บทที่ 6
การจัดการกระบวนการ
(Process Management)

หน้าที่สำคัญประการหนึ่งของระบบปฏิบัติการ ก็คือการจัดสรรทรัพยากรของระบบการทำงานที่มีอยู่ทั้งภายในและภายนอก รวมทั้งอุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์ ได้อย่างมีประสิทธิภาพ สิ่งที่เกี่ยวข้องหลักก็คือ กระบวนการ (Process) ซึ่งเป็นขั้นตอนหรือวิธีการทำงานตามคำสั่งต่างๆ ที่สัมพันธ์กันกับการทำงานของระบบคอมพิวเตอร์ ดังนั้นระบบปฏิบัติการจะต้องจะต้องควบคุมกระบวนการ (Process) ต่างๆ ที่เกี่ยวข้องให้การทำงานอย่างเป็นระบบและมีลำดับการทำงานที่ชัดเจนตรงตามวัตถุประสงค์ที่ต้องการ โดยทุก ๆ กระบวนการจะให้ความสำคัญในการประมวลผล ณ เวลาปัจจุบันกับหน่วยประมวลผลกลาง (CPU) เป็นหลัก

ในบทนี้จะกล่าวถึงวิธีการจัดการกับกระบวนการ (Process) การทำงานของกระบวนการ การจัดตาราง การสลับและการติดต่อสื่อสารระหว่างกระบวนการ ภายใต้การควบคุมและสั่งการโดยระบบปฏิบัติการ (ศัพท์บัญญัติ ราชบัณฑิตยสถาน, 2544)

6.1 แนวคิดเกี่ยวกับกระบวนการ (Process Concept)

การทำงานของระบบคอมพิวเตอร์ในอดีต จะมีวิธีการจัดการกับกระบวนการ (Process) ในรูปแบบกระประมวลผลทีละคำสั่งหรือทีละกระบวนการ (Single Process) ทำการใช้งานทรัพยากรที่มีอยู่ได้ไม่เต็มประสิทธิภาพ เนื่องจากระบบจะจัดสรรทรัพยากรทั้งหมดไว้สำหรับงานที่กำลังประมวลผลอยู่เท่านั้น จึงทำให้มีการพัฒนาเทคโนโลยีสารสนเทศสมัยใหม่ขึ้นมาเพื่อรองรับการทำงาน กรณีที่มีผู้ใช้ระบบคอมพิวเตอร์หลายคน (multi-user computer system) สามารถทำงานหลายกระบวนการไปพร้อมกันได้ ดังนั้นการทำงานของงานหนึ่ง อาจมีผลกระทบทางอ้อมกับอีกงานหนึ่ง โดยผ่านทางทรัพยากรที่ใช้ร่วมกัน เพื่อมิให้งานต่างๆ ส่งผลกระทบกันอันจะก่อให้เกิดความเสียหายต่อระบบ ระบบปฏิบัติการจะต้องสามารถควบคุมงานหรือสับหลีกการทำงานของแต่ละชุดคำสั่งที่มาเกี่ยวข้องหน้าที่นี้เรียกว่า การเข้าจังหวะกันของกระบวน (process synchronization) เพื่อให้การทำงานได้อย่างต่อเนื่อง และมีประสิทธิภาพ

6.1.1 เกี่ยวกับกระบวนการ (The Process) กระบวนการ (Process) ไม่ได้หมายถึงการทำงานหรือชุดคำสั่งที่กำลังประมวลผลในช่วงระยะเวลาใดเวลาหนึ่งเท่านั้น แต่

อาจจะรวมถึงส่วนของข้อความ (Text Section) ที่ประกอบไปด้วยค่าของชุดคำสั่งตัวนับ (Program Counter) และรายละเอียดของรีจิสเตอร์ที่เกี่ยวข้องกับหน่วยประมวลผลกลาง (CPU) ดังนั้นการทำงานของระบบคอมพิวเตอร์จำเป็นจะต้องมีการจัดการและจัดสรรเวลาการทำงานเป็นอย่างดี เพื่อให้การดำเนินงานต่างๆ เกี่ยวกับกระบวนการ (Process) ที่มีอยู่ทั้งหมดได้อย่างมีประสิทธิภาพ

6.1.2 สถานะของกระบวนการ (Process State) การดำเนินการของการประมวลผลของแต่ละกระบวนการ (Process) จะเกี่ยวข้องกับการเปลี่ยนสถานะ (State) ของการทำงานที่เกิดขึ้น 5 ขั้นตอน ดังนี้

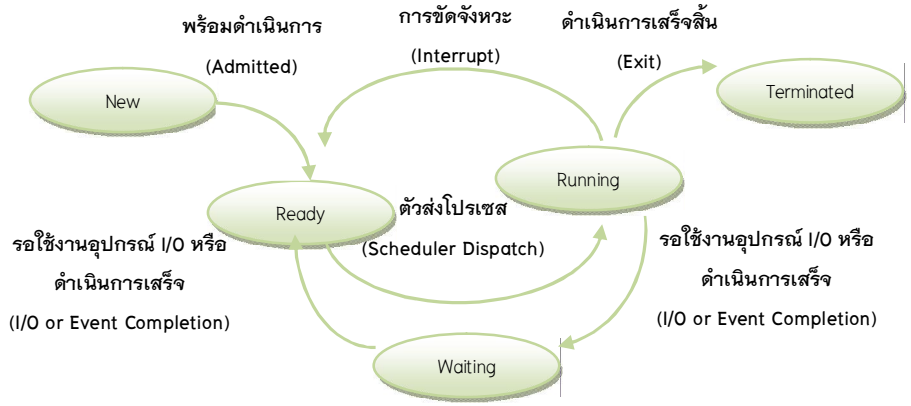
1. New เพื่อแสดงถึง การสร้างกระบวนการใหม่ (Created) ขึ้นมา

2. Running เพื่อแสดงถึง คำสั่งที่จะได้รับการประมวลผล (Executed)

3. Waiting เพื่อแสดงถึง กระบวนการ (Process) ที่รอคอย (Waiting) บางเหตุการณ์ที่กำลังจะเกิดขึ้น รอใช้งานอุปกรณ์ I/O หรือรอรับสัญญาณ (signal) บางอย่าง

4. Ready เพื่อแสดงถึง กระบวนการ (Process) ที่รอเพื่อที่จะใช้งานหน่วยประมวลผลกลาง (CPU)

5. Terminal เพื่อแสดงถึง กระบวนการ (Process) ที่ประมวลผลเสร็จสิ้นแล้วการแสดงการเปลี่ยนสถานะการทำงานของแต่ละกระบวนการ (Process) จะมีรูปแบบที่แตกต่างกันไปตามหน้าที่ (Function) หรือแสดงให้รู้ว่ามีกระบวนการ (Process) ที่กำลังดำเนินการอะไรอยู่บาง ณ เวลาในขณะนั้น แสดงได้ดังภาพที่ 6.1



ภาพที่ 6.1 แสดงแผนภาพขั้นตอนการทำงานของกระบวนการ
(Diagram of Process State)

6.1.3 บล็อกควบคุมกระบวนการ (Process Control Block: PCB) อยู่ภายในระบบปฏิบัติการมีหน้าที่เป็นตัวควบคุมการทำงานของแต่ละกระบวนการ (Process) บางครั้งอาจเรียกว่า **"บล็อกควบคุมการทำงาน (Task Control Block)"** แสดงได้ดังภาพที่ 6.2 ซึ่งประกอบไปด้วย

ภาพที่ 6.2 บล็อกควบคุมกระบวนการ (Process Control Block: PCB)

1. สถานะของกระบวนการ (Process State) อาจจะเป็นสถานะที่เกิดขึ้นใหม่ (New) ของกระบวนการสถานะพร้อม (Running) สถานะที่กำลังประมวลผล (Running) สถานะคอย (Waiting) และสถานะหยุดการทำงานชั่วคราว (Halted) เป็นต้น

2. ชุดคำสั่งนับ (Program Counter) จะใช้แสดงตำแหน่งของคำสั่งถัดไปของกระบวนการ (Process) ที่จะถูกประมวลผล

