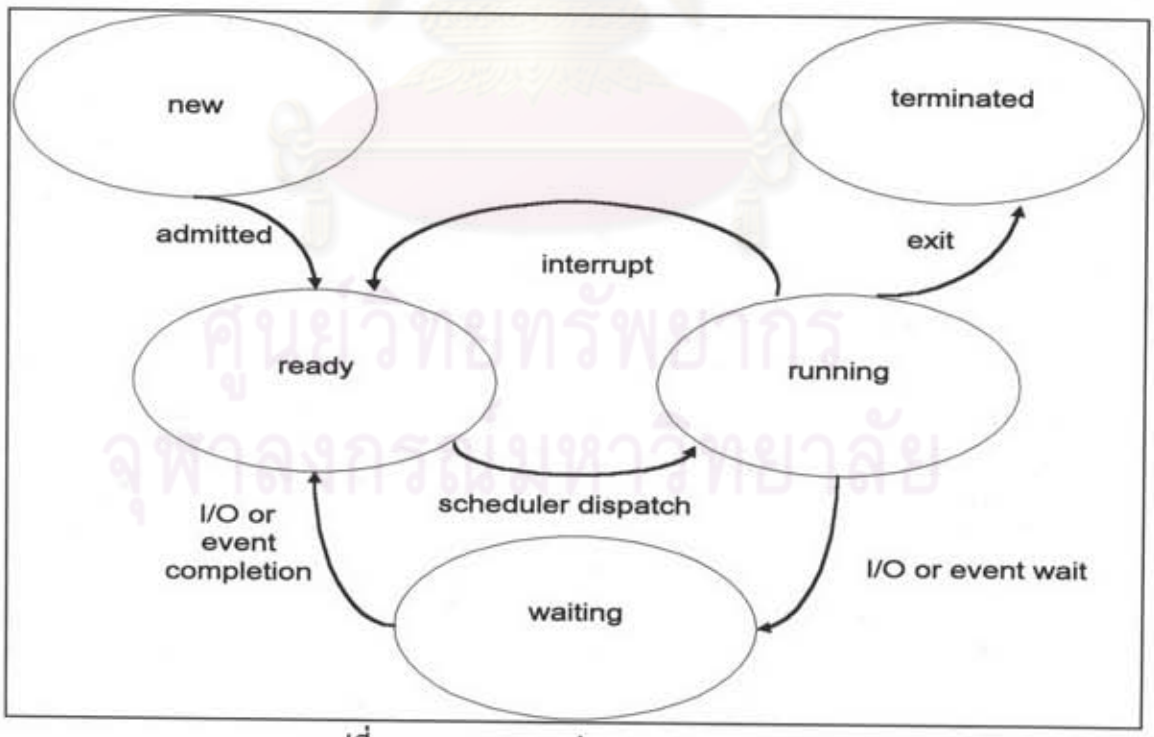
บทที่ 2

แนวคิดและทฤษฎีที่เกี่ยวข้อง

ระบบปฏิบัติการ(Operating system)

ระบบปฏิบัติการ คือ กลุ่มของโปรแกรมซึ่งได้รับการจัดระเบียบให้เป็นส่วนเชื่อมโยงระหว่างผู้ใช้และเครื่องคอมพิวเตอร์ โดยจะเอื้ออำนวยในการพัฒนาและการใช้โปรแกรม รวมทั้งการจัดสรรทรัพยากร(resource) ต่างๆให้มีประสิทธิผลที่ดี ในระบบปฏิบัติการแบบแบ่งกันใช้เวลา มีการจัดสรรทรัพยากรต่างๆให้กับกระบวนการที่ร้องขอซึ่งมีภาวะที่แตกต่างกันในระบบและแถวคอย ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 แสดงภาวะต่างๆของกระบวนการ

เมื่อกระบวนการทำงานอยู่ในระบบจะมีภาวะตามกิจกรรมที่ทำ ซึ่งอาจเป็นดังนี้

- กระบวนการถูกสร้างขึ้นใหม่(new)
- กระบวนการกำลังถูกดำเนินการอยู่(running)
- กระบวนการกำลังรอคอย(waiting) เหตุการณ์บางอย่างอยู่ เช่น รอให้การทำงานของอุปกรณ์รับเข้าและแสดงผลดำเนินการเสร็จ
- กระบวนการที่รอพร้อมเข้ารับบริการ(ready)
- กระบวนการดำเนินการเสร็จสิ้น(terminated)

