย่างต่อเนื่อง จึงเป็นหน้าที่ของระบบปฏิบัติการในการจัดสรรพื้นที่หน่วยความจำในหน่วยความจำให้มีประสิทธิภาพสูงสุด
- ซึ่งโดยทั่วไปแล้วหน่วยความจำหลัก (Main Memory) จะถูกแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำระดับบน (High Memory) ซึ่งเป็นส่วนที่ใช้ติดต่อกับส่วนของผู้ใช้ (User) และหน่วยความจำระดับล่าง (Low Memory) ซึ่งเป็นส่วนของระบบปฏิบัติการ (Operating System) ซึ่งแสดงได้ดังรูปที่ 2.6

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

การจัดสรรพื้นที่ในหน่วยความจำแบ่งออกเป็น 2 รูปแบบ ดังนี้

1.การจัดสรรหน่วยความจำแบบพื้นที่เดียว (Single-partition Allocation)

วิธีนี้หน่วยความจำจะไม่ถูกแบ่งพื้นที่ โดยโปรเซสในส่วนของระบบปฏิบัติการจะอยู่ในส่วนหน่วยความจำระดับล่าง (Low Memory) และส่วนของโปรเซสของผู้ใช้อยู่ในส่วนหน่วยความจำระดับบน (High Memory) โดยเกี่ยวข้องกับรีจิสเตอร์ย้ายตำแหน่ง (Relocation Register) ในการแยกโปรเซส (Process) ในส่วนของผู้ใช้ออกจากส่วนของระบบปฏิบัติการและรีจิสเตอร์ขอบเขต (Limit Register) เป็นรีจิสเตอร์ที่ใช้ระบุขนาดของค่าตำแหน่งทางตรรกะ (Logical Address) โดยค่าตำแหน่งทางตรรกะ (Logical Address) ต้องน้อยกว่ารีจิสเตอร์ขอบเขต (Limit Register) เสมอ

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

การจัดสรรพื้นที่ในหน่วยความจำแบ่งออกเป็น 2 รูปแบบ ดังนี้ (ต่อ)

2.การจัดสรรหน่วยความจำแบบหลายพื้นที่ (Multiple-partition Allocation)

วิธีนี้หน่วยความจำจะถูกแบ่งตามจำนวน Process มีวิธีการจัดสรร 2 รูปแบบ ดังนี้2.1 การแบ่งแบบคงที่ (Fixed Partition) โดยการแบ่งพื้นที่หน่วยความจำออกเป็นหลาย ๆ พาร์ทิชั่น (Partition) แต่ละพาร์ทิชั่นมีขนาดเท่ากันและบรรจุโปรเซสอยู่ภายในเพียงหนึ่งโปรเซสเท่านั้น โดยที่จำนวนพาร์ทิชั่นถูกกำหนดโดยจำนวนโปรเซสที่มีอยู่ในหน่วยความจำ

ข้อเสียของวิธีคือ หากโปรเซสที่บรรจุอยู่ในหน่วยความจำมีขนาดเล็กกว่าพาร์ทิชั่นที่กำหนดจะทำให้เหลือพื้นที่ในหน่วยความจำ (พาร์ทิชั่นมีพื้นที่ว่างเหลือ) เรียกพื้นที่ว่างที่เหลือนี้ว่า “Internal Fragmentation” กรณีที่ขนาดของโปรเซสมีขนาดใหญ่กว่าพาร์ทิชั่นที่กำหนดก็จะไม่สามารถนำโปรเซสนั้นเข้าไปใช้งานพื้นที่ในหน่วยความจำหลักได้

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

การจัดสรรพื้นที่ในหน่วยความจำแบ่งออกเป็น 2 รูปแบบ ดังนี้ (ต่อ)

2.2 การแบ่งแบบพลวัต (Dynamic Partition)

เป็นการแบ่งพื้นที่ตามขนาดของโปรเซส โดยการใช้พื้นที่ว่างทางกายภาพที่เรียกว่า โฮล (Hold) เมื่อมีโปรเซสต้องการใช้งานพื้นที่ในหน่วยความจำ ระบบปฏิบัติการจำมีหน้าที่ในค้นหาพื้นที่โฮล (Hold) ที่มีขนาดใหญ่เพียงพอกับโปรเซส แล้วจึงนำโปรเซสนั้นเข้าไปใช้งานหน่วยความจำและจะทำการเก็บข้อมูลของโปรเซสอื่นที่ยังไม่ได้นำเข้าไปใช้งานหน่วยความจำและพื้นที่ว่าง (Free Partition Hold) ที่ยังไม่ได้ถูกจัดสรรให้กับโปรเซสใดๆ ด้วยแถวคอยการทำงาน (Job Queue) โดยสัมพันธ์กับเวลาที่ใช้งานจริง (Use time) ของแต่โปรเซสด้วย แสดงดังรูปที่ 2.8

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

แถวคอยงาน (Job Queue)

กระบวนการ Process	หน่วยความจำ (Memory)	เวลาที่ใช้ (Use Time)
$Process_1$	600K	10
$Process_2$	1000K	5
$Process_3$	300K	20
$Process_4$	700K	8
$Process_5$	500K	15

รูปที่ 2.8 แสดงการแบ่งหน่วยความจำแบบพลวัต (Dynamic Partition)

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

การป้องกันหน่วยความจำ (Memory Protection)

โดยปกติจำนวนบิต (Bits) ที่เก็บอยู่ในตารางเพจ (Page Table) เราสามารถกำหนดบิตเพื่อใช้การตรวจสอบและกำหนดเพจในการ อ่าน-เขียน (Read-Write) หรืออ่านข้อมูลเท่านั้น (Read-Only) ซึ่งเรียกบิตพิเศษนี้ว่า “กลุ่มบิตป้องกัน (Associating Protection Bits)” ให้กับทุกๆ เฟรม ที่อยู่ในหน่วยความจำ (Main Memory) ซึ่งแบ่งบิตสถานะออกเป็น 2 บิต คือ

1. บิตใช้งานได้ (Valid Bit) เป็นบิตสถานะที่บอกว่าข้อมูลในเพจถูกอ่านเข้าสู่หน่วยความจำทางกายภาพแล้ว และสามารถนำไปใช้งานได้ทันที

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

การป้องกันหน่วยความจำ (Memory Protection) (ต่อ)

2. บิตใช้งานไม่ได้ (Invalid Bit) เป็นบิตสถานะที่บอกว่าข้อมูลในเพจนั้นไม่มีอยู่ในหน่วยความจำทางกายภาพแล้ว (Physical Memory) และไม่สามารถนำไปใช้งานได้ อาจเกิดจากกรณีที่ระบบปฏิบัติการยังไม่ได้อ่านข้อมูลจากเพจเข้าสู่หน่วยความจำหลัก (Main Memory) หรือข้อมูลของเพจที่ต้องการอ่านนั้นถูกสลับ (Swapped) ออกจากหน่วยความจำแล้ว

ซึ่งจะทำให้เกิดข้อผิดพลาดขึ้นที่เรียกว่า “การผิดหน้า (Page Fault)” จึงจำเป็นต้องโหลดเพจข้อมูลเข้าสู่หน่วยความจำก่อนแล้วจึงเปลี่ยนค่าของบิตใช้งานไม่ได้ (Invalid Bit) ให้เป็นบิตใช้งานได้ (Valid Bit) แล้วจึงทำการประมวลผลข้อมูลนั้นใหม่อีกครั้งหนึ่ง

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

สรุป (Conclusion)

การจัดการหน่วยความจำ (Memory Management) เป็นหน้าที่หนึ่งของระบบปฏิบัติการ ซึ่งมีหน้าที่หลักในการจัดสรรพื้นที่ในการใช้งานให้กับหลายโปรแกรม (Multiprogramming) หรือข้อมูลต่างๆ โดยใช้ฮาร์ดแวร์ (Hardware provided) เข้าไปช่วยสนับสนุนวิธีการเข้าถึงตำแหน่งในหน่วยความจำทั้งในส่วนหน่วยความจำทางตรรกะ (Logical memory) และหน่วยความจำทางกายภาพ (Physical memory) ซึ่งมีหน่วยประมวลผลกลาง (CPU) เป็นตัวประมวลผล (Generated)

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

สรุป (Conclusion) (ต่อ)

Memory Management Algorithms มี 4 แบบ

1. การจัดสรรแบบต่อเนื่อง (Contiguous allocations)
2. การแบ่งหน้า (Paging)
3. การแบ่งส่วน (Segmentation)
4. การผสมผสานระหว่างวิธีแบ่งหน้าและแบ่งส่วน (Combination of Paging and Segmentation

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

สรุป (Conclusion) (ต่อ)

Memory Management Strategy

1.ความสามารถของฮาร์แวร์ (Hardware Support) เป็นพื้นฐานที่ชาวยสนับสนุนการทำงานของกระบวนการการแบ่งหน้า (Paging) และการแบ่งส่วน (Segmentation) โดยโครงสร้างตารางการจับคู่ระหว่างข้อมูลและตำแหน่งของหน่วยความจำหลัก

2. ประสิทธิภาพ (Performance) วิธีที่ใช้จัดการหน่วยความจำหลักแต่ละวิธีจะมีความซับซ้อน และใช้ระยะเวลาในการจับคู่ระหว่างหน่วยความจำทางตรรกะ (Logical memory) และหน่วยความจำทางกายภาพ (Physical memory) ดังนั้น ระบบปฏิบัติการจึงต้องเลือกใช้วิธีที่เหมาะสมและรวดเร็ว

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

สรุป (Conclusion) (ต่อ)

3. การจัดการพื้นที่ว่าง (Fragmentation) ในระบบคอมพิวเตอร์แบบหลายโปรแกรม (Multiprogramming) มีความสามารถในการประมวลผลสูง จะต้องมีการจัดการพื้นที่ว่างที่เหมาะสมกับขนาดของ Process เพื่อป้องกันปัญหาที่เกิดขึ้น

4. การย้ายตำแหน่ง (Relocation) การนจัดการหน่วยความจำหลักที่ดีควร สามารถย้ายโปรเซส (Process) ในหน่วยความจำหลักได้อย่างอัตโนมัติเพื่อแก้ปัญหาการเกิดพื้นที่ว่าง (Fragmentation) ในหน่วยความจำหลัก

5. การสลับ (Swapping) กลไกการสลับ Process เข้า-ออก จากหน่วยความจำหลัก ควรให้มีการสลับโปรเซสให้น้อยที่สุดภายในหนึ่งหน่วยเวลา

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.

สรุป (Conclusion) (ต่อ)

6. การใช้งานร่วมกัน (Sharing) ระบบปฏิบัติการที่ดีควรสามารถจัดสรรโรปแกรมหรือข้อมูลต่างๆ ให้สามารถทำงานร่วมกันได้ เพื่อให้โปรเซส (Process) จำนวนมากสามารถทำงานร่วมกันได้

7. การป้องกัน (Protection) การป้องกันคำสั่งหรือข้อมูลต่างๆ โครงสร้างข้อกำหนดเงื่อนไขในการอ่านหรือเขียนข้อมูล โดยระบบต้องมีการตรวจสอบคำสั่งหรือข้อมูลต่างๆ ขณะกำลังประมวลผล พร้อมถึงแสดงข้อผิดพลาดได้

Reference : สุจิตรา อดุลย์เกษม,ทฤษฎีระบบปฏิบัติการ.กรุงเทพฯ:โปรวิชั่น,2552.