3. รีจิสเตอร์ที่เกี่ยวกับหน่วยประมวลผลกลาง (CPU Registers) ตัวเลขและชนิดของรีจิสเตอร์ทุกตัวจะขึ้นอยู่กับสถาปัตยกรรมคอมพิวเตอร์ที่เลือกใช้ นอกจากนี้ยังเกี่ยวข้องกับ ตัวสะสมค่า (Accumulator) รีจิสเตอร์ตัวชี้ (Index Register) สแต็ก (Stack) ตัวชี้ (Pointer) และ รีจีสเตอร์ทั่วไป (General-Purpose Registers) รวมไปถึงคำสั่งเพื่อกำหนดเงื่อนไขต่างๆ (Condition-Code Information) เป็นต้น โดยที่ชุดคำสั่งตัวชี้จะเก็บสถานะของกระบวนการ (Process) เมื่อเกิดมีการขัดจังหวะขึ้น (Interrupt) และสามารถกลับมาทำงานเดิมต่อได้ภายหลัง แสดงได้ดังภาพที่ 6.3

4. การจัดตารางการทำงานของหน่วยประมวลผล (CPU-Scheduling Information) ข้อมูลที่เก็บอยู่จะเกี่ยวของกับการจัดลำดับความสำคัญของกระบวนการ (Process Priority) ตัวชี้ไปยังที่แถวลำดับ (Queues) และค่าพารามิเตอร์ (Parameters) อื่นๆ ที่จัดเก็บอยู่ในตารางการทำงานของหน่วยประมวลผล

5. การจัดการข้อมูลภายในหน่วยความจำ (Memory-Management Information) ประกอบด้วยรีจิสเตอร์ที่ใช้เก็บค่าของตำแหน่งต่ำสุดของแต่ละกระบวนการ (Base Register) และเก็บค่าขนาดของแต่ละกระบวนการ (Limit Register) ลงบนตารางเพจ (Page Tables) หรือตารางเซ็กเมนต์ (Segment Tables) โดยจะขึ้นอยู่กับระบบการจัดการหน่วยความจำ (Memory System) ของระบบปฏิบัติการ (Operating System) ที่เลือกใช้

6. บัญชีข้อมูล (Account Information) เป็นข้อมูลเกี่ยวข้องกับจำนวนของหน่วยประมวลผลที่ใช้ (Amount CPU) เวลาที่ใช้งานจริง ๆ เวลาทั้งหมด (Limit Times) บัญชีตัวเลข (Account Numbers) งาน (Job) และจำนวนกระบวนการที่มีอยู่ (Process Numbers)

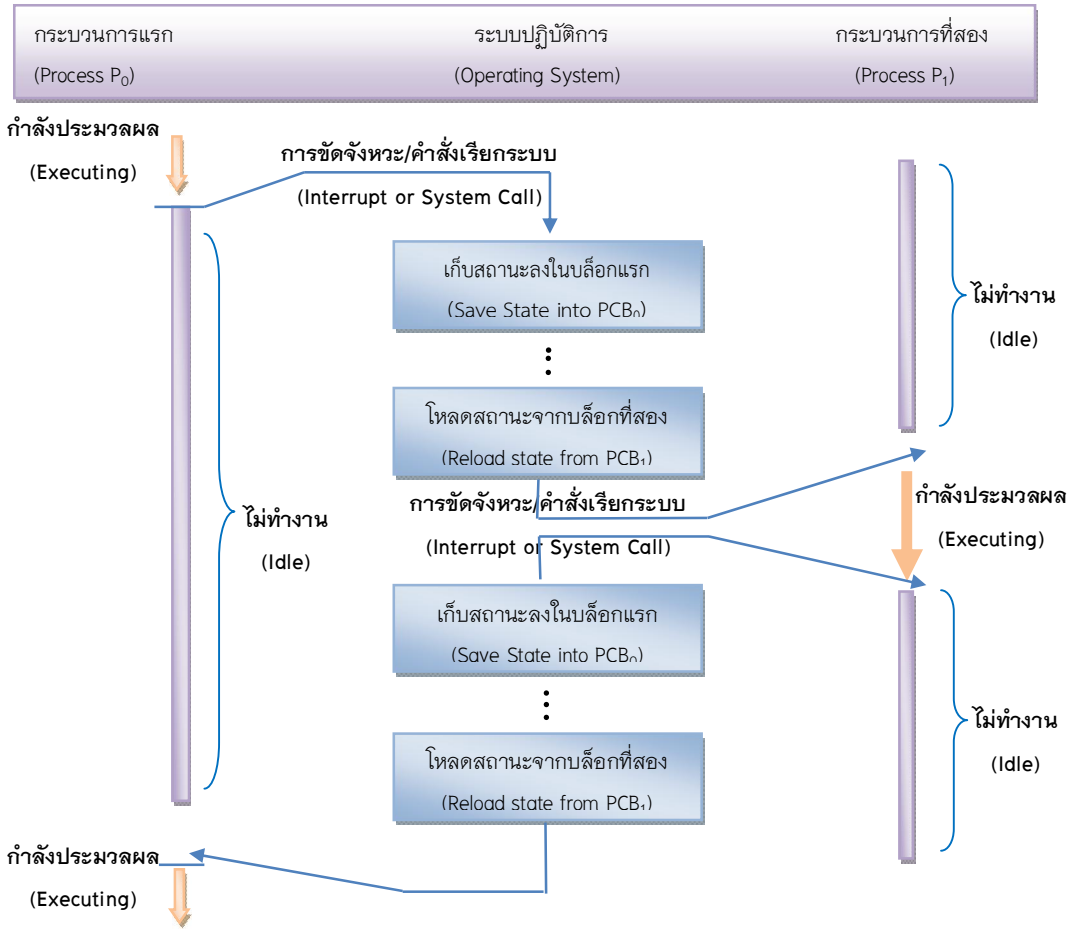
7. สถานะอุปกรณ์รับและแสดงผลข้อมูล (I/O Status Information) เป็นข้อมูลที่มีไว้สำหรับแสดงรายการการจัดสรร (Allocated) การใช้งานอุปกรณ์รับและแสดงผล (I/O Devices) หรืออาจจะเป็นการเปิดไฟล์ (Open Files) ของแต่ละกระบวนการ (Process)

6.1.4 หน่วยงานย่อยภายในกระบวนการ (Threads) เริ่มต้นของการพัฒนาระบบปฏิบัติการนั้น ภายในกระบวนการ (Process) จะประกอบไปด้วยหน่วยงานย่อยเพียงหน่วยงานเดียวเท่านั้น (Single Thread) ที่ได้รับการประมวลผล (Execution) ซึ่งในเวลาต่อมาได้มีการพัฒนาระบบปฏิบัติการให้กระบวนการหนึ่งสามารถที่จะมีหน่วยงานย่อยภายในกระบวนการ (Threads) จำนวนมากได้ โดยการเพิ่มประสิทธิภาพในกับหน่วยการประมวลผลให้สามารถ ที่จะประมวลผลหน่วยงานย่อยภายในกระบวนการ (Threads) พร้อมๆ กันและเป็นอิสระต่อกันได้ ทำให้ประสิทธิภาพการทำงานของกระบวนการ (Process) รวดเร็วขึ้น โดยกระบวนการ (Process) ที่ภายในประกอบด้วยหน่วยงานย่อย (Threads) ทำงานอยู่นี้ว่า **"Multithreading"**

6.2 การจัดตารางกระบวนการ (Process Scheduling)

จุดประสงค์หลักของระบบการทำงานต้องการประมวลผลแต่ละกระบวนการ (Process) ได้อย่างรวดเร็วและมีประสิทธิภาพและใช้งานหน่วยประมวลผลกลางได้อย่างเต็มที่ (CPU Utilization) ดังนั้นจำเป็นจะต้องอาศัยองค์ประกอบเพื่อช่วยในการจัดการกับกระบวนการ