การจัดสรรทรัพยากรต่างๆให้กับกระบวนการที่อยู่ในภาวะพร้อมเข้ารับบริการ เป็นหน้าที่ของตัวสับหลีกงานของแต่ละทรัพยากรในระบบ ที่สำคัญมีดังนี้

1. การสับหลีกงานของหน่วยประมวลผลกลาง ดังแสดงในรูปที่ 2.2 เป็นการจัดสรรแบ่งเวลาของหน่วยประมวลผลกลางให้กับแต่ละกระบวนการ ที่กำลังดำเนินอยู่ในหน่วยความจำ โดยเมื่อไรที่กระบวนการใดกระบวนการหนึ่งมีการเรียกใช้ทรัพยากรอื่น หรือมีการรอเกิดขึ้น ระบบปฏิบัติการจะส่งหน่วยประมวลผลกลางไปทำงานให้กับกระบวนการอื่นแทน อัลกอริธึมที่สำคัญมีดังนี้

1.1 การสับหลีกงานแบบมาก่อนให้บริการก่อน (First-Come, First-Served Scheduling) กระบวนการที่ร้องขอใช้หน่วยประมวลผลกลางก่อนจะได้รับบริการก่อนโดยใช้อัลกอริธึมแถวคอยแบบเข้าก่อนออกก่อน (First-In, First-Out queue or FIFO queue) เมื่อมีกระบวนการเข้ามารออยู่ในแถวคอย ส่วนของโปรเซสคอนโทรลบล็อก(Process Control Block, PCB) ซึ่งเป็นส่วนที่เก็บข้อมูลต่างๆของแต่ละกระบวนการจะถูกนำไปต่อท้ายในแถวคอย เมื่อหน่วยประมวลผลกลางว่างจะให้บริการแก่กระบวนการที่รออยู่ในส่วนหัวสุดของแถวคอย หลังจากนั้นกระบวนการนี้จะถูกนำออกจากแถวคอย การสับหลีกงานแบบนี้จะไม่มีการตัดตอน (preemption)เข้ามาเกี่ยวข้อง เวลาเฉลี่ยในการคอยของแต่ละกระบวนการมีค่าค่อนข้างมาก

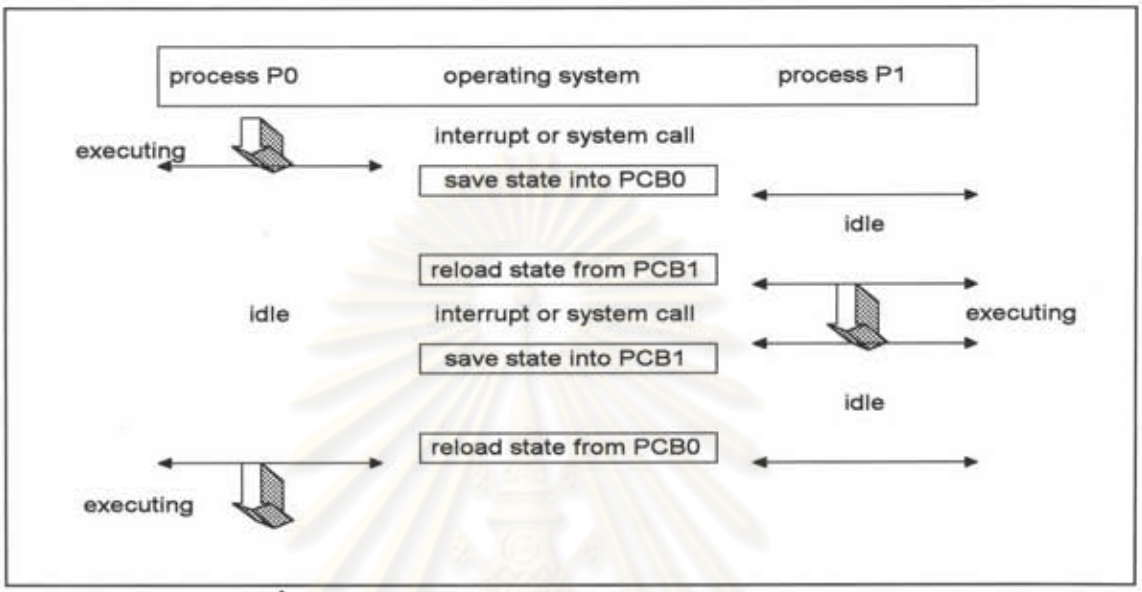
1.2 การสับหลีกงานแบบงานสั้นที่สุดก่อน (Shortest-Job-First Scheduling) อัลกอริธึมนี้มีความสัมพันธ์กับระยะเวลาที่กระบวนการจะใช้หน่วยประมวลผลกลาง (CPU burst time) ลำดับถัดไป โดยกระบวนการที่มีเวลาน้อยที่สุดจะได้รับบริการจากหน่วยประมวลผลกลางก่อน แต่ถ้าทั้งสองกระบวนการใช้เวลาเท่ากันกระบวนการที่มาก่อนจะได้รับบริการก่อน การสับหลีกงานแบบนี้จะไม่มีการตัดตอนเข้ามาเกี่ยวข้อง เวลาเฉลี่ยในการคอยของแต่ละกระบวนการจะมีค่าต่ำที่สุด ความยากของอัลกอริธึมนี้คือ การที่ต้องรู้ระยะเวลาในการใช้

หน่วยประมวลผลกลางของแต่ละกระบวนการ ซึ่งปกติกำหนดโดยผู้ใช้สำหรับการประมวลผลแบบกลุ่ม(batch processing) ส่วนการสับหลีกงานแบบระยะสั้น(short-term scheduling) จะไม่ใช้อัลกอริธึมนี้ เนื่องจากไม่สามารถรู้ระยะเวลาล่วงหน้าในการใช้หน่วยประมวลผลกลางของแต่ละกระบวนการ

1.3 การสับหลีกงานตามระดับความสำคัญ (Priority Scheduling) กระบวนการที่มีระดับความสำคัญสูงสุดจะได้รับบริการจากหน่วยประมวลผลกลางก่อน ส่วนกระบวนการที่มีระดับความสำคัญเท่ากันจะได้รับบริการตามลำดับแบบมาก่อนได้รับบริการก่อน โดยทั่วไปตัวเลขแสดงระดับความสำคัญจะมีค่าเป็นช่วง เช่น 0 ถึง 9 หรือ 0 ถึง 4095 เป็นต้น การกำหนดให้เลข 0 แทนระดับความสำคัญสูงสุดหรือต่ำสุดขึ้นอยู่กับข้อตกลงของแต่ละระบบ ระดับความสำคัญสามารถกำหนดได้ในสองลักษณะคือแบบภายในและแบบภายนอก การกำหนดแบบภายในจะพิจารณาจากระยะเวลา หน่วยความจำที่ต้องการ จำนวนแฟ้มข้อมูลที่เปิด และสัดส่วนค่าเฉลี่ยเวลาที่ใช้อุปกรณ์รับเข้า-ส่งออก(I/O device) ต่อค่าเฉลี่ยเวลาที่ใช้หน่วยประมวลผลกลางของแต่ละกระบวนการ ส่วนการกำหนดแบบภายนอกจะพิจารณาจากองค์ประกอบที่ไม่เกี่ยวข้องกับระบบปฏิบัติการ เช่น ความสำคัญ จำนวนเงินที่จ่ายสำหรับการใช้คอมพิวเตอร์ เป็นต้น การสับหลีกงานแบบนี้มีทั้งแบบตัดตอนได้และตัดตอนไม่ได้