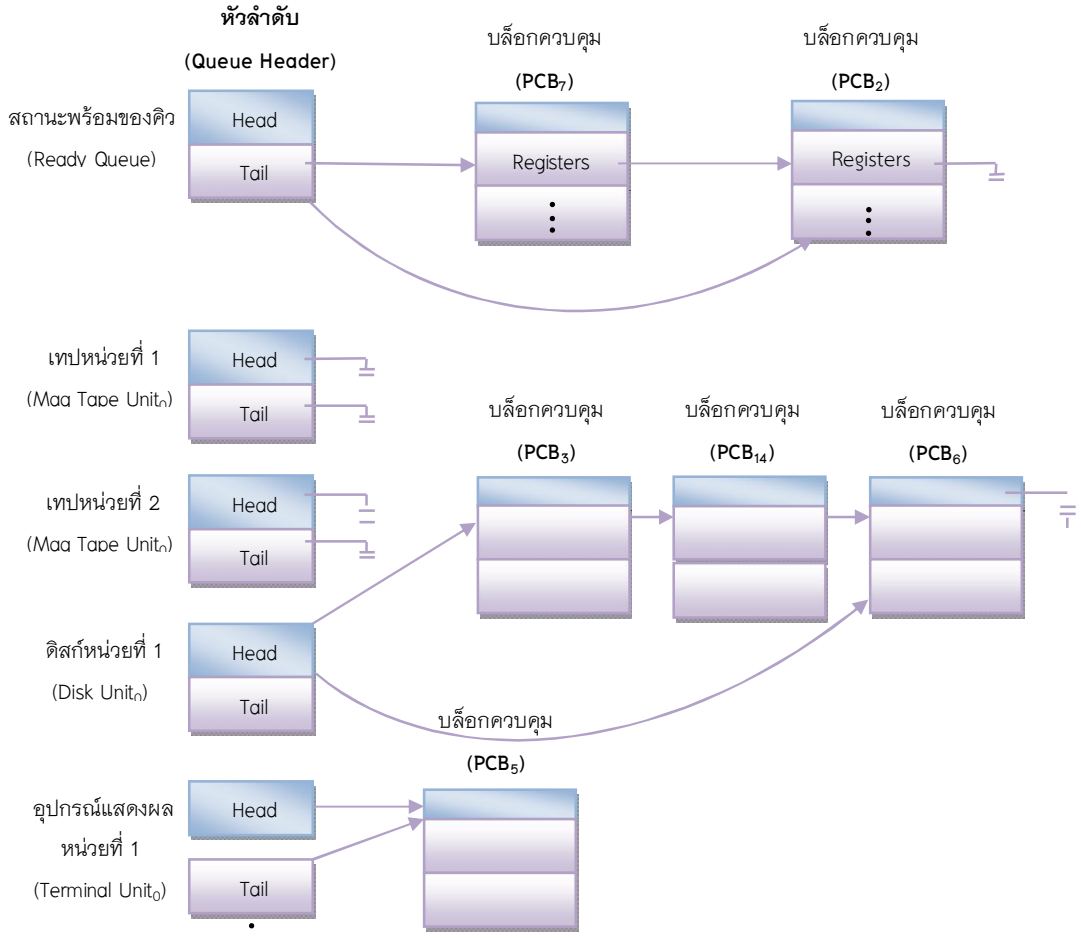
(Process) ที่มีการเปลี่ยนแปลงสถานะอยู่ตลอดเวลา ตลอดจนการจัดการความถี่ของกระบวนการ (Process) ในการสลับเข้าและออกเพื่อเข้าไปใช้งานหน่วยประมวลผลการ (CPU) โดยที่ผู้ใช้งานสามารถที่จะใช้งานชุดคำสั่งได้ขณะที่กระบวนการ (Process) นั้นๆ ถูกประมวลผลอยู่ ดังนั้นการจัดตารางการทำงานของกระบวนการ (Process) จะต้องประกอบไปด้วยสิ่งที่เกี่ยวข้องดังต่อไปนี้



ภาพที่ 6.3 แสดงแผนภาพการทำงานของหน่วยประมวลผลกลางในการเปลี่ยนสถานะระหว่าง 2 กระบวนการ (Diagram Showing CPU Switch from Process to Process)

6.2.1 การจัดตารางแถวลำดับ (Scheduling Queues) เมื่อกระบวนการ (Process) ถูกนำเข้าระบบ จะถูกใส่ลงไปในแถวลำดับการทำงาน (Job Queue) โดยที่กระบวนการ (Process) ที่มีอยู่ภายในหน่วยความจำหลัก (Main Memory) ที่อยู่ในสถานะพร้อม (Ready) และสถานะคอย (Wait) พร้อมจะถูกประมวลผลจะถูกจัดเก็บลงบนรายการ (List)

เรียกสถานะนี้ว่า สถานะพร้อมของแถวลำดับ (Ready Queue) โดยที่สถานะพร้อมของแถวลำดับหัว (Ready Queue Header) ภายในรายการ (Lists) จะมีตัวชี้ (Pointers) เพื่อชี้ไปยังตำแหน่งแรกและตำแหน่งสุดท้ายของบล็อกควบคุมกระบวนการ (Process Control Block: PCB) อีกทั้งยังสามารถที่จะเพิ่มจำนวนบล็อกควบคุมกระบวนการ (PCB) และมีขอบเขตของตัวชี้ (Pointer Field) เพื่อชี้ไปยังบล็อกควบคุมกระบวนการ (PCB) ตัวถัดไปได้ ซึ่งระบบโดยทั่วไปที่ประกอบด้วยหลายๆ กระบวนการ พื้นที่บนจานบันทึก (disk)อาจไม่ว่างเมื่อมีการร้องขอการใช้งานอุปกรณ์ I/O จากบางกระบวนการ ดังนั้นกระบวนการจำเป็นต้องค่อยการใช้พื้นที่บนจานบันทึก (disk) รายการ (List) ของกระบวนการที่คอยใช้งานอุปกรณ์ I/O เรียกว่า แถวลำดับการใช้งานอุปกรณ์ ซึ่งในโดยแต่ละกระบวนการจะมีลำดับการใช้งานอุปกรณ์เป็นของตัวเอง แสดงได้ดังภาพที่ 6.4



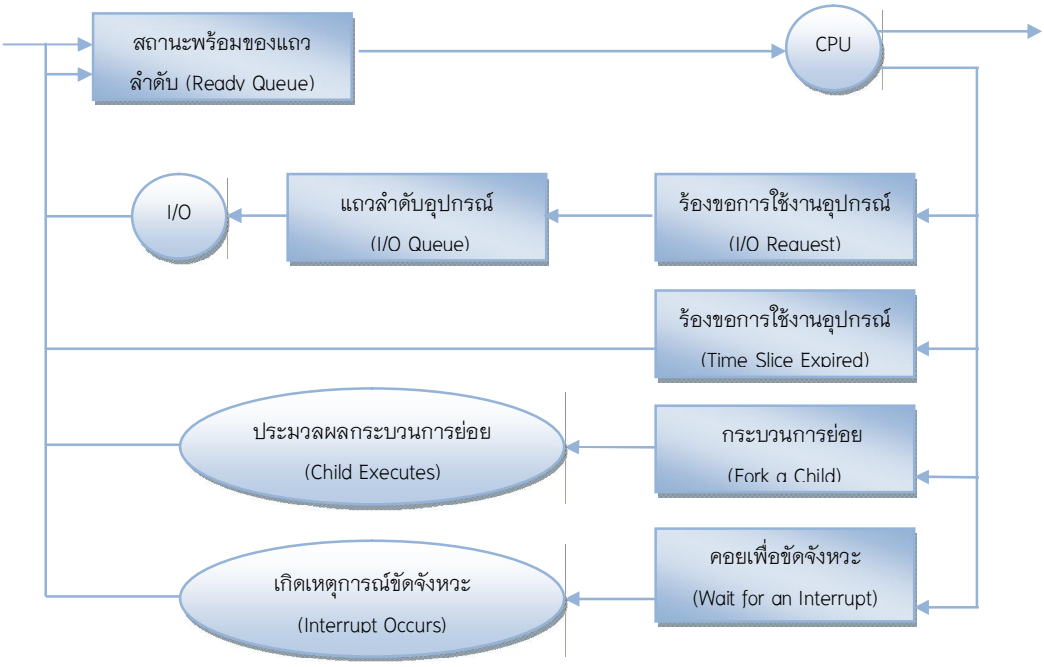
ภาพที่ 6.4 แสดงสถานะพร้อมของคิวและลำดับการใช้งานอุปกรณ์รับและแสดงผล
(The Ready Queue and Various I/O Device Queues)