1.4 การสับหลีกงานแบบเวียนเทียน (Round-Robin Scheduling) ลักษณะของอัลกอริธึมคล้ายกับแบบมาก่อนได้รับบริการก่อนโดยมีเพิ่มการตัดตอนเข้ามา เพื่อสลับให้กระบวนการอื่นได้ใช้หน่วยประมวลผลกลาง แต่ละกระบวนการจะได้รับบริการจากหน่วยประมวลผลกลางเป็นระยะเวลาเท่ากับช่วงเวลาที่แบ่งไว้ ซึ่งเรียกว่าไทม์ควอนตัมหรือไทม์สไลซ์ (time quantum or time slice) โดยทั่วไปมีค่าอยู่ระหว่าง 0 ถึง 100 มิลลิวินาที แถวคอยของการสับหลีกงานแบบนี้เป็นแบบวงกลม (circular queue) โดยใช้อัลกอริธึมแบบเข้าก่อนออกก่อน ตัวสับหลีกงานจะเลือกกระบวนการจากส่วนหัวของแถวคอยพร้อมทั้งตั้งค่าเวลา เพื่อใช้ขัดจังหวะ (interrupt) หลังจากครบเวลา 1 ไทม์ควอนตัม ไปยังระบบปฏิบัติการ สำหรับกระบวนการที่ใช้เวลาน้อยกว่า 1 ไทม์ควอนตัม จะตัดตอนตัวเองออกจากการใช้หน่วยประมวลผลกลาง เพื่อให้กระบวนการอื่นในแถวคอยได้ใช้บริการ กระบวนการที่ใช้เวลาของหน่วยประมวลผลกลางมากกว่า 1 ไทม์ควอนตัม จะถูกตัดตอนออกมาโดยการขัดจังหวะไปยังระบบปฏิบัติการให้ทำคอนเท็กซ์สวิทช์(context switch)และนำเอากระบวนการนี้ไปต่อแถวคอย เพื่อให้ตัวสับหลีกงานเลือกกระบวนการอื่นที่ส่วนหัวของแถวคอยเข้ารับบริการแทน เวลาเฉลี่ยในการคอยของแต่ละกระบวนการขึ้นอยู่กับเวลาที่ใช้ทำคอนเท็กซ์สวิทช์และเทิร์นอะราวด์ไทม์(turnaround time) ซึ่งโดยทั่วไปไทม์ควอนตัมควรมีขนาดใหญ่เมื่อเทียบกับเวลาในการทำคอนเท็กซ์สวิทช์ และขนาด

ของเทิร์นอะราวด์ไทม์ขึ้นอยู่กับขนาดของไทม์ควอนตัมกล่าวคือ ถ้ากระบวนการส่วนใหญ่สามารถดำเนินได้เสร็จภายใน 1 ไทม์ควอนตัม ค่าของเทิร์นอะราวด์ไทม์เฉลี่ยจะลดลง



ที่ 2.2 แสดงการสับหลีกงานของหน่วยประมวลผลกลาง

2. การจัดการหน่วยความจำแบบเพจ การแบ่งหน่วยความจำทางกายภาพออกเป็นส่วนย่อยๆ ขนาดคงที่เรียกว่าเฟรม (frame) และการแบ่งหน่วยความจำทางตรรกะออกเป็นส่วนย่อยๆ ที่มีขนาดเท่ากันเรียกว่าเพจ (page) การอ้างอิงเพจในทางตรรกะจะต้องแปลงที่อยู่ของเพจในทางตรรกะให้เป็นที่อยู่จริงในหน่วยความจำทางกายภาพ โดยใช้ตารางเพจซึ่งเก็บข้อมูลเกี่ยวกับหมายเลขเพจและค่าเพจอ็อฟเซ็ท(page offset) ดังแสดงในรูปที่ 2.3 การศึกษาอัลกอริธึมการแทนที่เพจ ใช้การแก้ปัญหาแบบศึกษาสำนึก (approach) ดังนี้คือ ถ้าไม่มีเฟรมว่างจะค้นหาเฟรมที่ไม่ได้ถูกใช้ในขณะนั้นและทำให้เฟรมนั้นว่างโดยการนำข้อมูลไปเก็บไว้ในพื้นที่สับเปลี่ยน (swap space) และเปลี่ยนแปลงข้อมูลในตารางเพจ (page table) และตารางอื่นที่เกี่ยวข้องเพื่อแจ้งให้ระบบรู้ว่าไม่มีเพจนั้นอยู่ในหน่วยความจำแล้ว หลังจากนี้เฟรมที่ว่างสามารถนำไปใช้เก็บเพจจากกระบวนการที่ต้องการได้ซึ่งวิธีการนี้จะทำให้การบริการเมื่อขาดเพจ (page fault) ของระบบทำการปรับปรุงข้อมูลเพื่อรวมเอาเพจจากการแทนที่เข้าไว้ด้วย อัลกอริธึมการแทนที่เพจที่สำคัญมีดังนี้