โดยปกติการแสดงการจัดตารางลำดับของกระบวนการจะเกี่ยวข้องกับแผนภาพแสดงลำดับการทำงาน (Queuing Diagram) แสดงได้ดังภาพที่ 6.5 ซึ่งกรอบสี่เหลี่ยมแสดงแถวลำดับการทำงาน (Queue) และรูปวงรีแสดงทรัพยากรที่ค่อยให้บริการแถวลำดับการทำงาน โดยประกอบด้วยแถวลำดับที่แสดงสถานะพร้อม (Ready Queue) และกลุ่มของอุปกรณ์ที่แถวลำดับเรียกใช้ (Device Queue) และลูกศรแสดงทิศทางการไหลของกระบวนการทำงานในระบบเริ่มต้นโดยที่กระบวนการใหม่ (New Process) ถูกใส่ลงในสถานะพร้อมของแถวลำดับ (Ready Queue) และคอยจนกระทั้งถูกเลือกเพื่อเข้าเข้าไปประมวลผล (โดยตัวจัดส่ง Dispatcher) ในหน่วยประมวลผลกลาง (CPU) โดยระหว่างการดำเนินการอยู่นั้นอาจมีเหตุการณ์ใดเหตุการณ์หนึ่งสามารถที่จะเกิดขึ้นได้ดังนี้

1. กระบวนการ (Process) ร้องขอการใช้งานอุปกรณ์ I/O ถูกจัดให้รอในแถวลำดับ

2. กระบวนการ (Process) สร้างกระบวนการย่อยและคอยจนกว่ากระบวนการย่อยจะดำเนินการแล้วเสร็จ

3. กระบวนการ (Process) สามารถที่ถูกย้ายออกจากหน่วยประมวลผลกลาง (CPU) เนื่องจากมีการขัดจังหวะ (Interrupt) และเปลี่ยนสถานะเป็นสถานะพร้อมของแถวลำดับ (Ready Queue)



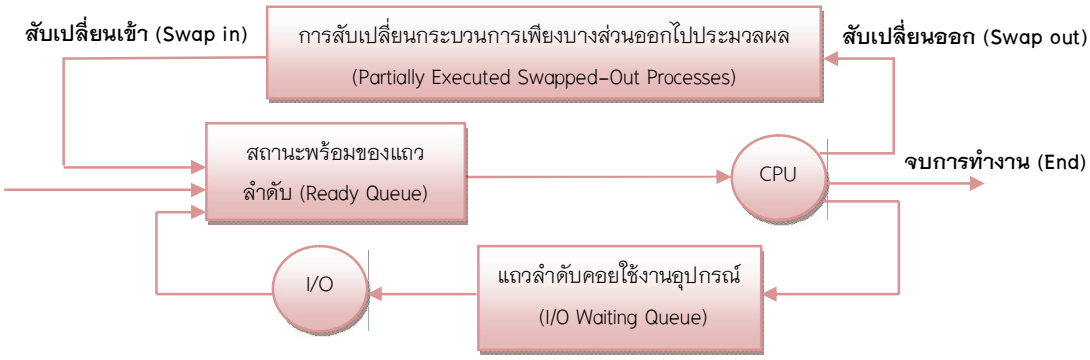
ภาพที่ 6.5 แสดงแผนภาพการทำงานของแถวลำดับของการจัดตารางกระบวนการ (Queuing–Diagram Representation of Process Scheduling)

6.2.2 การจัดตารางการทำงาน (Scheduler) เป็นวิธีการจัดการกระบวนการ (Process) ที่เหมาะสมโดยนำเข้าหรือออกจากตารางแถวลำดับ (Scheduling Queues) ซึ่งเป็นหน้าที่ของระบบปฏิบัติการ ในระบบงานแบบกลุ่มการจัดการกระบวนการมักจะนำไปพักไว้ชั่วคราว (Spooled) ณ อุปกรณ์หน่วยความจำสำรองที่มีความจุขนาดใหญ่ (Mass-Storage Device) เช่น จานบันทึก (disk) เป็นต้น เพื่อที่จะนำไปประมวลผลในเวลาต่อมาซึ่งมีวิธีการจัดตารางการทำงานหลายวิธี ดังนี้

1. การจัดตารางการทำงานระยะยาว (Long-term scheduler) หรือเรียกวิธีนี้ว่า การจัดตารางงาน(Job Scheduler) วิธีนี้การทำงานจะทำการเลือกกระบวนการ (Processes) ที่เก็บพักไว้ชั่วคราว (Spool) แล้วทำการบรรจุ (Load) ข้อมูลเข้ามาในหน่วยความจำเพื่อทำการประมวลผลความถี่ในการประมวลผลจะถกเรียกใช้ไม่บ่อยมากนักระหว่างการสร้างกระบวนการขึ้น ในระบบ นอกจากนี้การจัดตารางการทำงานระยะยาวจะทำการควบคุมระดับของการทำงานหลายๆ ชุดคำสั่งพร้อมในระหว่างการประมวลผล

2. การจัดตารางการทำงานระยะกลาง (Medium-term scheduler) โดยแต่ละระบบปฏิบัติการจะมีวิธีการจัดตารางการทำงานที่ต่างกัน เช่น การแบ่งปันเวลาในการทำงาน (Time-Sharing Systems) ซึ่งกระบวนการ (Process) ที่เข้ามาใช้งานในหน่วยความจำมีการสับเปลี่ยนเข้าและออก (Swapped in/Swapped Out) เป็นต้น แสดงได้ดังภาพที่ 6.6

3. การจัดตารางการทำงานระยะสั้น (Short-term scheduler) หรือเรียกวิธีนี้ว่าการจัดการหน่วยประมวลผลกลาง (CPU Scheduler) วิธีนี้การทำงานจะทำการเลือกกระบวนการ (Processes) ที่อยู่ในสถานะพร้อม (Ready) เพื่อนำมาประมวลผล (Execute) และจัดสรรเนื้อที่การใช้งานประมวลผลกลางกลาง (CPU) ให้กับกระบวนการนั้นที่เลือกมา



ภาพที่ 6.6 แสดงแผนภาพการเพิ่มวิธีการจัดตารางการทำงานระยะกลาง
(Addition of Medium-Term Scheduling to The Queuing Diagram)

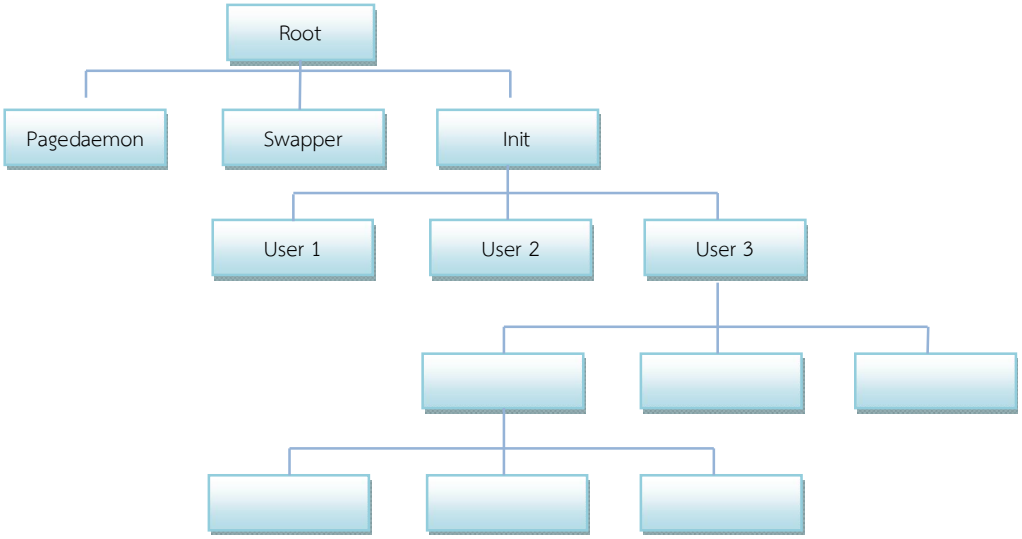
6.2.3 การสลับกระบวนการทำงาน (Context Switch) เป็นวิธีการสลับ (Switching) ให้กระบวนการ (Process) อื่นเข้ามาใช้งานหน่วยประมวลผลกลาง (CPU) เพื่อจัดเก็บค่าสถานะเดิมของกระบวนการ (Old Process) ก่อนหน้า ขณะเดียวกันก็จะทำการบรรจุและจัดเก็บค่าสถานะให้กับกระบวนการใหม่ (New Process) ลักษณะการดำเนินงานดังกล่าวเรียกว่า การสลับกระบวนการทำงาน (Context Switch) โดยการสลับกระบวนการทำงาน จะอยู่ภายในบล็อกควบคุมกระบวนการ (PCB) ซึ่งจัดเก็บค่ารีจิสเตอร์ของหน่วยประมวลผลกลางไว้ (CPU Register) โดยสถานะของกระบวนการ (Process State) แสดงได้ดังภาพที่ 6.1

6.3 การดำเนินการกับกระบวนการ (Operations on Processes)

กระบวนการ (Process) ต่างๆ ที่อยู่ในระบบสามารถประมวลพร้อมกันได้ภายในเวลาเดียวกัน ขณะเดียวกันการสร้าง (Created) และลบ (Deleted) กระบวนการก็เกิดขึ้นได้ตลอดเวลาเช่นเดียวกัน ดังนั้นระบบปฏิบัติการจะต้องมีกลไกหรือความสามารถในการสร้างกระบวนการ (Process Creation) ขึ้นมาและเมื่อกระบวนการดำเนินการเสร็จสิ้น (Process Termination) ก็สามารถที่จะลบกระบวนการนั้นทิ้งไปได้ ซึ่งสามารถอธิบายการดำเนินกับกระบวนการต่างๆ ได้ดังนี้

6.3.1 การสร้างกระบวนการ (Process Creation) กระบวนการ (Process) อาจจะสร้างขึ้นมาใหม่โดยผ่านทางคำสั่งระบบ (System Call) โดยเรียกกระบวนการที่ถูกสร้างขึ้นมาใหม่นี้ว่า กระบวนการพ่อหรือแม่ (Parent Process) และเรียกกระบวนการที่ถูกสร้างขึ้นภายหลังว่า กระบวนการลูก (Children Process) นอกจากนี้แต่ละกระบวนการก็สามารถที่จะสร้างกระบวนใหม่ตามลำดับในรูปแบบต้นไม้ แสดงได้ดังภาพที่ 6.7

โดยทั่วไปแล้ว กระบวนการจะต้องการใช้ทรัพยากร เช่น เวลาใช้งานหน่วยประมวลผลกลาง (CPU Time) หน่วยความจำ (Memory) ไฟล์ (File) อุปกรณ์รับและแสดงผล (I/O Device) เป็นต้น เพื่อจะให้งานที่ดำเนินการเสร็จสมบูรณ์ เมื่อกระบวนการได้สร้างกระบวนการย่อยขึ้นมา กระบวนการย่อย (Subprocess) อาจจะใช้งานทรัพยากรได้โดยตรงจากระบบปฏิบัติการหรือใช้งานผ่านกลุ่มงานย่อยของกระบวนการพ่อหรือแม่ก็ได้ (Parent) และเมื่อหลังจากประมวลผลกระบวนการ (Process) เสร็จสิ้นแล้วจะต้องคืนทรัพยากรที่ใช้ในการทำงานกลับคืนสู่ระบบ และกระบวนการก็จะลบหรือทำลายทิ้งตามกลไกการทำงานของระบบปฏิบัติการ



ภาพที่ 6.7 แสดงโครงสร้างต้นไม้ในการสร้างกระบวนการใหม่ของระบบปฏิบัติการ UNIX (A Tree of Processes on a typical UNIX System)

6.3.2 การสิ้นสุดกระบวนการ (Process Termination) จะเกิดขึ้นเมื่อคำสั่งสุดท้ายถูกประมวลผลจนเสร็จและส่งคำถามไปยังระบบปฏิบัติการว่าจะทำการลบกระบวนการนี้เพื่อออกจากระบบโดยเรียกใช้คำสั่งระบบ (System Call) ณ จุดนี้กระบวนการ (Process) อาจจะส่งข้อมูลออกไปยังกระบวนการพ่อหรือแม่ (Parent) ก็ได้ ดังนั้นกระบวนการพ่อหรือแม่ยังสามารถทำงานหรือสร้างกระบวนการลูกขึ้นมาใหม่ได้ตามต้องการ กระบวนการพ่อหรือแม่ (Parent) สามารถที่จะยกเลิกการทำงานของกระบวนการลูก (Children) ได้หลายเหตุผล ดังนี้

1. กระบวนการลูก (Children) ใช้บางทรัพยากรที่มีอยู่ในระบบมากเกินไป ดังนั้นกระบวนการพ่อหรือแม่ (Parent) จำเป็นต้องมีกลไกในการตรวจสอบสถานะของกระบวนการลูก (Children) ด้วย

2. งานที่มอบหมายให้กับกระบวนการลูก (Children) ไม่มีความจำเป็น

3. กระบวนการพ่อหรือแม่ (Parent) ประมวลผลเสร็จแล้วและระบบปฏิบัติการมาอนุญาตให้กระบวนการลูก (Children) ไม่สามารถทำงานต่อได้ถ้ากระบวนการพ่อหรือแม่ (Parent) สิ้นสุดการทำงานแล้วทุกกระบวนการลูก (Children) ต้องสิ้นสุดการทำงานเหมือนกัน ปรากฏการณ์ลักษณะนี้มักเรียกว่า การสิ้นสุดการเชื่อมโยง (Cascading Termination) ถูกจัดการโดยระบบปฏิบัติการ

6.4 การร่วมกันของกระบวนการ (Cooperating Process)

การประมวลผลกระบวนการ (Process) พร้อมกันในระบบปฏิบัติการ แต่ละกระบวนการอาจจะเป็นอิสระจากกัน (Independent) โดยเมื่อกระบวนการหนึ่งกำลังประมวลผลอยู่จะไม่มีส่งผลกระทบ (Effect) กับอีกกระบวนการหนึ่งหรือการร่วมกันของกระบวนการ (Cooperating) ถ้ากระบวนการหนึ่งกำลังประมวลผลอยู่จะมีผลกระทบ (Effect) กับอีกกระบวนการหนึ่ง เพราะมีการใช้ข้อมูลบางตัวรวมกันอยู่ โดยผู้ใช้งานสามารถที่จะกำหนดสภาพแวดล้อมที่จะอนุญาตให้มีการร่วมกันของกระบวนการด้วยเหตุผลทั่วไปต่อไปนี้

1. การอนุญาตให้ใช้ข้อมูลร่วมกัน (Information Sharing) เช่น การใช้แฟ้มข้อมูลร่วมกัน (Shared File) เป็นต้น

2. การเพิ่มความเร็วในการคำนวณ (Computation Speedup) กรณีที่ต้องการให้งานที่ประมวลผลเร็วขึ้น จำเป็นต้องแบ่งงานออกเป็นงานย่อย (Subtask) โดยแต่ละงานย่อย (Subtask) จะประมวลผล ในรูปแบบคู่ขนาน (Parallel) ดังนั้นความเร็วจะเพิ่มขึ้นได้ถ้าคอมพิวเตอร์มีตัวประมวลผลหลายตัว (Multiple Processing Elements) เช่น หน่วยประมวลผลกลาง (CPUs) หรือช่องของการรับและแสดงผล (I/O Channels) เป็นต้น

3. การแบ่งเป็นโมดูล (Modularity) เป็นการแบ่งกระบวนการ (Process) ในแต่ละกระบวนการออกเป็นโมดูลย่อย ๆ

4. ความสะดวกสบาย (Convenience) หมายถึงผู้ใช้แต่ละอาจจะทำงานได้หลายๆ งานในเวลาหนึ่ง เช่น เพิ่มข้อมูล (Editing) การพิมพ์ (Printing) และการแปลชุดคำสั่งในระบบคู่ขนาน (Compiling in Parallel) เป็นต้น

6.5 การติดต่อสื่อสารระหว่างกระบวนการ (Interprocess Communication)

ในระบบคอมพิวเตอร์โดยส่วนใหญ่สามารถที่จะประมวลผลได้หลายๆ ชุดคำสั่งในเวลาเดียวกัน (Multiprogramming) มีการเชื่อมโยงและติดต่อสื่อสารกันในระบบเครือข่าย ร่วมทั้งมีการแบ่งปันทรัพยากรและใช้งานกระบวนการ (Process) ร่วมกันได้ ซึ่งบางครั้งอาจเรียกวิธีการติดต่อสื่อสารข้อมูลแบบนี้ว่า การติดต่อสื่อสารระหว่างกระบวนการ (Interprocess Communication: IPC) โดยที่ระบบจะมีกลไกการทำงานโดยจะอนุญาตให้แต่ละกระบวนการ (Processes) ติดต่อและส่งข้อมูลระหว่างกันได้อย่างต่อเนื่อง (Synchronize) โดยไม่อนุญาตให้ใช้พื้นที่ว่างตำแหน่งเดียวกัน ซึ่งการติดต่อสื่อสารระหว่างกระบวนการ สามารถแบ่งออกได้ดังนี้

6.5.1 ระบบการส่งข้อความ (Message–Passing System) ซึ่งหน้าที่ของระบบการส่งข้อความจะยอมให้กระบวนการ (Processes) ติดต่อสื่อสารกับกระบวนการ (Processes) อื่นโดยไม่ต้องการที่จะใช้ทรัพยากรหรือข้อมูลร่วมกัน เพียงต้องการที่จะส่งข้อความด้วยวิธีการติดต่อสื่อสารกับไมโครคอร์เนล (Microkernels) เท่านั้น ปกติการบริการแบบนี้จะคำนึงถึงผู้ใช้กระบวนการ (Processes) เป็นหลัก โดยการดำเนินการจะอยู่ภายนอกคอร์เนล ซึ่งการสื่อสารของผู้ใช้กระบวนการ (Processes) ทั้งหมดจะเกี่ยวข้องกับการส่งผ่านข้อความเท่านั้น และการติดต่อสื่อสารระหว่างกระบวนการ (IPC) โดยจะใช้ตัวดำเนินการที่เกี่ยวข้อง 2 ตัว คือ ตัวส่งข้อความ (send (message)) และตัวรับข้อความ (receive (message)) โดยมีตัวเชื่อมในการติดต่อสื่อสาร (Communication Link) เพื่อส่งข้อความระหว่าง 2 กระบวนการ (Processes)

วิธีการทางกายภาพที่สามารถนำไปใช้จัดการกับการเชื่อมโยงระหว่างตัวดำเนินการส่งและตัวดำเนินการรับข้อความ (Send/Receive Operations) ทำได้ดังนี้

1. วิธีการสื่อสารทางตรงหรือทางอ้อม (Direct or Indirect Communication)

2. วิธีการสื่อสารแบบสมมาตรหรือไม่สมมาตร (Symmetric or Asymmetric Communication)

3. วิธีการพักข้อมูลที่แน่นอนหรืออัตโนมัติ (Automatic or Explicit Buffering)

4. วิธีการส่งแล้วทำสำเนาหรือส่งแล้วทำการอ้างอิง (Send by Copy or Send by Reference)

5. วิธีการระบุขนาดของข้อความที่แน่นอนหรือขนาดของข้อความที่ไม่แน่นอน (Fixed–Sized or Variable–Sized Massage)

6.5.2 การระบุช่องทางสื่อสาร (Naming) โดยกระบวนการ (Process) ที่ต้องการติดต่อสื่อสารระหว่างกันต้องมีการอ้างถึงทิศทางในการสื่อสาร เช่น การสื่อสารแบบทางตรง (Direct Communication) หรือการสื่อสารแบบทางอ้อม (Indirect Communication) เป็นต้น

6.5.2.1 การสื่อสารแบบทางตรง (Direct Communication) เป็นวิธีการติดต่อสื่อสารโดยตรงระหว่างกระบวนการ (Process) โดยระบุชื่อผู้ส่งต้นทางและผู้รับปลายทางให้ถูกต้องโดยใช้ของคำสั่งที่ติดต่อสื่อสารระหว่างกันดังนี้ การระบุตำแหน่งแบบสมมาตร (Symmetry)

(P, message) หมายถึง ส่งข้อความข่าวสารไปยังกระบวนการ (Process) P

Receive (Q, message) หมายถึง วิธีการรับข้อความข่าวสารจากกระบวนการ (Process) Q โดยการเชื่อมโยงการติดต่อสื่อสารประกอบด้วยคุณสมบัติดังต่อไปนี้

1. สร้างการเชื่อมโยงเพื่อติดต่อสื่อสารระหว่างกระบวนการ (Process) ทุกคู่ (ผู้ส่งและผู้รับ) อย่างอัตโนมัติและแต่ละกระบวนการ (Process) จะมีความสัมพันธ์เมื่อมีการติดต่อสื่อสารและเชื่อมโยงกันเท่านั้น

2. การเชื่อมโยงของกลุ่มที่แน่นอนระหว่างสองกระบวนการ (Process)

3. ระหว่างกระบวนการ (Process) ทุกคู่ (ผู้ส่งและผู้รับ) จะมีเส้นทางการเชื่อมโยงภายเพียงหนึ่งเส้นทางเท่านั้น

การระบุตำแหน่งแบบไม่สมมาตร (Asymmetry)

Send (P, message) หมายถึง ส่งข้อความข่าวสารไปยังกระบวนการ (Process)

Receive (id, message) หมายถึง รับข้อความข่าวสารจากกระบวนการ (Process) ใด ๆ โดยระบุตัวแปร id เพื่อบอกกลุ่มชื่อของกระบวนการ (Process) เพื่อใช้ในการ

6.5.2.2 การสื่อสารแบบทางอ้อม (Indirect Communication) เป็นวิธีการติดต่อสื่อสารระหว่างกระบวนการ (Process) ข้อมูลข่าวสารจากผู้ส่งไปถึงผู้รับผ่านตัวกลางที่เรียกว่า กล่องข้อความ (Mail Box) หรือช่องทางการสื่อสาร (Port) โดยกล่องข้อความจะใช้จัดเก็บข้อความของกระบวนการ (Process) เพื่อจัดส่งไปยังกระบวนการ (Process) อื่นต่อไป และแต่ละกล่องข้อความจะมีหมายเลขที่ระบุไว้ไม่ซ้ำกัน (Unique) โดยการติดต่อสื่อสานระหว่างกระบวนการไปยังกระบวนการ (Process) อื่นๆ ผ่านทางหมายเลขกล่องข้อความที่แตกต่างกันซึ่งระหว่างกระบวนการ (Process) สามารถที่จะใช้กล่องข้อความร่วมกันได้ โดยการส่งและการรับสามารถกำหนดได้ดังนี้

send (A, message) หมายถึง วิธีการส่งข้อความข่าวสารไปยังกล่องข้อความ (Mail Box) A

Receive (A, message) หมายถึง วิธีการรับข้อความข่าวสารจากกล่องข้อความ (Mail Box) A โดยการเชื่อมโยงการติดต่อสื่อสารประกอบด้วยคุณสมบัติดังต่อไปนี้

1. สร้างการเชื่อมโยงเพื่อติดต่อสื่อสารระหว่างกระบวนการ (Process) ทุกๆ คู่ (ผู้ส่งและผู้รับ) เฉพาะคู่ที่มีการใช้ข้อมูลในกล่องข้อความ (Mail Box) ร่วมกันเท่านั้น

2. การเชื่อมโยงของกลุ่มซึ่งมีการเชื่อมโยงไปยังกระบวนการอื่นๆ มากกว่าสองกระบวนการ (Process)

3. การเชื่อมโยงระหว่างกระบวนการ (Process) อาจทำได้หลายเส้นทาง แต่จะมีเพียงหนึ่งเส้นทางที่เชื่อมโยงไปยังกล่องข้อความ (Mail Box) เดียวเท่านั้น

6.5.3 การส่งข้อความประสานกันหรือไม่ประสานกัน (Synchronization or Asynchronization) เป็นวิธีการส่งข้อมูลในรูปแบบของบล็อกข้อความ (Blocking) ที่ต้องประสานการส่งหรือการส่งข้อมูลในรูปแบบบล็อกข้อความที่ไม่ต้องประสานการส่ง (Nonblocking) โดยแต่ละรูปแบบจะมีรูปแบบวิธีการส่งข้อความ (Send) และการรับข้อความ (Receive) ดังนี้

รูปแบบการส่งข้อความประสานกัน (Synchronization) แบ่งออกเป็น

1. วิธีการต้องประสานงานจากฝั่งส่ง (Blocking Sent) เป็นวิธีการส่งของกระบวนการ (Process) ในรูปแบบบล็อกข้อความ โดยที่ข้อความใหม่จะถูกส่งต่อไปได้ต้องได้รับสัญญาณยืนยัน (Acknowledgement) หรือการตอบกลับมาจากฝั่งผู้รับว่าได้รับบล็อกข้อความที่ส่งไปแล้ว

2. วิธีการต้องประสานงานจากฝั่งรับ (Blocking Receive) เป็นวิธีการรับของกระบวนการ (Process) ในรูปแบบบล็อกข้อความ โดยที่ฝั่งผู้รับจะต้องรอคอยสัญญาณข้อความจากฝั่งผู้ส่ง โดยฝั่งผู้รับจะไม่สามารถทำงานอย่างอื่นต่อไปได้ต้องรอจนกว่าบล็อกข้อความจะถูกส่งมาจนครบแล้ว

รูปแบบการส่งข้อความไม่ประสานกัน (Asynchronization) แบ่งออกเป็น

1. วิธีการไม่ต้องประสานงานจากฝั่งส่ง (Nonblocking Sent) เป็นวิธีการส่งของกระบวนการ (Process) ในรูปแบบบล็อกข้อความ โดยที่ข้อความใหม่จะถูกส่งต่อไปได้ โดยไม่ต้องรอสัญญาณยืนยัน (Acknowledgement) หรือการตอบกลับมาจากฝั่งผู้รับว่าได้รับบล็อกข้อความที่ส่งไปแล้ว โดยฝั่งผู้รับจะต้องมีที่พักข้อมูล (Buffer) สำหรับจัดเก็บข้อความที่ได้รับมาจากฝั่งผู้ส่งข้อความมา

2. วิธีการไม่ต้องประสานงานจากฝั่งรับ (Nonblocking Receive) เป็นวิธีการรับของกระบวนการ (Process) ในรูปแบบบล็อกข้อความ การรับข้อความในรูปแบบนี้ ฝั่งผู้รับสามารถทำงานอย่างอื่นไปพร้อมๆ กันได้ โดยไม่ต้องรอข้อความตอบกลับมา

6.5.4 การพักข้อมูล (Buffering) ไม่ว่าจะเป็นการติดต่อสื่อสารแบบทางตรง (Direct) หรือการสื่อสารแบบทางอ้อม (Indirect) การแลกเปลี่ยนข้อความข่าวสาร การลดขนาดของกระบวนการ (Process) สื่อสารภายในแถวลำดับชั่วคราว การจัดการเกี่ยวกับคิวสามารถกระทำได้ 3 ทางดังต่อไปนี้

1. ความจุเป็นศูนย์ (Zero Capacity) แถวลำดับมีความจุสูงสุดมีขนาดเท่ากับ 0 ดังนั้นการเชื่อมโยงจะไม่มีข้อความที่คอยอยู่ในแถวลำดับ ในกรณีนี้ ผู้ส่งจะส่งข้อความประสานกันจนกระทั้งผู้รับได้รับข้อความนั้น

2. ความจุเต็มขอบ (Bounded Capacity) แถวลำดับมีความจุเต็มขอบมีขนาดเท่ากับ n ดังนั้นข้อความทั้งหมดจำนวน n สามารถที่จะลดขนาด (Resize) ภายในแถวลำดับ ถ้าแถวลำดับไม่เต็มเมื่อมีข้อความใหม่ส่งเข้ามาจะถูกจัดเก็บลงในแถวลำดับ (ข้อความจะถูกคัดลอกหรือตัวชี้ไปยังข้อความจะถูกจัดเก็บเช่นเดียวกัน) และผู้ส่งสามารถที่จะประมวลผลข้อมูลต่อไปได้โดยไม่ตรงรอการเชื่อมโยงที่มีความจุเต็มขอบ อย่างไรก็ตามถ้าการเชื่อมโยงเต็มผู้รับต้องส่งข้อความประสานกันระหว่างกระบวนการว่ามีเนื้อที่ว่างในคิวที่สามารถใช้งานได้

3. ความจุไม่เต็มขอบ (Unbounded Capacity) แถวลำดับมีความจุไม่เต็ม ดังนั้นทุก ๆ จำนวนข้อความสามารถรอคอยอยู่ในแถวลำดับ ผู้ส่งไม่จำเป็นต้องส่งข้อความประสานกัน

สรุป

หน้าที่สำคัญประการหนึ่งของระบบปฏิบัติการ ก็คือการจัดสรรทรัพยากรของระบบการทำงานที่มีอยู่ทั้งภายในและภายนอก รวมทั้งอุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์ ได้อย่างมีประสิทธิภาพ สิ่งที่เกี่ยวข้องหลักก็คือ กระบวนการ (Process) ซึ่งเป็นขั้นตอนหรือวิธีการทำงานตามคำสั่งต่าง ๆ ที่สัมพันธ์กันกับการทำงานของระบบคอมพิวเตอร์

กระบวนการ (Process) ไม่ได้หมายถึงการทำงานหรือชุดคำสั่งที่กำลังประมวลผลในช่วงระยะเวลาใดเวลาหนึ่งเท่านั้น แต่อาจจะรวมถึงส่วนของข้อความ (Text Section) ที่ประกอบไปด้วยค่าของชุดคำสั่งตัวนับ (Program Counter) และรายละเอียดของรีจิสเตอร์ที่เกี่ยวข้องกับหน่วยประมวลผลกลาง (CPU) ดังนั้นการทำงานของระบบคอมพิวเตอร์จึงจำเป็นจะต้องมีการจัดการและจัดสรรเวลาการทำงานเป็นอย่างดี เพื่อให้วิธีการดำเนินงานแบบต่างๆ เกี่ยวกับกระบวนการ (Process) ที่มีอยู่ทั้งหมดอย่างมีประสิทธิภาพ

สถานะของกระบวนการ (Process State) การดำเนินการของการประมวลผลของแต่ละกระบวนการ (Process) จะเกี่ยวข้องกับการเปลี่ยนสถานะ (State) ของการทำงานที่เกิดขึ้นดังนี้ New เพื่อแสดงถึง การสร้างกระบวนการใหม่ (Created) ขึ้นมา Running เพื่อแสดงถึงคำสั่งที่จะได้รับการประมวลผล (Executed) Waiting เพื่อแสดงถึง กระบวนการ (Process) ที่รอคอย (Waiting) บางเหตุการณ์ที่กำลังจะเกิดขึ้น (รอใช้งานอุปกรณ์ I/O หรือรอรับสัญญาณ (signal) บางอย่าง) Ready เพื่อแสดงถึง กระบวนการ (Process) ที่รอ ว่าพร้อมจะใช้งานหน่วยประมวลผลกลาง (CPU) Terminal เพื่อแสดงถึง กระบวนการ (Process) ที่ประมวลผลอยู่ดำเนินการเสร็จสิ้นแล้ว

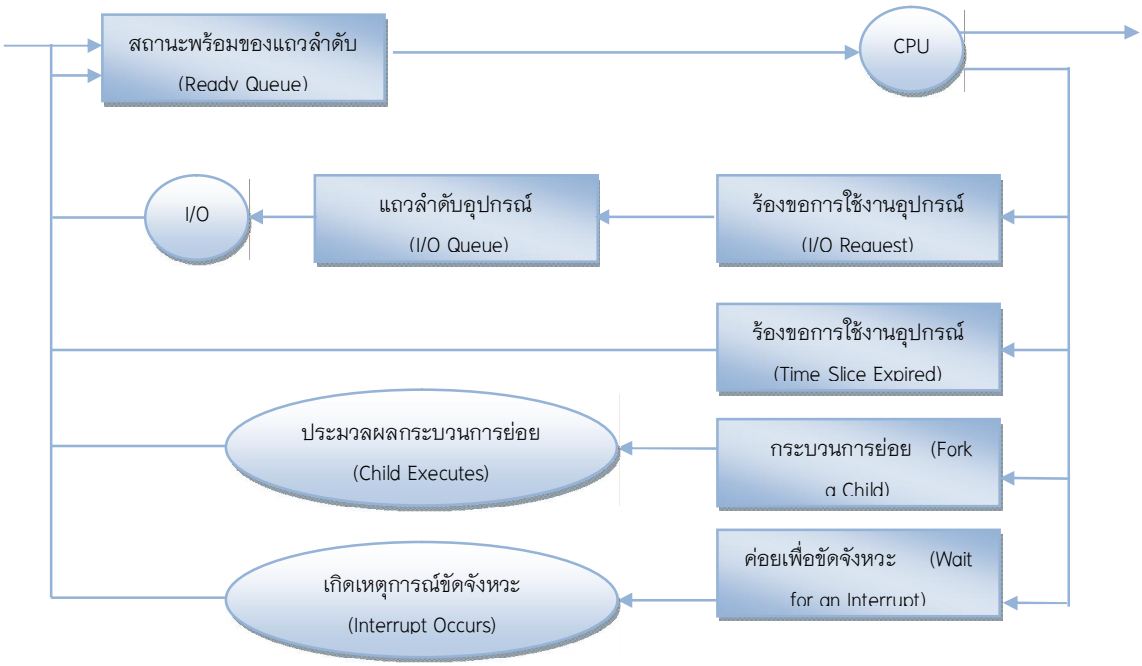
จุดประสงค์หลักของระบบการทำงานต้องการประมวลผลแต่ละกระบวนการ (Process) ได้อย่างรวดเร็วและมีประสิทธิภาพและใช้งานหน่วยประมวลผลกลางได้อย่างเต็มที่ (CPU Utilization) ดังนั้นจำเป็นจะต้องอาศัยองค์ประกอบเพื่อช่วยในการจัดการกับกระบวนการ (Process) ที่มีการเปลี่ยนแปลงสถานะอยู่ตลอดเวลา ตลอดจนการจัดการความถี่ของกระบวนการ (Process) ในการสลับเข้าและออกเพื่อเข้าไปใช้งานหน่วยประมวลผลการ (CPU) โดยที่ผู้ใช้งานสามารถที่จะใช้งานชุดคำสั่งได้ขณะที่กระบวนการ (Process) นั้น ๆ ถูกประมวลผลอยู่

กระบวนการ (Process) ต่างๆ ที่อยู่ในระบบสามารถประมวลพร้อมกันได้ภายในเวลาเดียวกัน ขณะเดียวกันการสร้าง (Created) และลบ (Deleted) กระบวนการก็เกิดขึ้นได้ตลอดเวลาเช่นเดียวกัน ดังนั้นระบบปฏิบัติการจะต้องมีกลไกหรือความสามารถในการสร้างกระบวนการ (Process Creation) โดยเมื่อกระบวนการได้ดำเนินการเสร็จสิ้น (Process Termination) ก็สามารถที่จะลบกระบวนการนั้นทิ้งไปได้

ในระบบคอมพิวเตอร์โดยส่วนใหญ่สามารถที่จะประมวลผลได้หลาย ๆ ชุดคำสั่งในเวลาเดียวกัน (Multiprogramming) มีการเชื่อมโยงและติดต่อสื่อสารกันในระบบเครือข่าย รวมทั้งมีการแบ่งปันทรัพยากรและใช้งานกระบวนการ (Process) ร่วมกันได้ ซึ่งบางครั้งเราจะเรียกวิธีการติดต่อสื่อสารแบบนี้ว่า การติดต่อ สื่อสารระหว่างกระบวนการ (Interprocess Communication: IPC)

คำถามทบทวน

1. จงอธิบายการเปลี่ยนสถานะต่างๆ ของการดำเนินการของการประมวลผลของแต่ละกระบวนการ
2. บล็อกควบคุมการทำงาน (Task Control Block) มีหน้าที่อะไรและทำงานอย่างไร
3. Multithreading คืออะไรจงอธิบาย
4. จงแสดงแผนภาพการทำงานของหน่วยประมวลผลกลางในการเปลี่ยนสถานะระหว่าง 2 กระบวนการว่ามีวิธีการทำงานอย่างไร
5. ในหน่วยประมวลผลกลาง (CPU) โดยระหว่างการดำเนินการอยู่นั้นอาจมีเหตุการณ์ใดเหตุการณ์หนึ่งสามารถที่จะเกิดขึ้นได้อย่างไรบ้าง
6. จงอธิบายแผนภาพลำดับการทำงาน (Queuing Diagram) ที่ให้มาอย่างละเอียด



7. จงอธิบายวิธีการจัดตารางการทำงาน (Scheduler) ต่อไปนี้
    - 7.1 การจัดตารางการทำงานระยะยาว (Long-term scheduler)
    - 7.2 การจัดตารางการทำงานระยะกลาง (Medium-term scheduler)
    - 7.3 การจัดตารางการทำงานระยะสั้น (Short-term scheduler)

8. การสิ้นสุดกระบวนการ (Process Termination) เกิดขึ้นได้อย่างไรบ้างจงอธิบาย

9. เราสามารถที่จะกำหนดสภาพแวดล้อมที่อนุญาตให้การร่วมกันของกระบวนการด้วยเหตุผลใดได้บ้าง

10. จงอธิบายวิธีการติดต่อสื่อสารระหว่างกระบวนการ (Interprocess Communication)

    10.1 วิธีระบบการส่งข้อความ (Message-Passing System)

    10.2 วิธีการระบุช่องทางสื่อสาร (Naming)

    10.3 วิธีการส่งข้อความแบบประสานกันหรือแบบไม่ประสานกัน (Synchronization or Asynchronization)

    10.4 วิธีการพักข้อมูล (Buffering)

เอกสารอ้างอิง

ราชบัณฑิตยสถาน. (2544). *ศัพท์บัญญัติ ราชบัณฑิตยสถาน*. ค้นเมื่อ 21 ธันวาคม 2556, จาก: http://rirs3.royin.go.th/coinages

กระบวนการ. (2556). *วิกิพีเดีย*. ค้นเมื่อ 21 ธันวาคม 2556, จาก: http://th.wikipedia.org /wiki

พีรพร หมุนสนิท, สุธี พงศาสุกุลชัย, อัจจิมา เลี้ยงอยู่. (2553). *ระบบปฏิบัติการ: Operating Systems*. กรุงเทพฯ : เคทีพี แอนด์ คอนซัลท์.

พีระพนธ์ โสพัศสถิตย์. (2552). *ระบบปฏิบัติการ*. Operating Systems. กรุงเทพฯ : สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย.

Silberschartz, Galvin, Gangne. (2011). *Operating System Concepts*. 8 th (ed), New York: McGra Hill.