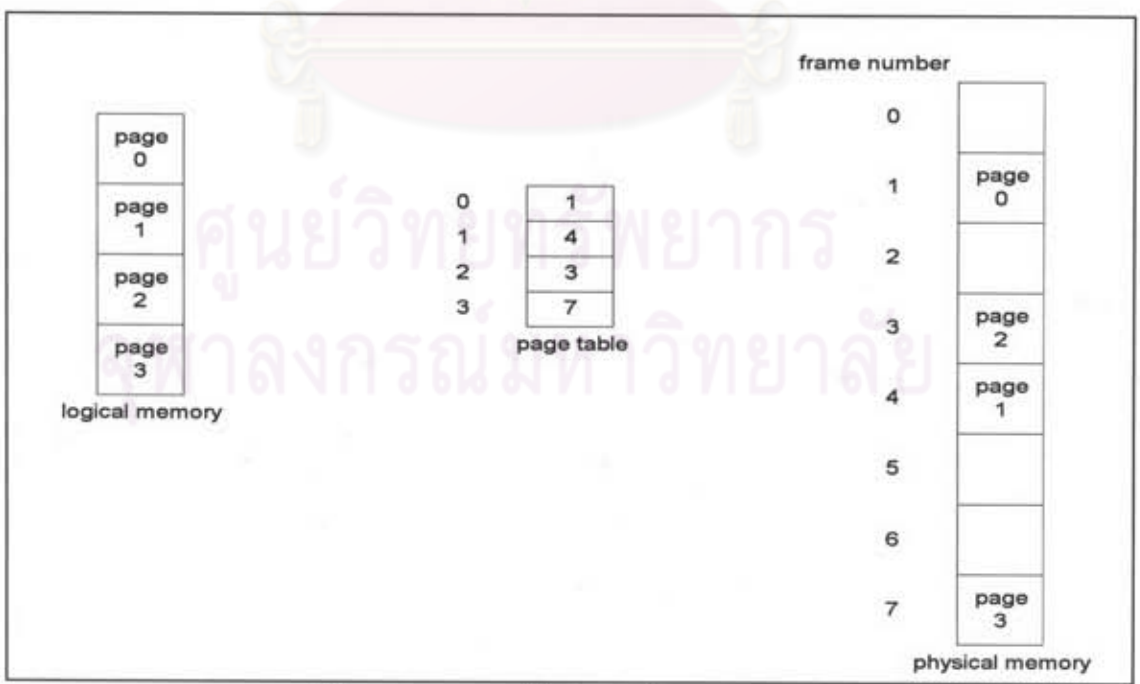
2.1 อัลกอริธึมแบบเข้าก่อนออกก่อน อัลกอริธึมนี้แต่ละเพจในหน่วยความจำจะมีความสัมพันธ์กับเวลาซึ่งการแทนที่เพจจะเกิดขึ้นกับเพจที่อยู่ในหน่วยความจำมาแล้วนานที่สุดก่อน

2.2 อัลกอริธึมการแทนที่ซึ่งได้ผลดีที่สุด (Optimal Algorithm) ใช้หลักดังนี้คือให้แทนที่เพจซึ่งจะไม่ได้ใช้นานที่สุด วิธีนี้จะให้ปริมาณการขาดเพจต่ำที่สุดเมื่อเทียบกับวิธีอื่น โดย

การมองไปในอนาคตดูว่าเพจต่างๆ ในหน่วยความจำขณะนั้นเพจใดจะถูกอ้างอิงช้าที่สุดก็เลือกเอาเพจนั้นออกแล้วใส่เพจใหม่เข้าไปแทนที่

2.3 อัลกอริธึมการแทนที่แบบไม่ได้ใช้นานที่สุด(Least Recently Used Algorithm) ใช้หลักดังนี้คือ ให้แทนที่เพจซึ่งไม่ได้ใช้นานที่สุด โดยในจำนวนเพจที่อยู่ในหน่วยความจำ เพจที่ไม่ได้ถูกอ้างอิงนานที่สุดในขณะนั้นจะมีโอกาสสูงที่จะไม่ถูกอ้างอิงในอนาคตอันใกล้ ซึ่งวิธีนี้เป็นการใช้วิธีแบบได้ผลดีที่สุด และการมองย้อนไปในอดีตเพื่อดูสถิติเวลาของแต่ละเพจว่ามีการถูกอ้างอิงมากน้อยขนาดไหน

2.4 อัลกอริธึมการแทนที่แบบไม่ได้ใช้นานที่สุดโอกาสที่สอง (Second-Chance Least Recently Used Algorithm) โดยการใช้บิตอ้างอิง(reference bit) จากฮาร์ดแวร์ตั้งค่าเพจที่ถูกอ้างอิง ซึ่งบิตอ้างอิงของแต่ละเพจจะถูกเก็บอยู่ที่ตารางเพจ อัลกอริธึมนี้ใช้หลักการแทนที่แบบเข้าก่อนออกก่อนคือ เมื่อเพจถูกเลือกจะดูบิตอ้างอิงว่ามีค่าเป็นอะไร ถ้ามีค่าเป็น 0 เพจที่เลือกนี้จะถูกแทนที่ ถ้ามีค่าเป็น 1 เพจที่ถูกเลือกนี้จะยังคงอยู่ในหน่วยความจำต่อไปและระบบจะเลือกเพจถัดไปที่อยู่ในแถวคอยจนกระทั่งเพจที่เคยถูกเลือกกลับมาถูกเลือกอีกครั้ง ถ้าครั้งนี้บิตอ้างอิงมีค่าเป็น 0 และมีค่าเวลาเป็นเวลาปัจจุบัน เพจนี้จะได้รับโอกาสให้อยู่ในหน่วยความจำอีกครั้งหนึ่ง จนกระทั่งเพจอื่นๆ ถูกแทนที่หมดแล้วเพจดังกล่าวนี้จึงจะถูกแทนที่ ในกรณีที่เพจใดถูกใช้บ่อยมากค่าของบิตอ้างอิงจะถูกตั้งค่าเป็น 1 อยู่ตลอดเวลา ดังนั้นเพจนี้จะไม่ถูกแทนที่



รูปที่ 2.3 แสดงการอ้างอิงเพจผ่านทางตารางเพจ

3. การสับหลีกงานของจานบันทึกข้อมูล ระบบปฏิบัติการใช้การสับหลีกงานของจานบันทึกข้อมูลเพื่อให้บริการแก่กระบวนการต่างๆในระบบที่อยู่ในแถวคอย แต่ละกระบวนการที่ร้องขอใช้บริการจะประกอบด้วยข้อมูลที่สำคัญต่างๆ เช่น การร้องขอเป็นแบบอ่านหรือเขียน ที่อยู่ของข้อมูลบนจานบันทึกข้อมูล ซึ่งประกอบด้วยหมายเลขบล็อก(block number) วงซ้อน (cylinder) ส่วนวง (sector) พื้นผิวของจานบันทึก (surface) ที่อยู่ของข้อมูลในหน่วยความจำ จำนวนข้อมูลที่ต้องการอ่านหรือเขียน เวลาในการคอยของแต่ละกระบวนการจะขึ้นอยู่กับเวลาที่หัวอ่าน-เขียนใช้ค้นหาแทรคที่ต้องการ(seek time) เวลาที่จานบันทึกข้อมูลหมุนมาอยู่ตรงกับส่วนวงที่ต้องการ(latency time) และเวลาที่ใช้อ่านหรือเขียนข้อมูล(transfer time) อัลกอริธึมที่สำคัญมีดังนี้

3.1 การสับหลีกงานแบบมาก่อนให้บริการก่อน กระบวนการที่ร้องขอใช้จานบันทึกข้อมูลก่อนจะได้รับบริการก่อน

3.2 การสับหลีกงานแบบใช้เวลาค้นหาสั้นที่สุดก่อน (Shortest-Seek-Time-First Scheduling) อัลกอริธึมนี้จะให้บริการแก่กระบวนการที่อยู่ในแถวคอยซึ่งใช้เวลาน้อยที่สุดในการค้นหาแทรค (track) ที่ต้องการหรือแทรคที่อยู่ใกล้ที่สุดก่อน

3.3 การสับหลีกงานแบบกวาดตรวจไปมา (Scan Scheduling) หัวอ่าน-เขียนของจานบันทึกข้อมูลจะเคลื่อนที่โดยเริ่มต้นจากปลายจานบันทึกข้อมูลด้านหนึ่งไปยังอีกด้านหนึ่งทั้งไปและกลับ โดยระหว่างทางที่เคลื่อนที่ไปถ้าพบแทรคใดถูกร้องขอจากกระบวนการใดก็จะให้บริการแก่กระบวนการนั้นก่อน

3.4 การสับหลีกงานแบบกวาดตรวจไปมาเป็นวงกลม (Circular Scan Scheduling) หัวอ่าน-เขียนของจานบันทึกข้อมูลจะเคลื่อนที่จากปลายด้านหนึ่งไปยังอีกด้านหนึ่ง ในระหว่างทางที่ผ่าน ถ้าพบแทรคใดถูกร้องขอโดยกระบวนการใดก็จะหยุดให้บริการแก่กระบวนการนั้น เมื่อหัวอ่าน-เขียนเคลื่อนที่ไปสุดปลายด้านหนึ่งจะเคลื่อนที่กลับไปยังจุดเริ่มต้นทันที โดยไม่ให้บริการแก่กระบวนการใดๆ ทั้งสิ้น และจากนั้นจะเคลื่อนหัวอ่าน-เขียนจากจุดเริ่มต้นไปปลายทางและให้บริการแก่กระบวนการที่ร้องขอเมื่อพบแทรคที่ต้องการในระหว่างทางที่หัวอ่าน-เขียนเคลื่อนที่ผ่านต่อไปสลับกันเช่นนี้เรื่อยไป อัลกอริธึมนี้จะมองแทรคเริ่มต้นและแทรคสุดท้ายของจานบันทึกข้อมูลเหมือนเป็นวงกลมโดยมีจุดทั้งสองเป็นจุดบรรจบกัน

โปรแกรมภาษาวิชวลเบสิกสำหรับวินโดว์ส(Visual Basic for Windows)

โปรแกรมภาษาวิชวลเบสิกสำหรับวินโดว์ส เป็นซอฟต์แวร์ที่มีความสามารถในการพัฒนาโปรแกรมต่างๆ ภายใต้ระบบปฏิบัติการวินโดว์ส วิชวลเบสิกได้รวมเอาความสามารถของภาษาเบสิกเข้าไว้กับวิชวลดีซายน์ทูลส์ (visual design tools) ทำให้ใช้งานง่ายและสมรรถนะทางด้านกราฟิกของวินโดว์สไม่เปลี่ยนแปลง การออกแบบเมนู (menu) ฟอนต์ (font) ไดอะล็อกบ็อกซ์ (dialog box) สกรอลลิงเท็กซ์ฟิลด์ (scrolling text field) และส่วนประกอบอื่นๆสามารถทำได้โดยง่าย นอกจากนี้วิชวลเบสิกยังเป็นภาษาคอมพิวเตอร์ที่สนับสนุนอีเวนต์ไดรเวนโปรแกรมมิ่ง (event-driven programming) ซึ่งเป็นรูปแบบการเขียนโปรแกรมที่เหมาะกับกราฟิคัลยูเซอร์อินเตอร์เฟส (Graphical User Interface, GUI) โดยนักเขียนโปรแกรม (programmer) ไม่ต้องเขียนโปรแกรมแสดงขั้นตอนทุกๆ ขั้นออกมาเรียงตามลำดับเพียงแต่เขียนโปรแกรมที่ตอบสนองต่อกิจกรรมที่ทำ เช่น การเลือกคำสั่ง การคลิกเม้าส์(click mouse)ในวินโดว์ การเคลื่อนย้ายเม้าส์ หรือเขียนโปรแกรมเป็นกลุ่มของมินิโปรแกรม (mini-program) ซึ่งจะทำงานโดยอีเวนต์ที่ผู้ใช้ริเริ่มขึ้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย