บทที่ 2
ทฤษฎีและแนวคิดที่นำมาใช้ในการวิจัย

การควบคุมติดตามการทำงานของระบบคอมพิวเตอร์ หมายถึง การตรวจสอบการใช้งานของทรัพยากรระบบ คือหน่วยประมวลผลกลาง หน่วยความจำหลัก และหน่วยอุปกรณ์รับเข้าและส่งออก ซึ่งแต่ละหน่วยจะมีคุณลักษณะเฉพาะแตกต่างกันไป ทรัพยากรของระบบเหล่านี้มีผลต่อความสามารถหรือประสิทธิภาพของระบบคอมพิวเตอร์ในทางหนึ่งทางใด ผู้บริหารระบบควรพิจารณาถึงปัญหาและอุปสรรค ตลอดจนแนวทางการแก้ไขปัญหาของการใช้งานทรัพยากรในแต่ละหน่วยเพื่อการให้บริการแก่ผู้ใช้งานในระบบอย่างสอดคล้องเท่าเทียมกัน และเนื่องจากทรัพยากรเหล่านี้มีความเกี่ยวข้องสัมพันธ์กัน การปรับเปลี่ยนต่างๆอาจมีผลทำให้ทรัพยากรบางหน่วยทำงานดีขึ้น ในขณะเดียวกันเป็นการเพิ่มภาระงานให้แก่หน่วยอื่นๆ หรือไม่เป็นการเพิ่มประสิทธิภาพการทำงานของระบบ เช่นกรณีปรับค่ารอบการทำงานของหน่วยประมวลผลกลางเพิ่มขึ้น แต่โปรแกรมที่ทำงานมีการเรียกใช้หน่วยอุปกรณ์รับเข้าและส่งออกมาก การเพิ่มรอบการทำงานดังกล่าวก็ไม่มีผลต่อความสามารถโดยรวมของระบบแต่อย่างใด

2.1 ทรัพยากรของระบบคอมพิวเตอร์

การตรวจสอบถึงปัญหาจากการทำงานของระบบคอมพิวเตอร์ที่เกิดขึ้นว่าเกิดจากส่วนใดนั้นเป็นเรื่องที่ซับซ้อน เพราะการทำงานของทรัพยากรระบบเหล่านี้มีความสัมพันธ์ต่อกัน โดยอาศัยการตรวจสอบการทำงานของแต่ละส่วนของทรัพยากรระบบ จะทำให้สามารถบอกข้อสมมุติฐานได้ว่าเกิดปัญหาขึ้นที่ส่วนใด

2.1.1 หน่วยประมวลผลกลาง

ในระบบการจัดแบ่งเวลา โปรแกรมการทำงานจะเข้าใช้งานในหน่วยประมวลกลางพร้อมกันหลายโปรแกรม ระบบปฏิบัติการจะจัดการการทำงานเหล่านี้ โดยใช้วงรอบการทำงานของหน่วยประมวลผลกลาง (CPU Cycle) เมื่อมีการทำงานของโปรแกรมมากขึ้นๆ จนจุดหนึ่งหน่วยประมวลผลกลางไม่สามารถทำงานได้ หรือโปรแกรมใช้เวลาในการทำงานมากขึ้นกว่าเดิม ลักษณะนี้ควรตรวจสอบการทำงานของหน่วยประมวลกลางในขณะนั้น ว่ามีจำนวนโปรเซสที่ทำงานอยู่เท่าใด หรือกิจกรรมของระบบมีอะไรบ้าง นอกจากการที่ระบบทำงานช้าลงจากการช่วงชิงการทำงานในหน่วยประมวลผลกลางแล้ว ขนาดของหน่วยความจำหลักที่เหลือน้อย หรือการ

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ทำงานของส่วนอุปกรณ์รับเข้า และส่งออกที่มีความเร็วต่ำ ก็มีผลทำให้ระบบทำงานช้าลงได้ ดังนั้นจึงต้องตรวจสภาพการทำงานของทั้งหน่วยความจำหลัก และหน่วยอุปกรณ์รับเข้า และส่งออก

2.1.2 หน่วยความจำหลัก

ปัญหาของหน่วยความจำหลักจะเกิดขึ้น เมื่อมีจำนวนโปรเซสที่ทำงานมากเกินกว่าความจุของหน่วยความจำหลัก ระบบปฏิบัติการจะเริ่มกระบวนการสับค่าหน้า (Paging) เพื่อจัดการป้องกันมิให้ระบบเกิดขัดข้องและยังคงทำงานต่อไปได้ โดยการนำโปรเซสบางส่วนเก็บไว้ในหน่วยความจำสำรองเพื่อเพิ่มเนื้อที่ใช้สอยในหน่วยความจำหลัก กระบวนการนี้มีผลให้ความสามารถของระบบลดลง ความแตกต่างระหว่างกระบวนการสับค่าหน้า และกระบวนการเปลี่ยนโปรเซส (Swapping) คือกระบวนการสับค่าหน้าจะนำทั้งโปรเซสเก็บไว้ในหน่วยความจำสำรอง แต่ทั้งสองกระบวนการเป็นข้อบ่งชี้ว่า ระบบมีเนื้อที่ในหน่วยความจำหลักไม่เพียงพอสำหรับทุกๆ โปรเซสที่กำลังทำงาน

2.1.3 หน่วยอุปกรณ์รับเข้า และส่งออก

งานทุกงานในระบบ หมายรวมถึงงานในส่วนของระบบเอง จะร่วมกันใช้อุปกรณ์หน่วยรับเข้า และส่งออกซึ่งมีจำนวนจำกัด ทำให้เกิดการจัดลำดับการเข้าใช้อุปกรณ์ และการรอคอยเพื่อเข้าใช้อุปกรณ์ขึ้น เนื่องจากอุปกรณ์เหล่านี้มีคุณลักษณะเฉพาะแตกต่างกัน ดังนั้นข้อจำกัด และปัญหาในการใช้งานแตกต่างกัน ทรัพยากรระบบในส่วนการทำงานของหน่วยรับเข้าและส่งออก ที่มีผลต่อการสมรรถนะการทำงานของระบบคอมพิวเตอร์มาก คือการทำงานของหน่วยความจำสำรองชนิดจานแม่เหล็ก และการเชื่อมโยงเครือข่าย

ปัญหาจากทรัพยากรระบบเหล่านี้เป็นปัญหาขั้นพื้นฐานที่ต้องตัดสินใจว่าระบบคอมพิวเตอร์นั้นจะต้องมีการปรับปรุงหรือไม่ หรือปรับปรุงอย่างไร เช่น การขยายระบบซึ่งต้องเสียค่าใช้จ่ายมาก หรือการปรับการทำงานของทรัพยากรระบบที่มีอยู่ ซึ่งต้องอาศัยการควบคุมติดตามการทำงานของระบบคอมพิวเตอร์เพื่อเก็บข้อมูลสำหรับใช้วิเคราะห์ความสามารถของระบบ

นอกจากจะปัญหาดังกล่าวแล้วก็ต้องพิจารณาถึงผู้ใช้งานในระบบซึ่งเป็นผู้ส่งงานเข้าสู่ระบบเพื่อการประมวลผลด้วย การจัดการทรัพยากรของระบบคอมพิวเตอร์อย่างมีประสิทธิภาพนั้นต้องอาศัยความร่วมมือของทุกๆ ส่วนในระบบ ความร่วมมือนี้เกิดจากการทำความเข้าใจกันระหว่างผู้ใช้งานและผู้บริหารระบบ เช่น การให้ผู้ใช้งานส่งงานเข้าสู่ระบบแบบแบทช์ ควรชี้แจงให้ผู้ใช้งานเข้าใจว่าจะช่วยให้ระบบทำงานดีขึ้นอย่างไร ลักษณะของผู้ใช้งานในระบบจะมีผลต่อความสามารถ และขั้นตอนในการทำงานเพื่อการปรับปรุงความสามารถของระบบ ผู้ใช้อาจแบ่งตามลักษณะงาน ดังนี้

1) ผู้ใช้งานทั่วไป เช่น การทำงานของโปรแกรมอรรถประโยชน์ต่างๆ การสร้าง หรือการแก้ไขแฟ้มข้อมูล การจัดลำดับข้อมูล การค้นหาแฟ้มข้อมูล

2) ผู้ใช้ที่ทำงานเกี่ยวข้องกับข้อมูลมาก เช่น งานการจำลองแบบ

3) ผู้ใช้ที่ต้องการเวลาการทำงาน ของหน่วยประมวลผลกลางมาก เช่น งานการพัฒนา และทดสอบการทำงานของโปรแกรม

ผู้ใช้งานในระบบแต่ละกลุ่มจะเกิดปัญหาแตกต่างกันไปในระบบ เช่น การจัดระบบแฟ้มข้อมูลให้มีค่าแบนวิดท์ (Bandwidth) หรืออัตราการส่งข้อมูลสูง ช่วยให้การทำงานของจานแม่เหล็กสำหรับโปรแกรมขนาดใหญ่ที่ต้องการใช้หน่วยอุปกรณ์รับเข้า และส่งออกมากทำงานดีขึ้น แต่โปรแกรมขนาดเล็กจะทำงานช้าลง หรือการกำหนดให้ผู้ใช้ส่งงานเข้าสู่ระบบแบบแบทช์ ทำให้งานที่ต้องการเวลาของหน่วยประมวลผลกลางมาก ทำงานเร็วขึ้น แต่ถ้าปัญหานั้นเกิดจากการเรียกใช้โปรแกรมการแก้ไขแฟ้มข้อมูล หรือการอ่านจดหมาย การส่งงานแบบแบทช์ก็ไม่ช่วยให้ระบบทำงานดีขึ้นแต่อย่างใด

ข้อบ่งชี้ความสามารถของระบบ ขึ้นกับความรู้สึกของผู้ใช้งานในระบบว่า งานนั้นใช้เวลาในการทำงานมากขึ้น ผู้ใช้งานเหล่านี้จะมีระดับของความยอมรับได้อยู่ระดับหนึ่ง ซึ่งระดับของความยอมรับได้นี้ไม่สามารถวัดได้เพราะเกี่ยวข้องกับบุคคลิกภาพ และความรู้สึกของผู้ใช้งานในระบบแต่ละคน

2.2 สมรรถนะการทำงานของระบบคอมพิวเตอร์

สมรรถนะการทำงานของระบบคอมพิวเตอร์คือการตรวจสอบการทำงานของระบบคอมพิวเตอร์ว่ามีการทำงานในระบบอย่างไรโดยพิจารณาถึงการใช้เวลา การตอบสนอง และความพร้อมที่จะให้บริการ

2.2.1 ปัญหาที่มีผลกระทบต่อสมรรถนะการทำงานของระบบคอมพิวเตอร์

2.2.1.1 ปัญหาทั่วไป

- หน่วยความจำสำรองไม่เพียงพอ
- การใช้งานของหน่วยความจำสำรองแต่ละส่วนไม่สมดุลย์กัน
- การร่วมกันใช้ทรัพยากรไม่สมดุลย์
- ไดเรคทอรี่ ใหญ่เกินไป
- โปรเซสสูญหาย
- งานที่เกี่ยวข้องกับข้อมูลจำนวนมาก

2.2.1.2 ปัญหาของผู้ใช้งาน

- การจัดตัวแปรพาธที่ซับซ้อน

2.2.1.3 ปัญหางานการบำรุงรักษา

- การจัดระบบแฟ้มข้อมูลไม่ดีพอ
- การขัดจังหวะของโมเด็มบ่อยเกินไป

2.2.1.4 ปัญหาการจัดระบบ

- ชนิดของระบบแฟ้มข้อมูล และพารามิเตอร์ของระบบ
- หน่วยความจำหลักไม่พอ
- จำนวนไอโหนดไม่พอ
- ขนาดของตารางควบคุมต่างๆ
- ขนาดของบัฟเฟอร์

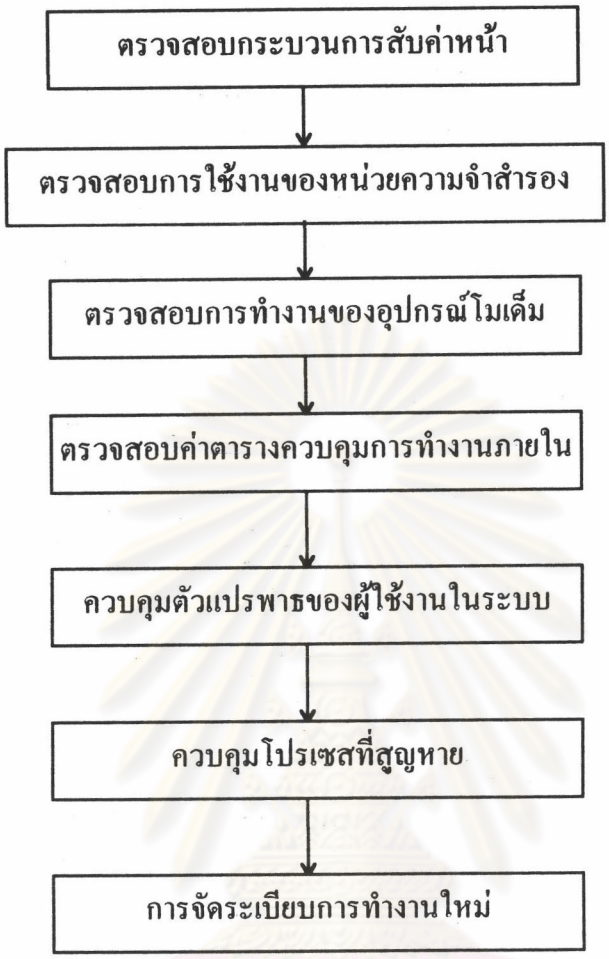
2.2.2 ขั้นตอนในการตรวจสอบการทำงานของระบบคอมพิวเตอร์

การระบุปัญหาการทำงานของระบบคอมพิวเตอร์ว่าเกิดจากส่วนใดนั้น มีวิธีการหรือขั้นตอนการทำงานแตกต่างกันไปในแต่ละระบบเครื่องคอมพิวเตอร์ โดยไม่มีขั้นตอนสำเร็จรูป (Canned procedure) ที่สามารถนำมาใช้ได้กับทุกระบบ ในระบบคอมพิวเตอร์ซึ่งใช้ระบบปฏิบัติการยูนิกซ์ ซิสเต็มไฟว์นั้น ขั้นตอนที่สามารถนำมาใช้เพื่อเป็นแนวทางในการตรวจสอบการทำงานดังผังงาน (System flowchart) รูปที่ 2.1 ดังนี้คือ

2.2.2.1 ตรวจสอบกระบวนการสับค่าหน้า โดยตรวจค่าเปอร์เซนต์การสับค่าหน้า ซึ่งคือตรวจค่าความยาวของแถวคอย และเปอร์เซนต์ของเวลาที่ใช้ในการทำงานของลำดับการทำงานปัจจุบัน รวมทั้งค่าการเปลี่ยนโปรเซสการทำงาน

2.2.2.2 ตรวจสอบการใช้งานของหน่วยความจำสำรอง โดยตรวจสอบการเกิดเหตุการณ์คอขวดของจานแม่เหล็ก พิจารณาว่าในขณะใดๆ หน่วยประมวลผลกลางมีการทำงานหรือไม่ กรณีที่มีการทำงานเป็นการทำงานในส่วนของโปรแกรมผู้ใช้งาน หรือส่วนโปรแกรมระบบ ถ้าไม่มีการทำงานเป็นการว่างเพื่อรอคอยการทำงานของส่วนอุปกรณ์รับเข้า และส่งออกหรือการว่างเพราะไม่มีโปรเซส การทำงาน

2.2.2.3 ตรวจสอบการทำงานของอุปกรณ์โมเด็ม เพื่อตรวจสอบการทำงานของสายอุปกรณ์



รูปที่ 2.1 การตรวจสอบการทำงานของระบบเครื่องคอมพิวเตอร์

2.2.2.4 ตรวจสอบค่าตารางควบคุมการทำงาน เพื่อดูสถานะของตารางควบคุมการทำงานต่างๆ ว่าเพียงพอสำหรับการใช้งานหรือไม่ เช่น ตารางควบคุมจำนวนโปรเซส หรือตารางควบคุมจำนวนแฟ้มข้อมูลในระบบ

2.2.2.5 ควบคุมตัวแปรพาธของผู้ใช้งานในระบบ โดยให้ความรู้แก่ผู้ใช้งานในระบบว่า การกำหนดค่าตัวแปรพาธนี้มีความสำคัญ และมีผลต่อระบบอย่างไร รวมทั้งแนะนำวิธีการกำหนดที่ถูกต้องให้แก่ผู้ใช้งานในระบบด้วย

ข้อควรพิจารณาในการกำหนดตัวแปรพาธ

1) ควรกำหนดตัวแปรพาธให้สามารถค้นหาคำสั่งที่ต้องการ พบได้ในครั้งแรกของการทำงาน เพราะการค้นหาคำสั่งจะค้นหาจากซ้ายไปขวา

2) แต่ละไดเรคทอรี่ควรกำหนดให้มีการค้นหาเพียงครั้งเดียว โดยอย่ากำหนดไดเรคทอรี่ซ้ำซ้อน เพราะเป็นการทำงานเพิ่มโดยไม่จำเป็น

3) ไม่ควรกำหนดให้ ไดเรคทอรี่ปัจจุบันที่ทำงาน เป็นไดเรคทอรี่แรกในตัวแปรพาธ เช่น PATH=.:/usr/bin

4) ข้อมูลของตัวแปรพาธไม่ควรให้มีจำนวนของไดเรคทอรี่ในรายการค้นหามากเกินไป

5) แต่ละไดเรคทอรี่ในตัวแปรพาธไม่ควรเป็นไดเรคทอรี่ที่ใหญ่เกินไป ในกรณีที่จำเป็นควรกำหนดให้ไดเรคทอรี่ที่ใหญ่ เป็นไดเรคทอรี่สุดท้ายของตัวแปรพาธนั้น

6) ไดเรคทอรี่ซึ่งมีการเชื่อมโยง (symbolic link) กับไดเรคทอรี่อื่นไม่ควรกำหนดเป็นส่วนหนึ่งของตัวแปรพาธ

2.2.2.6 การควบคุมโปรเซสที่สูญหาย เพื่อแสดงสถานะของโปรเซสที่กำลังทำงาน เวลาที่โปรเซสเริ่มทำงาน และเวลาที่โปรเซสนั้นใช้ในการทำงาน (หน่วยเป็นนาที/วินาที) ซึ่งถ้า โปรเซสที่ใช้เวลาในการทำงานนานมาก โดยโปรเซสนั้นยังไม่เสร็จสมบูรณ์ ต้องตรวจสอบกับเจ้าของโปรเซสว่าเกิดอะไรขึ้น โปรเซสเหล่านี้อาจเป็นโปรเซสที่ควบคุมไม่ได้ และยังคงอาศัยทรัพยากรต่างๆ ในระบบ ซึ่งอาจเป็นสาเหตุให้เกิดความเสียหายขึ้นในระบบได้

2.2.2.7 การจัดระเบียบการทำงานใหม่ โดยการตรวจสอบสถานะของแต่ละโปรเซสว่า มีงานใดบ้างที่ใช้งานทรัพยากรระบบมาก และทำความเข้าใจกับเจ้าของโปรเซส โดยให้นำโปรแกรมงานที่มีขนาดใหญ่ หรือที่มีการใช้ข้อมูลมากมาทำในเวลาที่มีงานในระบบน้อย เช่น ในตอนกลางคืนหรือวันหยุด หรืออาจลดลำดับความสำคัญของงาน (lower piority) เป็นต้น

2.3 คุณลักษณะของระบบปฏิบัติการยูนิกซ์

เนื่องจากการวิจัยนี้ อาศัยการทำงานของโปรแกรมระบบปฏิบัติการยูนิกซ์ ซิสเต็มไฟว์ เป็นกรณีศึกษา ดังนั้นเพื่อความเข้าใจการทำงานของระบบคอมพิวเตอร์ จึงขอกล่าวถึงคุณลักษณะและการทำงานของระบบปฏิบัติการยูนิกซ์

2.3.1 ประวัติและความเป็นมาของระบบปฏิบัติการยูนิกซ์

ระบบปฏิบัติการยูนิกซ์ เกิดขึ้นจากการทำงานร่วมกัน ของนักวิทยาศาสตร์ในโครงการพัฒนาระบบปฏิบัติการใหม่เรียกว่าโครงการมัลติคส์ (Multics) ซึ่งเป็นความร่วมมือของหน่วยงาน 3 แห่ง คือสถาบันเทคโนโลยีแมสสาจูเซสส์หรือ MIT (Messachusetts Institute of Technology) สถาบันวิจัยของบริษัทเอทีแอนด์ที (AT&T Bell Labs) และบริษัทจีอี (General Electric Co.) นักวิจัยเหล่านี้ได้ออกแบบระบบปฏิบัติการเพื่อใช้กับเครื่องคอมพิวเตอร์เฉพาะด้านเรียกว่าเครื่องจีอี 635 (GE 635) ซึ่งเป็นเครื่องขนาดใหญ่และนำสมัยในขณะนั้น ประมาณปี ค.ศ. 1965-1969 ระบบปฏิบัติการใหม่นี้คาดว่าจะได้รับการออกแบบให้มีความยืดหยุ่นและใช้งานง่าย

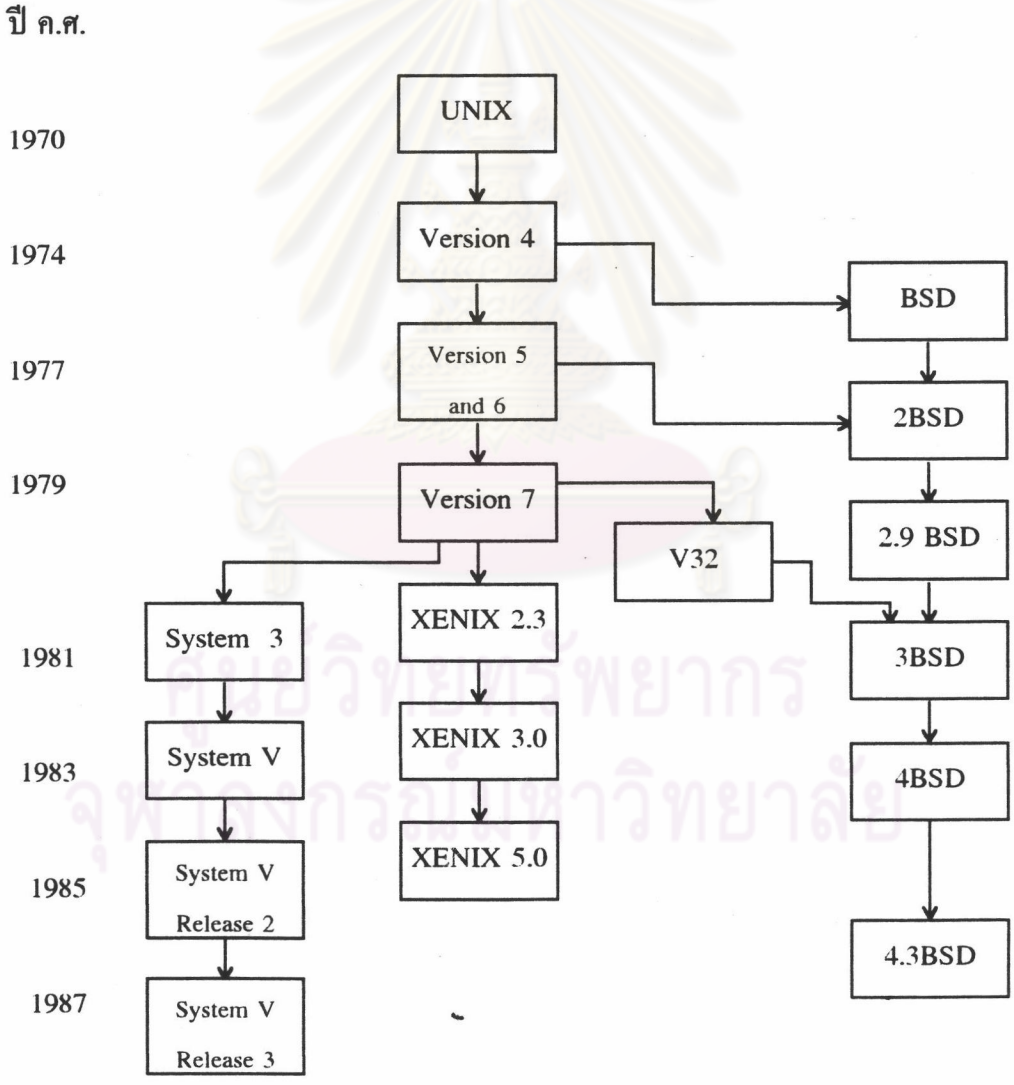
ในขณะเดียวกันสถาปัตยกรรมต่างๆ ต้องมีระบบการรักษาความปลอดภัย ระบบแฟ้มข้อมูลที่ง่าย และมีคุณสมบัติของการใช้งานร่วมกัน

อย่างไรก็ตามการพัฒนาระบบมัลติคส์นี้ เกิดความล่าช้าขึ้นจนไม่สามารถสำเร็จลงตามเวลาได้ และสถาบันวิจัยของบริษัทเอทีแอนด์ทีได้ถอนตัวออกจากโครงการนี้ ในปีค.ศ. 1969 นักวิจัยของบริษัท 2 คนซึ่งอยู่ในทีมพัฒนาระบบ คือนายเคน ทอมสัน และนายเดนนิส ริชชี (Ken Thomson and Dennis Ritchie) ได้เขียนโปรแกรมเกมส์การท่องเที่ยวอวกาศ (Space Travel) บนเครื่องจีอี 635 ซึ่งใช้ระบบปฏิบัติการมัลติคส์ เมื่อบริษัทถอนโครงการออกนักวิจัยทั้งสอง จึงต้องปรับเปลี่ยนโปรแกรมนี้ใหม่เพื่อให้สามารถทำงานได้ บนระบบเครื่องคอมพิวเตอร์พีดีพี 7 (PDP-7:Program Data Processor) ซึ่งเป็นเครื่องคอมพิวเตอร์ขนาดกลาง (Minicomputer) ที่ใช้งานอยู่ ทั้งสองได้พบกับปัญหามากมายอันเนื่องมาจากสภาพแวดล้อมของระบบเครื่องพีดีพี 7 ดังนั้นในการทำงานจึงต้องนำงานวิจัยทั้งหมดที่พัฒนาขึ้นมาทำงานบนระบบเครื่องจีอี 635 ใช้ระบบปฏิบัติการจีโกส์ (GECOS) ซึ่งเป็นระบบปฏิบัติการแบบแบทช์ แล้วนำโปรแกรมที่ผ่านการแปลมาใช้ทำงานบนระบบเครื่องพีดีพี 7 ซึ่งใช้เทปกระดาษ ความยุ่งยากเหล่านี้ผลักดันให้นักวิจัยเคน ทอมสัน สร้างระบบแฟ้มข้อมูลที่ง่าย และสร้างโปรแกรมอรรถประโยชน์บางอย่างเพื่อใช้งานบนเครื่องพีดีพี 7 ซึ่งในปีต่อมา ค.ศ. 1970 นายไบรอัน คันนิ่งแฮน (Brian Kernighan) ได้เริ่มเรียกระบบนี้ว่า ระบบยูนิกซ์ (UNIX) ความหมายของชื่อนี้นั้นไม่มีการเปิดเผยถึงที่มาแต่อาจพูดได้ว่าเป็นการเลียนเสียงของคำว่า มัลติคส์

ระบบยูนิกซ์ในยุคเริ่มแรกนั้น ยังคงต้องอาศัยการเอาใจใส่ดูแล จนกระทั่งเมื่อทีมงานซึ่งนำโดยนักวิจัยทั้งสอง ได้รับเครื่องรุ่นใหม่ที่ทันสมัยกว่าคือเครื่องพีดีพี 11 โดยการแลกเปลี่ยนกับการพัฒนาระบบการประมวลเอกสารเพิ่มเข้าในระบบ ในช่วงนี้สถาบันวิจัยของบริษัทเอทีแอนด์ที ได้นำโครงการการพัฒนาระบบยูนิกซ์นี้เป็นโครงการของบริษัทอย่างเป็นทางการ ระบบยูนิกซ์ในรุ่นที่ 2 ได้รับการติดตั้งโดยมีการทำงานในส่วนของการประมวลเอกสาร ซึ่งองค์กรที่ได้จดลิขสิทธิ์กับบริษัท ได้นำมาใช้ในการทำงานอย่างกว้างขวาง

ในปี ค.ศ. 1973 ระบบปฏิบัติการยูนิกซ์ถูกนำมาทำใหม่ ใช้ภาษาระดับสูงคือภาษาซี โดยปรับปรุงในส่วนการบำรุงรักษาให้ง่ายขึ้น และเพิ่มคุณสมบัติการนำไปใช้ได้ ในระบบเครื่องคอมพิวเตอร์อื่น ทำให้นักวิจัยในบริษัทเริ่มใช้งานมากขึ้น โดยในช่วงนี้เรียกว่ายูนิกซ์รุ่นที่ 4 ขณะเดียวกันตามมหาวิทยาลัยเริ่มใช้ระบบยูนิกซ์ในการสอนวิชาการออกแบบระบบปฏิบัติการ มีผลทำให้ระบบยูนิกซ์เริ่มมีการใช้งานอย่างกว้างขวาง ระบบปฏิบัติการยูนิกซ์นี้แตกต่างจากระบบปฏิบัติการอื่นในขณะนั้นเพราะเป็นระบบปฏิบัติการที่เล็กกระทัดรัด และถูกพัฒนาขึ้นด้วยภาษาระดับสูง โดยรวมแนวคิดใหม่ไว้ และไม่ถูกเก็บเป็นความลับ ราวปี ค.ศ. 1977 ยูนิกซ์ รุ่นที่ 5 และ รุ่นที่ 6

ถูกนำมาใช้งาน ซึ่งรุ่นนี้ผู้ใช้งานในระบบสามารถเพิ่มโปรแกรมอรรถประโยชน์ตามที่ต้องการได้ โปรแกรมอรรถประโยชน์เหล่านี้ในเวลาต่อมาถือเป็นส่วนประกอบหนึ่งของระบบ อย่างไรก็ตาม สถาบันวิจัยของบริษัทเอทีแอนด์ทีก็ยังมิได้ประกาศรับรองอย่างเป็นทางการ เพราะในปี ค.ศ. 1956 มีข้อบังคับให้บริษัททำธุรกิจได้เฉพาะในส่วนที่ได้รับสิทธิบัตรรับรอง คือกิจการโทรคมนาคม ดังนั้นจึงไม่สามารถดำเนินธุรกิจการขายระบบคอมพิวเตอร์ได้ แต่ในขณะเดียวกันระบบยูนิกซ์ยังคงได้รับการพัฒนาอยู่ตลอดเวลา มีการเพิ่มโปรแกรมอรรถประโยชน์เพื่อช่วยในการทำงานมากยิ่งขึ้น ทำให้ระบบยูนิกซ์ รุ่นที่เป็นทางการรุ่นแรก ที่เรียกว่า ยูนิกซ์รุ่นที่ 7 มีความสมบูรณ์พร้อมมากในขณะนั้น



รูปที่ 2.2 วิวัฒนาการของการพัฒนาระบบปฏิบัติการยูนิกซ์

ยูนิกซ์รุ่นที่ 7 นี้ได้รับการรับรองอย่างเป็นทางการต่อสาธารณชนเมื่อปี ค.ศ. 1978 และเป็นต้นแบบของระบบยูนิกซ์ที่มีอยู่มากมายในปัจจุบัน จากรุ่นที่ 7 การพัฒนาและปรับปรุงระบบซึ่งสามารถแยกเป็นสายหลักได้ 3 สายคือระบบยูนิกซ์ซิสเต็มไฟว์ของบริบัทเอทีแอนด์ที ระบบปฏิบัติการซีนิกซ์ (XENIX) ของบริษัทไมโครซอฟต์ (Microsoft Corporation) และระบบปฏิบัติการยูนิกซ์ของมหาวิทยาลัยแคลิฟอร์เนียร์ วิทยาเขตเบิร์คเลย์ (University of California , Berkely Campus) หรือเรียกว่ายูนิกซ์บีเอสดี (Berkeley Software Distribution) ดังรูป ที่ 2.2

ปี 1987 เกิดความร่วมมือในการพัฒนาระบบยูนิกซ์จากทั้ง 3 สาย เป็นระบบยูนิกซ์รุ่นใหม่เรียกว่า ระบบยูนิกซ์ ซิสเต็มไฟว์ รุ่นที่ 4 (SVR4) ซึ่งเป็นรุ่นล่าสุด โดยผนวกการทำงานจากทั้ง 3 สายเข้าด้วยกัน

ผู้ที่ศึกษาเกี่ยวกับระบบยูนิกซ์ในระยะเริ่มแรกมักจะสงสัยเกี่ยวกับรุ่นต่างๆ ของยูนิกซ์ที่ข้ามหายไป เช่น รุ่นซิสเต็ม 1 รุ่นซิสเต็ม 2 และรุ่นซิสเต็ม 4 หรือยูนิกซ์บีเอสดี รุ่น 1.0 เป็นต้น มีระบบยูนิกซ์มากมายที่ถูกพัฒนาขึ้นรวมทั้งรุ่นที่พัฒนาโดยบริษัทผู้ขายระบบคอมพิวเตอร์อื่น แต่มิได้ นำออกเผยแพร่สู่สาธารณะชน

2.3.2 ฟังก์ชันการทำงานของระบบปฏิบัติการยูนิกซ์

ระบบปฏิบัติการ คือกลุ่มของโปรแกรมที่คอยประสานการทำงานระหว่างส่วนฮาร์ดแวร์ และซอฟต์แวร์ และการที่ระบบคอมพิวเตอร์ มักมีจำนวนของฮาร์ดแวร์จำกัด จึงเป็นหน้าที่ของระบบปฏิบัติการที่จะจัดสรรการใช้งานของทรัพยากรต่างๆ ในระบบอย่างมีประสิทธิภาพ ซึ่งระบบปฏิบัติการยูนิกซ์ ได้จัดเตรียมไว้อย่างครบถ้วน ดังรูปที่ 2.3

2.3.2.1 การทำงานของส่วนอุปกรณ์รับเข้า และส่งออก

เป็นส่วนสำคัญสำหรับการปฏิบัติงานของระบบคอมพิวเตอร์ กล่าวคือเป็นส่วนที่จัดเก็บหรือเรียกใช้ข้อมูลที่เก็บอยู่ในสื่อบันทึกเช่น จานแม่เหล็ก เทป การติดต่อสื่อสารผ่านทางจอภาพกับผู้ใช้งานในระบบ หรือการพิมพ์ผลลัพธ์บนกระดาษต่อเนื่อง ระบบปฏิบัติการทุกระบบ จะต้องมีการจัดเตรียมการทำงานในส่วนนี้

2.3.2.2 การแปลคำสั่งในการทำงาน

ส่วนนี้มีหน้าที่ในการอ่านคำสั่งที่ผู้ใช้งานพิมพ์ผ่านแป้นพิมพ์แล้วนำไปเปลี่ยนแปลง หรือแปลเป็นคำสั่งให้ระบบเครื่องคอมพิวเตอร์เข้าใจได้ เพื่อให้ระบบสามารถปฏิบัติงานตามคำสั่งนั้นซึ่งจะแตกต่างในแต่ละระบบเครื่อง

รูปที่ 2.3 ฟังก์ชันการทำงานของระบบปฏิบัติการยูนิกซ์

2.3.2.3 การจัดการข้อมูล

คือการที่ผู้ใช้งานในระบบแต่ละคนสามารถจัดกลุ่มข้อมูลในลักษณะของแฟ้มข้อมูล (Logical files) โดยที่ไม่ต้องกังวลถึงลักษณะการจัดเก็บจริงของระบบ (Physical files) กรณีที่ระบบคอมพิวเตอร์ที่ไม่มีการจัดเตรียมการจัดการข้อมูลส่วนนี้ มักจะมีข้อจำกัดในด้านความยืดหยุ่นในการทำงาน และการใช้ประโยชน์

2.3.2.4 เครื่องมือในการพัฒนาโปรแกรม

คือเครื่องมือที่ใช้ในการเขียน และพัฒนาโปรแกรม เช่น ตัวแปลโปรแกรม ชุดคำสั่งเพื่อตรวจแก้จุดบกพร่อง ตัวแปลภาษาแอสเซมบลี (Assembler) และระบบการบำรุงรักษาซอฟต์แวร์

2.3.2.5 ระบบแบ่งเวลา

เป็นระบบที่ยอมให้ มีผู้ใช้งานได้หลายคนในขณะเดียวกัน โดยทำงานผ่านเครื่องปลายทางที่ต่อเชื่อมกับระบบคอมพิวเตอร์ ณ จุดต่างๆ การทำงานลักษณะนี้มักพบเฉพาะในระบบขนาดใหญ่

2.3.2.6 ระบบการรักษาความปลอดภัย

เป็นระบบที่ช่วยป้องกันข้อมูลของผู้ใช้งาน จากผู้ใช้อื่นที่มิใช่เจ้าของที่แท้จริง และป้องกันความเสียหายที่อาจเกิดขึ้นของระบบปฏิบัติการจากการทำงานจากโปรแกรมผู้ใช้งาน หน้าที่หลักคือ การอนุญาตให้เฉพาะผู้ที่ได้รับการมอบหมายสามารถเรียกใช้งานในระบบเฉพาะในส่วนที่ได้รับการมอบหมายเท่านั้น โดยทั่วไปในระบบคอมพิวเตอร์ที่ไม่มีการทำงานแบบระบบแบ่งเวลา จะมีคุณสมบัติของการรักษาความปลอดภัยน้อย หรือไม่มีเลย ทั้งนี้เพราะเกี่ยวข้องกับผู้ใช้งานเพียงหนึ่ง หรือสองคน แต่ในระบบคอมพิวเตอร์ขนาดใหญ่ซึ่งมีผู้ใช้งานในระบบมากต้องมีระบบการรักษาความปลอดภัยอย่างใดอย่างหนึ่งแตกต่างกันไปในเครื่องแต่ละแบบ

2.3.2.7 การสื่อสารข้อมูล

คือความสามารถในการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์เพื่อการส่งผ่านข้อมูลหรือโปรแกรม โดยมีกฎเกณฑ์และขั้นตอนที่ได้กำหนดไว้แล้ว

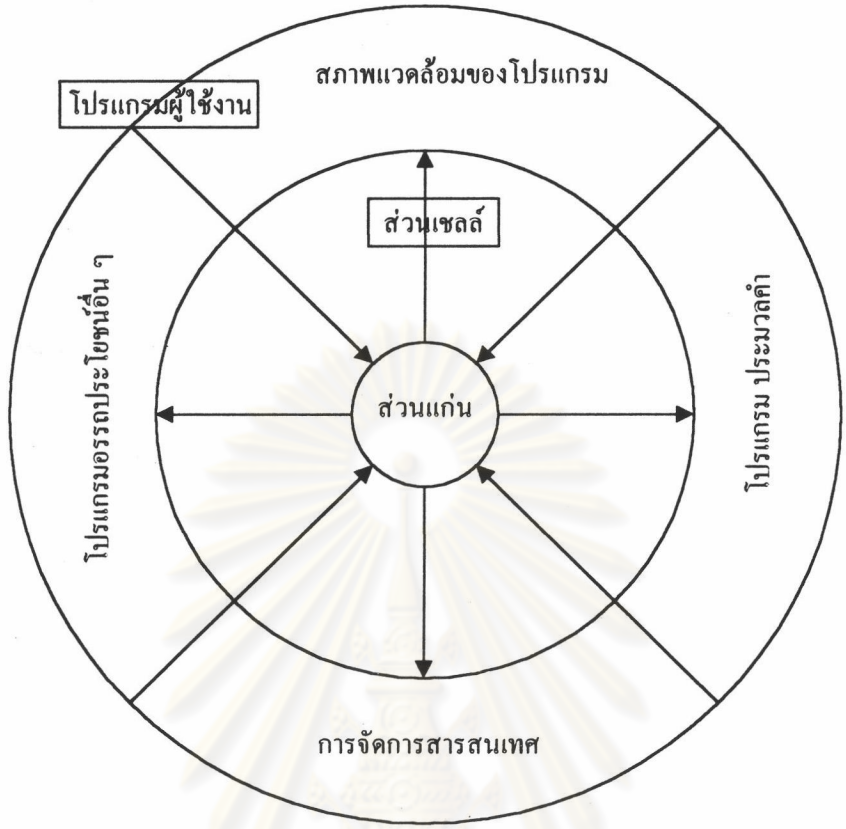
2.3.2.8 การทำบัญชีระบบ

คือการเฝ้าติดตามการทำงานของผู้ใช้งานแต่ละคนในระบบว่า มีการทำงานอะไรบ้าง โดยมีเป้าหมายเพื่อการเรียกเก็บเงินตามบัญชีการใช้งานนี้ ซึ่งจำเป็นในระบบที่มีผู้ใช้งานหลายคน และคิดเงินตามเวลาที่ใช้เครื่อง

2.3.3 ส่วนประกอบของระบบปฏิบัติการยูนิกซ์

ระบบปฏิบัติการยูนิกซ์ ประกอบด้วย ส่วนโปรแกรมระบบปฏิบัติการ ชุดของโปรแกรมอรรถประโยชน์ ส่วนโปรแกรมช่วยงานเฉพาะด้านซึ่งให้บริการในระบบแบ่งเวลา ทำให้สามารถทำงานได้หลายงาน (multitasking) และระบบแฟ้มข้อมูล การที่ระบบนี้แตกต่างจากระบบอื่น คือ การแบ่งส่วนการทำงานเหล่านี้อย่างชัดเจน โดยเรียกว่าส่วนแก่น (kernel) ส่วนเชลล์ (shell) และส่วนโปรแกรมงาน (user program) ดังรูป 2.4 วงกลมแต่ละวงแทนส่วนประกอบของระบบ คือส่วนแก่น ส่วนเชลล์ และส่วนคำสั่งที่ใช้งาน ลูกศรแสดงการทำงานของเชลล์ในการเป็นสื่อกลางระหว่างผู้ใช้งาน และส่วนแก่น

การทำงานของโปรแกรมอรรถประโยชน์เป็นการลดภาระการทำงานของส่วนแก่น ทำให้ขนาดของส่วนแก่นเล็กลง คำสั่งของระบบยูนิกซ์ทำงานแยกโดยอิสระจากระบบปฏิบัติการ และระบบอาศัยการทำงานของโปรแกรมอรรถประโยชน์ มากกว่าการทำงานจากส่วนแก่น มีผลให้สามารถปรับเปลี่ยนระบบเพื่อรองรับการทำงานของโปรแกรมงานต่างๆ โดยง่าย



รูปที่ 2.4 ส่วนประกอบของระบบปฏิบัติการยูนิกซ์

ระบบปฏิบัติการยูนิกซ์ อาจเรียกได้ว่าเป็นระบบปฏิบัติการแรก ที่ทำให้ผู้ทำงานด้านการพัฒนาระบบงานสามารถเขียนโปรแกรมได้ โดยไม่ต้องกังวลเกี่ยวกับระบบอุปกรณ์ (machine dependence) นอกจากนั้นโปรแกรมที่เขียนขึ้นนี้ สามารถนำไปใช้ได้บนระบบเครื่องคอมพิวเตอร์อื่นโดยมีการแก้ไขน้อยที่สุด ทำให้เกิดคุณสมบัติการข้ามระบบ คือสามารถใช้ร่วมกันได้ในระหว่างเครื่องที่ใช้ระบบปฏิบัติการยูนิกซ์เหมือนกัน เป็นผลดีทั้งสำหรับผู้พัฒนาระบบและผู้ใช้งานในระบบ เพราะผู้พัฒนาระบบสามารถเสนองานบนระบบเครื่องได้หลายแบบ เป็นการลดค่าใช้จ่ายในการด้านการทำโปรแกรมใหม่และการแปลงผัน (Conversion) สำหรับในด้านผู้ใช้งานในระบบก็สามารถซื้อระบบงานเหล่านั้นได้โดยไม่ต้องกังวลว่าระบบเครื่องคอมพิวเตอร์จะล้าสมัย

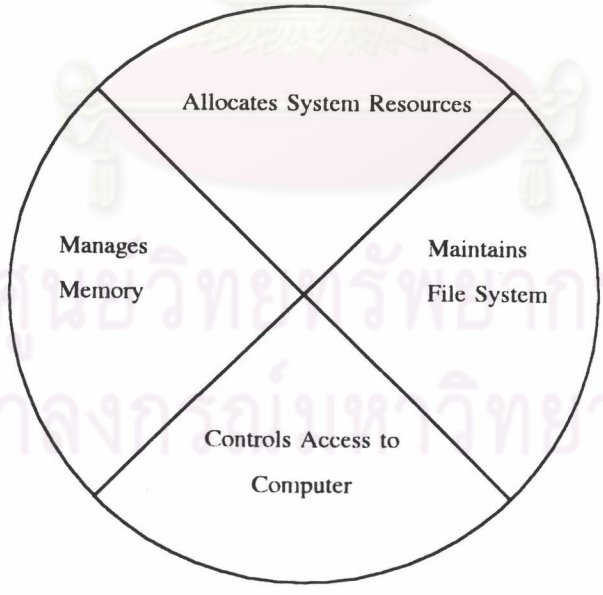
ความสามารถอีกด้านหนึ่งของระบบปฏิบัติการยูนิกซ์คือ การออกแบบส่วนการทำงานของโปรแกรมอรรถประโยชน์ให้สามารถช่วยทำงานได้โดยที่ผู้พัฒนาโปรแกรมอื่น ไม่ต้องนำมาเขียนเพิ่มไว้ในโปรแกรม โปรแกรมอรรถประโยชน์เหล่านี้จะถูกออกแบบให้ทำงานเฉพาะอย่างและสามารถใช้ช่วยกำหนดสภาวะแวดล้อมในการทำงานอย่างมีประสิทธิภาพ นอกจากนั้นระบบปฏิบัติการยูนิกซ์ยังทำหน้าที่เป็นผู้จัดการระบบ คือจัดสรรการทำงานของทรัพยากรระบบ อันได้

แก่การทำงานของตัวขับอุปกรณ์ เทป แฟ้มข้อมูล เครื่องพิมพ์ และหน่วยความจำหลักตามแต่การเรียกใช้งานของโปรแกรม โดยที่ผู้ใช้งานไม่ต้องเกี่ยวข้องกับการทำงานจริงของอุปกรณ์เหล่านั้น

นอกจากนั้นระบบปฏิบัติการยูนิกซ์ เป็นระบบที่เปิดทางของความเป็นมาตรฐานเดียวกัน และความสามารถในการนำไปใช้ได้ กับอุปกรณ์พ่วงต่อต่างชนิดกัน ช่วยลดความกังวลของผู้พัฒนาและผู้ใช้งานในระบบเกี่ยวกับค่าใช้จ่ายในการแปลงผัน เมื่อมีความจำเป็นต้องเปลี่ยนระบบเครื่องคอมพิวเตอร์ใหม่ นอกจากนั้นระบบยูนิกซ์เป็นระบบแบ่งเวลาระหว่างหลายผู้ใช้ คือสามารถมีผู้ใช้งานในระบบได้หลายคนโดยที่แต่ละคนจะรู้สึกว่าเป็นเจ้าของเครื่องคอมพิวเตอร์ และเป็นระบบที่ทำงานได้ครั้งละหลายงาน คุณสมบัตินี้เปิดโอกาสให้ผู้ใช้งานในระบบทำงานได้หลายงานในขณะเดียวกันโดยอาจใช้การกำหนดค่าบุริมภาพ คือการจัดลำดับความสำคัญของงาน หรือการทำงานแบบฉากหลัง

2.3.3.1 ส่วนแก่น หรือส่วนใจกลาง (The Kernel)

หัวใจของระบบของระบบปฏิบัติการยูนิกซ์คือส่วนแก่น มีหน้าที่รับผิดชอบการทำงานด้านปฏิบัติการ คือการจัดลำดับงาน การควบคุมการใช้งาน การติดตามการทำงานของแฟ้มข้อมูล การจัดการหน่วยความจำหลัก การควบคุมการทำงานของอุปกรณ์พ่วงต่อ รวมทั้งการจัดสรร ทรัพยากรระบบต่างๆ เพื่อการใช้งานร่วมกันของผู้ใช้ในระบบ ดูรูป 2.5



รูปที่ 2.5 หน้าที่ของส่วนแก่น

หน้าที่หลักของส่วนแก่นคือการคัดเลือกงานที่มีลำดับความสำคัญสูงสุด และอยู่ในสภาวะพร้อมที่จะทำงาน และกำหนดค่าสำหรับใช้ในการทำงาน ระบบปฏิบัติการยูนิกซ์สามารถเปลี่ยนแปลงค่าลำดับความสำคัญได้โดยจะกำหนดให้งานที่เกี่ยวกับการคำนวณมีค่าต่ำที่สุด และงานที่มีการโต้ตอบมีความสำคัญมากที่สุด ทำให้งานนี้มีการทำงานอย่างรวดเร็ว งานที่ถูกคัดเลือกโดยส่วนแก่นนี้ จะยังคงอยู่ทำงานในหน่วยความจำหลักไปเรื่อยๆ จนกว่าจะมีเหตุการณ์ดังต่อไปนี้เกิดขึ้น ส่วนแก่นจึงเปลี่ยนโปรเซสการทำงาน เพื่อเปิดโอกาสให้งานอื่นเข้าใช้งานในซีพียูต่อไป

- ครบรอบการใช้เวลาของงาน คือส่วนแก่นจะกำหนดให้แต่ละงานทำงานในช่วงเวลาหนึ่งเพื่อให้ทุกงานในระบบมีโอกาสเท่ากันในการเข้าใช้งานในซีพียู เมื่อครบเวลาแต่งานนั้นยังไม่เสร็จสมบูรณ์ งานนั้นจะกลับไปสู่สภาวะพร้อม เพื่อรอโอกาสในการทำงานครั้งต่อไป
- งานซึ่งต้องรอคอยเหตุการณ์บางอย่างจึงจะสามารถทำงานต่อไปได้ เช่น การรอคอยการอ่าน หรือการบันทึกข้อมูลในจานแม่เหล็ก ต้องรอคอยจนกว่าข้อมูลที่ต้องการถูกส่งมาให้จึงจะทำงานต่อไปได้
- มีเหตุการณ์ ซึ่งมีลำดับความสำคัญมากกว่าเกิดขึ้น ทำให้งานที่กำลังทำอยู่ ต้องถูกบังคับให้ปลดปล่อยทรัพยากรทั้งหลายที่ครอบครองอยู่ เพื่อเปิดโอกาสให้งานนั้นได้เข้าใช้งานในหน่วยประมวล ผลกลางต่อไป

2.3.3.2 ส่วนเชลล์

เชลล์เป็นส่วนติดต่อกับโปรแกรมผู้ใช้งาน แบ่งได้เป็น 3 แบบ คือ บอร์นเชลล์ (sh:Bourne shell) ซีเชลล์ (csh:C shell) และคอนเชลล์ (ksh:Korn shell) เชลล์เป็นตัวแทนติดต่อกับระบบปฏิบัติการ เพื่อให้ผู้ใช้งานในระบบทุกคนมีส่วนแบ่งในการใช้งานทรัพยากรระบบต่างๆ เช่น เวลาของตัวประมวลผล หน่วยความจำหลัก การเรียกใช้อุปกรณ์พ่วงต่อในระบบ เป็นต้น ส่วนเชลล์มีหน้าที่ดังต่อไปนี้

- แสดงความพร้อมที่จะทำงาน (The shell prompt) คือเป็นเครื่องหมายที่แสดงความพร้อมที่จะรับคำสั่งจากผู้ใช้งานในระบบ โดยใช้เครื่องหมายดอลลาร์ ($) แทนบอร์นเชลล์และคอนเชลล์ และเครื่องหมายเปอร์เซนต์ (%) แทนซีเชลล์ เมื่อปรากฏเครื่องหมายนี้ ผู้ใช้งานสามารถเรียกใช้คำสั่งได้ตามต้องการ
- แปลคำสั่งในการทำงาน และทำงานตามผลลัพธ์ที่ได้ (Command Interpretter) เป็นตัวบอกให้ส่วนแก่นทำการบรรจุคำสั่งและทำงานตามคำสั่งนั้น คำสั่งแบ่งออกเป็น 3 แบบ คือ แบบง่าย (Simple Commands) แบบสายท่อ (Pipeline Commands) และแบบรวม (Common List) คือนำคำสั่งมาเรียงต่อกันแล้วทำงานตามลำดับ หรือตามเงื่อนไขที่เกิดขึ้น

- ให้คำอธิบายกรณีเกิดความผิดพลาด กรณีที่ผู้ใช้งานใช้คำสั่งไม่ถูกต้อง หรือใช้คำสั่งที่เป็นต้องใช้สิทธิพิเศษ เชลล์จะแสดงคำอธิบายความผิดพลาดนั้น

- การทำงานแบบฉากหลัง เป็นส่วนหนึ่งของระบบปฏิบัติการที่ทำงานได้หลายงานในขณะเดียวกัน งานที่เหมาะสมสำหรับการทำงานแบบฉากหลัง คืองานที่ไม่ต้องการการโต้ตอบ เช่น การแปลโปรแกรม การจัดรูปแบบของเอกสาร การพิมพ์แฟ้มข้อมูล

- การเรียกใช้แฟ้มข้อมูลจากบางส่วนของชื่อแฟ้มข้อมูล (File Name Generator) คือ สามารถใช้การแทนค่าชื่อแฟ้มข้อมูล (wildcard) แทนการเรียกชื่อแฟ้มข้อมูลทุกครั้งที่ต้องการ โดยเชลล์จะสร้างชื่อแฟ้มข้อมูลที่สมบูรณ์จากรูปแบบที่ได้รับ

2.3.3.3 ระบบแฟ้มข้อมูล (The File System)

ฟังก์ชันในยุคแรกของการทำงานของระบบปฏิบัติการยูนิกซ์คือ ฟังก์ชั่นพื่อสนับสนุนการทำงานของระบบแฟ้มข้อมูล ซึ่งการทำงานในระบบปฏิบัติการยูนิกซ์ โปรแกรมงานและโปรแกรมอรรถประโยชน์ทุกโปรแกรมรวมทั้งข้อมูล ถูกจัดเก็บไว้ในลักษณะของแฟ้มข้อมูล ซึ่งมีโครงสร้างรูปต้นไม้กลับหัว ระบบแฟ้มข้อมูลแบ่งออกได้เป็น

- แฟ้มข้อมูลทั่วไป เก็บข้อมูลของระบบ โดยไม่มีการระบุรูปแบบ แฟ้มข้อมูลนี้เปรียบเสมือนใบไม้ ของต้นไม้

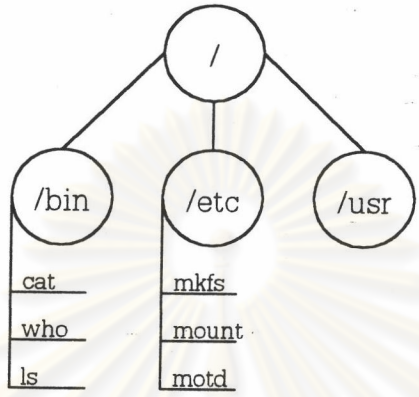
- แฟ้มข้อมูลไดเรคทอรี่ เป็นที่เก็บแฟ้มข้อมูลต่างๆ ไว้รวมกัน เปรียบเสมือนกับกิ่งไม้ ซึ่งเก็บรวบรวมใบไม้ ของต้นไม้

- แฟ้มข้อมูลไฟโฟ (FIFO:First in,First out Files) เป็นการอนุญาตให้ มีการติดต่อกันระหว่างโปรเซสที่ไม่เกี่ยวข้องกัน แฟ้มข้อมูลแบบนี้ มักใช้ในกรณีที่การติดต่อเป็นแบบทางเดียว หรือกรณีที่โปรเซสหลายโปรเซสต้องการติดต่อกับโปรเซสเดียว ซึ่งเรียกว่า โปรเซสดีมอน (Daemon Process) ซึ่งมีประโยชน์ในการบำรุงรักษาระบบ เพราะสามารถตั้งกำหนดเวลาในการทำงานได้โดยอัตโนมัติ

- แฟ้มข้อมูลพิเศษ เป็นแฟ้มข้อมูลที่ใช้แทนอุปกรณ์พ่วงต่อในระบบ เมื่อมีโปรเซสใดเรียกการทำงานกับแฟ้มข้อมูลพิเศษนี้ ทั้งการอ่านและเขียน ข้อมูลนั้นจะถูกส่งให้กับอุปกรณ์ที่มีการเรียกใช้นั้นทันที เรียกได้ว่าเป็นแฟ้มข้อมูลที่ทำหน้าที่ เป็นตัวชี้ไปยังตัวขับอุปกรณ์ซึ่งอยู่ในส่วนแก่น

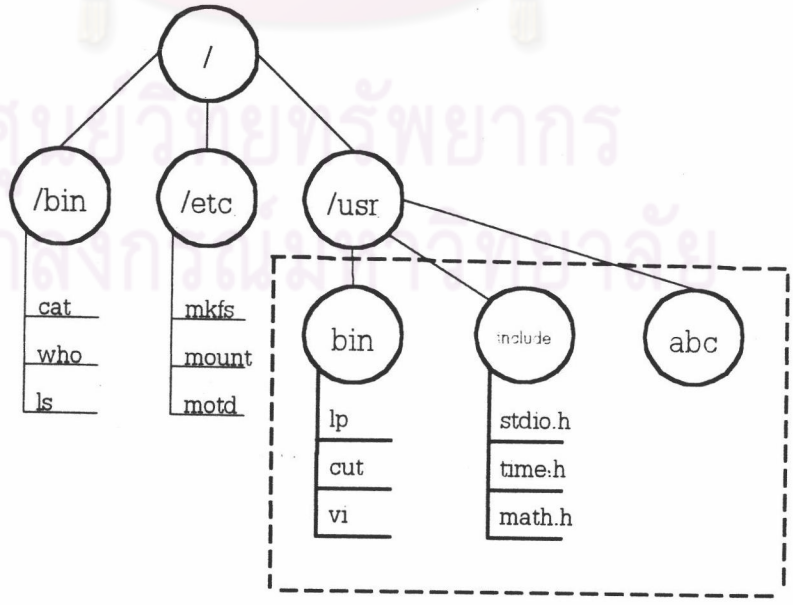
โครงสร้างของระบบแฟ้มข้อมูลแบบนี้ ทำให้แยกอรรถประโยชน์ โปรแกรมผู้ใช้งาน และข้อมูลออกจากกันได้อย่างชัดเจน แฟ้มข้อมูลเหล่านี้สามารถใช้งานร่วมกันในกลุ่มผู้ใช้งานเดียวกัน และสามารถป้องกันการเรียกใช้งานจากผู้ใช้งานอื่น การประกอบกันของแฟ้มข้อมูล และ

ไดเรคทอรี่นี้เองทำให้เกิด ระบบแฟ้มข้อมูลขึ้น จุดเริ่มต้นของระบบแฟ้มข้อมูลของยูนิกซ์ คือส่วนที่เรียกว่า รูท (root) ซึ่งโดยทั่วไปจะใช้เครื่องหมาย / แทนระบบแฟ้มข้อมูลนี้ รูป 2.6 แสดงความสัมพันธ์ ระหว่างแฟ้มข้อมูลและไดเรคทอรี่ วงกลมแทนความหมายของไดเรคทอรี่



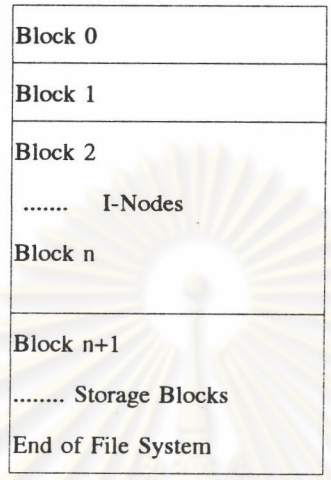
รูปที่ 2.6 ระบบแฟ้มข้อมูลของยูนิกซ์

สำหรับระบบแฟ้มข้อมูลที่ต่ออยู่กับระบบแฟ้มข้อมูลส่วนรูทนั้น อาจเรียกได้ว่าเป็นใบไม้ (leaf) หรือจุดต่อ (mount point) ที่ใช้เชื่อมต่อระหว่างส่วนรูท และส่วนอื่นที่สามารถใช้อ่านและบันทึกได้ (mountable file system) ชื่อของระบบแฟ้มข้อมูลมักเรียกตามชื่อของไดเรคทอรี่นั้น เช่นระบบแฟ้มข้อมูล usr ดังรูป 2.7



รูปที่ 2.7 ระบบแฟ้มข้อมูล /usr โดยเพิ่มต่อจากส่วนรูท

ระบบปฏิบัติการยูนิกซ์ รู้จักระบบแฟ้มข้อมูล จากการจัดเรียงของบล็อคในหน่วยความจำสำรอง โดยจัดแบ่งออกเป็น 4 ส่วน ดังรูปที่ 2.8 คือ



รูปที่ 2.8 การจัดระบบแฟ้มข้อมูล

2.3.3.3.1 บล็อคที่ 0 ไม่ใช้สำหรับการทำงานปกติ เก็บสำรองไว้ใช้เฉพาะในขั้นตอนการเปิดระบบ (boot procedure) เท่านั้น

2.3.3.3.2 บล็อคที่ 1 เรียกว่า ซูเปอร์บล็อค (the super-block) ใช้เก็บสารสนเทศเกี่ยวกับระบบแฟ้มข้อมูล

1) ขนาดและสถานะของระบบแฟ้มข้อมูล ได้แก่ ชื่อ ขนาด รหัสแสดงสถานะการอ่านและบันทึก สถานะการเปลี่ยนแปลงค่าในซูเปอร์บล็อค วันและเวลาของการแก้ไข ครั้งสุดท้าย

2) สารสนเทศของไอโหนด ได้แก่ จำนวนไอโหนดที่ใช้แล้ว จำนวนไอโหนดที่ยังไม่ใช้ แถวลำดับของข้อมูลไอโหนดที่ว่าง 100 ที่ และตัวชี้ไปยังแถวลำดับนั้น

3) สารสนเทศของบล็อคข้อมูล ได้แก่ จำนวนบล็อคที่ว่าง แถวลำดับที่เก็บข้อมูลของบล็อคที่ว่าง 50 บล็อค และตัวชี้ไปยังแถวลำดับนั้น

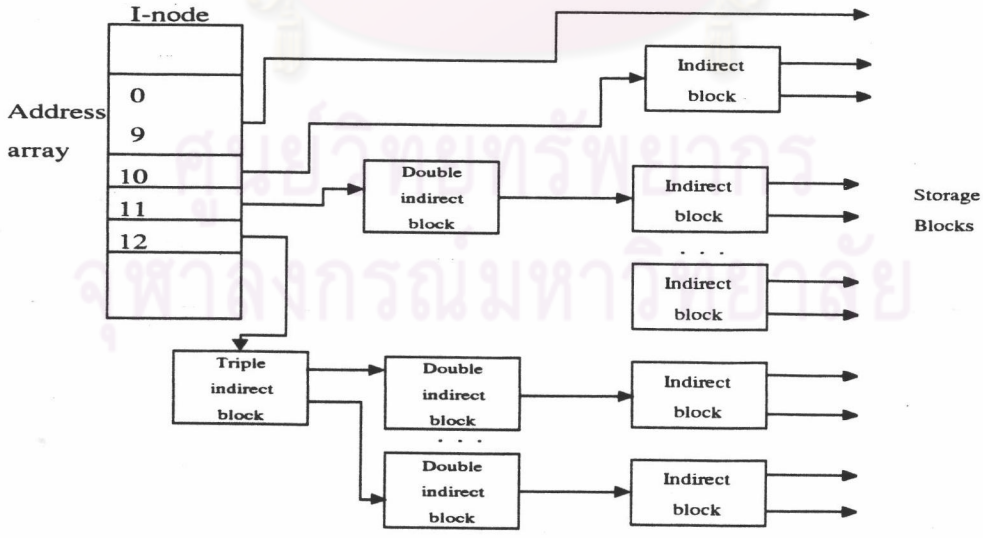
การที่ซูเปอร์บล็อคไม่เก็บเนื้อที่ว่างที่เหลืออยู่ทั้งหมด แต่เก็บสำรองให้เพียงพอกับการใช้งานของระบบเท่านั้น เพราะขณะใดขณะหนึ่งถ้าเนื้อที่ของระบบแฟ้มข้อมูลใกล้จะหมด (ทั้งกรณีของจำนวนไอโหนดและจำนวนบล็อคข้อมูล) ระบบก็ยังมีเนื้อที่จำนวนหนึ่ง เหลืออยู่เพื่อใช้ในการทำงานต่อไป โดยไม่เกิดการชะงักงัน (halt) ซึ่งข้อมูลนี้จะเก็บอยู่ในบล็อคข้อมูล

2.3.3.3.3 ไอโหนด (I-nodes) เรียกว่า information node หรือ index node บางครั้งเขียนอยู่ในรูป inode เช่นเดียวกับ i-list หรือ ilist ซึ่งหมายถึงรายการของไอโหนด และ i-number

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

หรือ inumber ซึ่งคือตำแหน่งของไอโหนด ในรายการไอโหนดจะเก็บสารสนเทศของแฟ้มข้อมูลที่อยู่ภายใต้ไดเรคทอรี่ ยกเว้นชื่อแฟ้มข้อมูล ไอโหนดมีความยาว 64 ไบต์ ซึ่งก็คือ 8 ไอโหนดต่อ 1 บล็อค (physical block) ไอโหนดประกอบด้วย

1) ชนิดและสถานะของแฟ้มข้อมูล โดยแยกตามสัญญลักษณ์คือ
   แฟ้มข้อมูลทั่วไป ใช้เครื่องหมาย - (hyphen)
   แฟ้มข้อมูลไดเรคทอรี่ " ตัวอักษรดี (d:directory)
   แฟ้มข้อมูลพิเศษแบบบล็อค " ตัวอักษรบี (b:block)
   แฟ้มข้อมูลพิเศษแบบอักขระ " ตัวอักษรซี (c:charecter)
   แฟ้มข้อมูลไฟโฟหรือเรียกว่าไปป์ " ตัวอักษรพี (p:pipe)
   ส่วนสถานะของแฟ้มข้อมูล คือการกำหนดค่าความปลอดภัยของแฟ้มข้อมูล
2) จำนวนแฟ้มข้อมูลที่เชื่อมโยงกัน
3) กลุ่มของแฟ้มข้อมูล
4) จำนวนไบต์ ของแฟ้มข้อมูล
5) แถวลำดับของที่อยู่ของแฟ้มข้อมูล 13 ตัว
6) วันและเวลาที่มีการเรียกใช้แฟ้มข้อมูลครั้งล่าสุด
7) วันและเวลาที่มีการแก้ไขแฟ้มข้อมูลครั้งล่าสุด
8) วันและเวลาที่สร้างแฟ้มข้อมูล



รูปที่ 2.9 การอ้างอิงที่อยู่ของระบบแฟ้มข้อมูล

ส่วนที่มีความสำคัญที่สุดคือแถวลำดับของที่อยู่ 13 ตัว โดยที่ 10 ตัวแรก (หมายเลข 0-9) จะเป็นเลขที่อยู่โดยตรง คือจะชี้ไปยังบล็อคข้อมูล 10 บล็อคแรกของบล็อคข้อมูลนั้น (ดังรูปที่ 2.9) กรณีที่แฟ้มข้อมูลมีขนาดใหญ่กว่า 10240 ไบต์ ระบบจะใช้แถวลำดับที่ 11 (หมายเลข 10) ชี้ไปยังบล็อคที่อยู่ ซึ่งเป็นการอ้างถึงข้อมูลโดยอ้อม บล็อคที่อยู่นี้สามารถเก็บข้อมูลได้ถึง 256 บล็อคข้อมูล ส่วนแถวลำดับที่ 12 และ 13 (หมายเลข 11 และ 12) จะชี้ไปยังบล็อคที่อยู่แบบ 2 ชั้น และ 3 ชั้น ตามลำดับ

ดังนั้นขนาดของแฟ้มข้อมูลจะจำกัดที่ขนาดใหญ่มาก ตารางที่ 2.1 แสดงจำนวนไบต์ที่สามารถอ้างถึงได้จากบล็อคที่อยู่เป็นชั้นๆ จากแถวลำดับในไอโหนด โดยจำนวนไบต์ที่อ้างถึงในตาราง เป็นค่าที่รวมค่าการอ้างถึงในลำดับต่ำกว่าไว้ด้วยแล้ว เช่น ค่าการอ้างถึงแบบ 3 ชั้น ได้รวมค่าการอ้างถึงแบบ 2 ชั้น การอ้างถึงแบบชั้นเดียว และการอ้างถึงโดยตรงไว้ด้วย

ตารางที่ 2.1 จำนวนไบต์ที่อ้างอิงได้ แบบต่างๆ

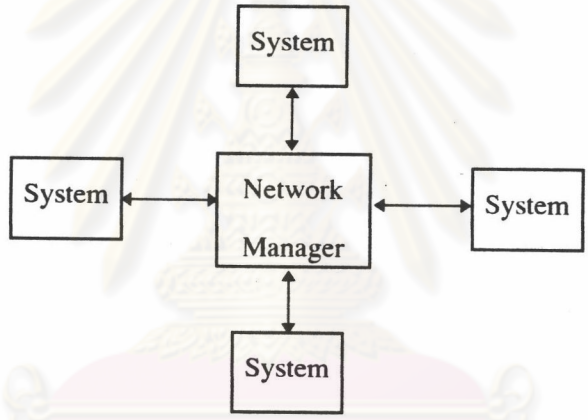
MAXIMUM NUNBER OF BYTES ADDRESSABLE BY				
Logical Block Size	Direct Blocks	Single Indirect Blocks	Double Indirect Blocks	Triple Indirect Blocks
512 bytes	5120	64K	8M	1G
1024 bytes	10240	256K	64M	16G
2048 bytes	20480	1M	512M	256G

ดังนั้นโดยทฤษฎีแล้ว ระบบแฟ้มข้อมูลอาจมีขนาดใหญ่มากได้ แต่ในการทำงานจริงขนาดของแฟ้มข้อมูลจะถูกจำกัดโดยค่าที่เก็บอยู่ในเขตข้อมูล (field) ในส่วนของไอโหนด ซึ่งความยาวของเขตข้อมูลนี้ยาว 32 บิท ดังนั้นแฟ้มข้อมูลอาจมีขนาดใหญ่ที่สุดได้ราว 4 พันล้านไบต์

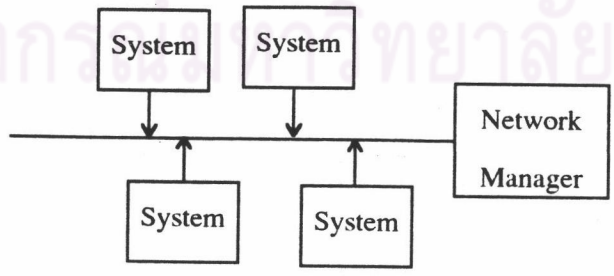
2.3.3.3.4 บล็อคข้อมูล (Storage or Data blocks) ใช้สำหรับเก็บข้อมูลจริง โดยระบบจะดูแลเนื้อที่ว่างทั้งหมดที่เหลืออยู่และจัดสรรเนื้อที่เพื่อเก็บข้อมูลนั้น ในกรณีแฟ้มข้อมูลทั่วไปจะเก็บเนื้อข้อมูลซึ่งไม่มีการกำหนดรูปแบบ กรณีที่เป็นไดเรคทอรี่ บล็อคนี้จะเก็บรายการข้อมูล ซึ่งแต่ละรายการมีขนาด 16 ไบต์ โดยใช้ เก็บค่าตำแหน่งของไอโหนด ความยาว 2 ไบต์ที่เหลือ 14 ไบต์เก็บชื่อแฟ้มข้อมูล หรือชื่อไดเรคทอรี่ย่อยที่อยู่ภายใต้ไดเรคทอรี่นั้น ส่วนเนื้อที่ที่ไม่ถูกใช้งานจะต่อกันเป็นสายของข้อมูลแบบรายการโยง (linked list) คือแต่ละบล็อคข้อมูลจะเก็บที่อยู่ของบล็อคที่อยู่ถัดไป

2.3.3.4 การเชื่อมโยงเครือข่าย (Network)

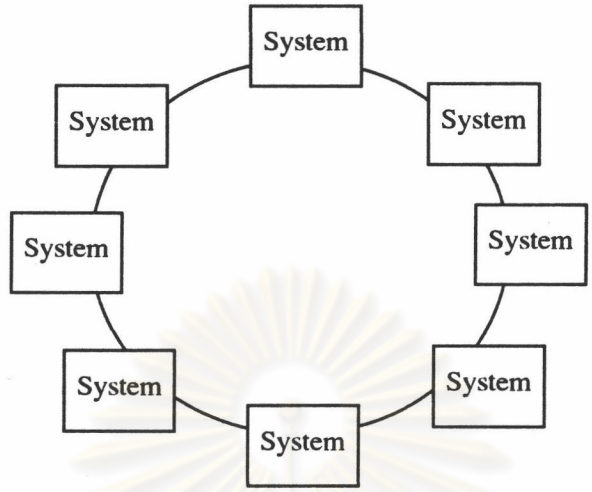
การเชื่อมโยงเครือข่าย คือวิธีการเชื่อมต่อระหว่างระบบเครื่องคอมพิวเตอร์ตั้งแต่ 2 ระบบขึ้นไป เข้าด้วยกันเพื่อการแลกเปลี่ยนข้อมูลข่าวสารระหว่างกัน การเชื่อมโยงนี้อาจต่อแบบง่ายโดยผ่านสายโทรศัพท์ หรือต่อเชื่อมโยงกับระบบเครือข่ายที่ได้ติดต่อตั้งไว้แล้ว เช่น เครือข่ายอินเตอร์เนท (Internet Network) ในการสื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์นั้น มีกฎเกณฑ์ (Protocol) หรือข้อกำหนดในการติดต่อสื่อสาร สำหรับให้ระบบเครื่องคอมพิวเตอร์ที่ต้องการติดต่อกันใช้เพื่อการสื่อสารข้อมูลระหว่างกัน การเชื่อมโยงเครือข่ายระหว่างระบบเครื่องคอมพิวเตอร์นี้เป็นลักษณะสำคัญอีกประการหนึ่งของระบบปฏิบัติการยูนิกซ์ การจัดรูปแบบการเชื่อมโยงสามารถทำได้หลายแบบ ดังตัวอย่างรูปที่ 2.10 - 2.12



รูปที่ 2.10 การจัดรูปแบบการเชื่อมโยงแบบสตาร์ (Star Network)



รูปที่ 2.11 การจัดรูปแบบการเชื่อมโยงแบบหลายจุด (Multidrop Network)



รูปที่ 2.12 การจัดรูปแบบการเชื่อมโยงแบบวงแหวน (Ring Network)

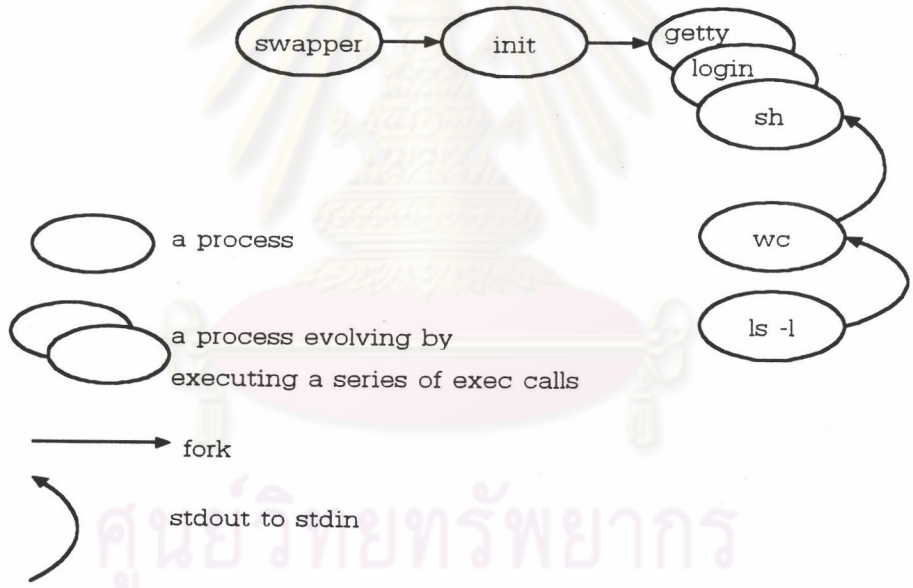
2.3.3.5 โครงสร้างของโปรเซส (Process Structure)

ดังได้กล่าวแล้วว่าส่วนแก่นเป็นหัวใจของระบบปฏิบัติการยูนิกซ์ ซึ่งส่วนใหญ่เป็นโปรแกรมในภาษาซี เฉพาะส่วนคำสั่งเพื่อการทำงานของฮาร์ดแวร์ที่ไม่อาจใช้ หรือไม่สะดวกที่จะใช้ภาษาซีเขียนโปรแกรมได้เท่านั้นที่เป็นโปรแกรมในภาษาแอสเซมบลี หรือเป็นคำสั่งที่มีการเรียกใช้งานบ่อย การใช้ภาษาแอสเซมบลีทำให้การทำงานในส่วนนี้มีประสิทธิภาพมากกว่า ส่วนแก่นนี้เป็นส่วนเล็กๆ ของระบบปฏิบัติการยูนิกซ์ การทำความเข้าใจการจัดการการทำงานของโปรเซสผู้ใช้งานในระบบ ช่วยให้สามารถทำการปรับปรุงประสิทธิภาพของระบบได้อย่างถูกต้อง

โปรเซส คือการทำงานจริงของโปรแกรมในส่วนเฉพาะเจาะจงส่วนใดส่วนหนึ่ง ซึ่งประกอบไปด้วยส่วนคำสั่ง (self-contained code or text section) ส่วนข้อมูล (data section) ทั้งที่มีการกำหนดค่าเริ่มต้นและยังไม่มีการกำหนดค่าเริ่มต้น และส่วนแสตกที่ใช้ในการทำงาน (stack section) โดยขนาดของส่วนแสตกนี้มีขนาดเปลี่ยนแปรไปตามการทำงานของส่วนแก่น โปรเซสจะทำงานตามคำสั่งในส่วนคำสั่งของตนเองโดยไม่สามารถเข้าไปทำงานในส่วนคำสั่งของโปรเซสอื่น การอ่านหรือเขียนข้อมูล และการทำงานในส่วนแสตกทำได้เฉพาะในส่วนข้อมูลและส่วนแสตกของตนเอง โดยไม่อาจเรียกใช้ข้อมูล หรือแสตกของโปรเซสอื่น

โปรเซสมีการติดต่อสื่อสาร กับระบบปฏิบัติการหรือกับโปรเซสอื่น โดยการเรียกใช้ฟังก์ชันการทำงานของระบบ เช่น การจัดเนื้อที่ การกำหนดค่าเริ่มต้น การควบคุมติดตามการทำงาน การปรับเปลี่ยนข้อมูลเพื่อผลในการทำงาน การหยุดการทำงาน ทุกโปรเซสที่ระบบสร้างขึ้น จะได้รับหมายเลขประจำโปรเซส (process id) ซึ่งไม่ซ้ำกัน โปรเซสหมายเลขศูนย์ ซึ่งเกิดขึ้น

ในขั้นตอนการเปิดระบบ มีหน้าที่ในการนำโปรเซสสับเปลี่ยนกัน เพื่อเข้าใช้งานในหน่วยความจำหลัก เรียกว่าแสวปเปอร์ (swapper) ดังรูปที่ 2.13 และสร้างโปรเซสหมายเลข 1 เพื่อการเริ่มต้นระบบ โปรเซสหมายเลข 1 นี้ถือเป็นโปรเซสแม่ของทุกโปรเซสที่ทำงานอยู่ในระบบ เมื่อเปิดระบบเรียบร้อยแล้ว ระบบจะสร้างโปรเซสเกทตี้ (getty process) เพื่อตรวจสอบการทำงานของสายเทอร์มินอล ที่ต่ออยู่ในระบบว่ามีผู้ใช้งานทำงานอยู่หรือไม่ กรณีที่มีการตอบกลับจากสายอุปกรณ์ โปรเซสเกทตี้นี้จะเปิดโอกาสให้ผู้ใช้งานเข้าใช้งานได้ (exec login) โดยการพิมพ์รหัสในการเข้าใช้งาน (login) และรหัสผ่าน (password) เพื่อการเข้าสู่ระบบ เมื่อทุกอย่างถูกต้องเรียบร้อยโปรเซสใหม่ถูกสร้างขึ้นเรียกว่า เชลล์เพื่อใช้ในการทำงาน เมื่อผู้ใช้เลิกใช้งานโปรเซสเชลล์จะหยุดการทำงาน และกลับสู่โปรเซสแม่คือโปรเซสเกทตี้ เพื่อการตรวจสอบสายเทอร์มินอล และรอคอยผู้ใช้งานต่อไป



รูปที่ 2.13 ขั้นตอนการเริ่มทำงานของระบบ

เมื่อมีผู้ใช้งาน ระบบจะสร้างโปรเซสของผู้ใช้งานนั้นขึ้น ในขณะเดียวกันระบบก็สร้างโปรเซสของระบบเองเพื่อการให้บริการในส่วนที่เป็นหน้าที่ของเคอร์เนล เช่น การสับค่าหน้า การแก้ไขการทำงานในส่วนเคอร์เนลเท่ากับต้องทำการสร้างระบบปฏิบัติการใหม่ นอกจากนั้นยังมีการสร้างโปรเซสดีมอนซึ่งคอยให้บริการในส่วนการควบคุมระบบเครือข่าย การพิมพ์ การทำบัญชี

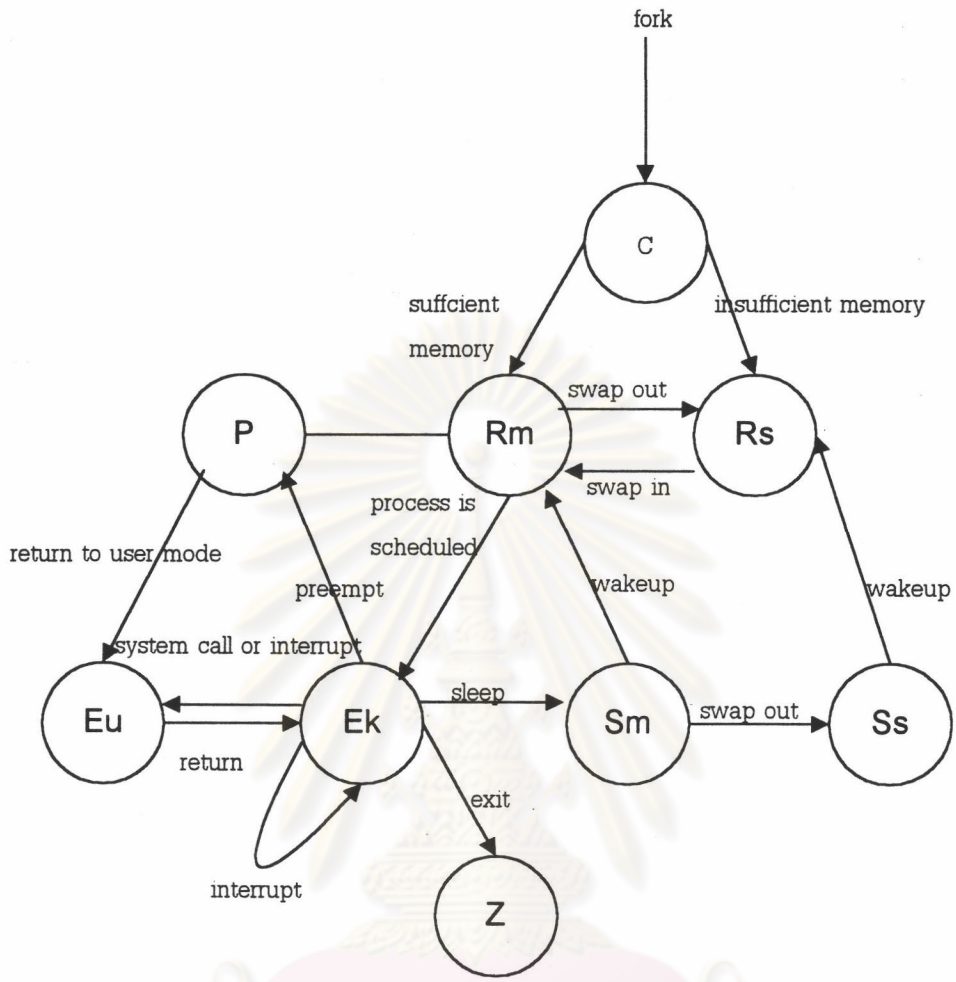
ระบบ โปรแกรมเหล่านี้สามารถแก้ไขและนำมาติดตั้งใหม่ได้ในลักษณะเดียวกับโปรแกรมของผู้ใช้งานอื่นโดยไม่ต้องทำการสร้างส่วนปฏิบัติการใหม่

2.3.3.6 วัฏจักรของโปรเซส (Process Life Cycle)

ในระบบคอมพิวเตอร์ที่มีการใช้งานของผู้ใช้หลายคน ระบบจะเต็มไปด้วยการทำงานของโปรเซส ส่วนเคอร์เนลจะจัดการแบ่งสรรการใช้งานทรัพยากรระบบ รวมทั้งการเข้าใช้งานในหน่วยความจำหลักซึ่งมีเนื้อที่จำกัด การจัดการบริหารงาน จึงอาศัยการเปลี่ยนแปลงสถานะของโปรเซส ดังรูปที่ 2.14 โปรเซสซีถูกสร้างขึ้นด้วยฟังก์ชันระบบฟอร์ค (fork system call) ถ้าในหน่วยความจำหลักมีเนื้อที่เพียงพอ โปรเซสซีจะถูกนำเข้าสู่หน่วยความจำหลักเพื่อต่อลำดับการเข้าใช้งาน แต่ถ้าเนื้อที่ใช้งานไม่เพียงพอต้องรอคอยเพื่อเข้าใช้งาน เมื่อโปรเซสถูกเลือกให้ทำงานได้โปรเซสจะทำงานไปจนกระทั่งถูกขัดจังหวะโดยโปรเซสอื่นที่มีความสำคัญมากกว่า หรือจนกระทั่งส่วนแบ่งช่วงเวลาที่ได้รับอนุญาตให้ทำงานหมดลง หรือเมื่อต้องรอคอยการทำงานในส่วนอื่นให้เรียบร้อยก่อนจึงจะสามารถทำงานต่อไปได้ เช่นการอ่านและบันทึกข้อมูล หรือการรอคอยกระบวนการสับค่าหน้าเพื่อนำหน้าใหม่ที่ต้องการมาทำงาน

2.3.3.7 การจัดการหน่วยความจำ (Memory management)

กระบวนการการจัดลำดับ เพื่อเข้าใช้งานในหน่วยความจำหลักมีอิทธิพลเป็นอย่างมากต่อการจัดการหน่วยความจำหลัก กล่าวคือโปรเซสทุกโปรเซสมีโอกาสในการได้รับเลือกให้เข้าใช้งานในหน่วยความจำหลักเท่าเทียมกัน ซึ่งหน่วยจัดการจะตัดสินว่าโปรเซสใดจะได้รับโอกาสนั้น ในขณะเดียวกันต้องบริหารหน่วยความจำสำรองเพื่อเตรียมเนื้อที่สำหรับการสับค่าหน้า (swap device) โดยการตรวจสอบว่าในขณะใดๆ มีเนื้อที่ในหน่วยความจำหลักเหลืออยู่เพียงพอหรือไม่ กรณีที่พบว่าเนื้อที่ที่เหลืออยู่ไม่เพียงพอ ต้องนำหน้าที่ไม่ได้ใช้งานเก็บลงหน่วยความจำสำรอง เพื่อเตรียมเนื้อที่ในหน่วยความจำหลัก ให้ว่างเพียงพอสำหรับการทำงานในหน้าต่อไปหรือสำหรับโปรเซสใหม่ในลำดับการทำงาน สมัยก่อนเมื่อเนื้อที่ของหน่วยความจำหลักมีน้อยการทำงานจึงเป็นการนำโปรเซสทั้งโปรเซสเก็บในหน่วยความจำสำรอง ปัจจุบันการปรับปรุงขนาดของหน่วยความจำหลักให้มีเนื้อที่มากขึ้นและราคาถูกลง และการใช้เทคนิคการทำงานแบบการบรรจุแผ่นเมื่อใช้ (demand paging) จึงเป็นการนำหน้าที่ไม่มีการเรียกใช้งาน หรือมีการเรียกใช้น้อยเก็บลงในหน่วยความจำสำรองเท่านั้น



รูปที่ 2.14 การเปลี่ยนแปลงสถานะของโปรเซสผู้ใช้งาน

คำอธิบาย

c เป็นโปรเซสที่ถูกสร้างขึ้นใหม่ด้วยฟังก์ชันระบบฟอร์ค

Rs โปรเซสซึ่งพร้อมจะทำงานแต่ถูกเปลี่ยน (Swap Out) ออกจากหน่วยความจำหลัก

Rm โปรเซสที่ได้รับโอกาศให้เข้าใช้งานในหน่วยความจำหลัก

P โปรเซสที่ถูกบังคับให้ปลดปล่อยทรัพยากรต่างๆ และรอที่จะเปลี่ยนกลับไปใช้งานในส่วนโปรแกรมผู้ใช้งาน (User Mode)

Eu โปรเซสที่กำลังทำงานในส่วนโปรแกรมผู้ใช้งาน

Ek โปรเซสที่กำลังทำงานในส่วนโปรแกรมระบบ (Kernel Mode)

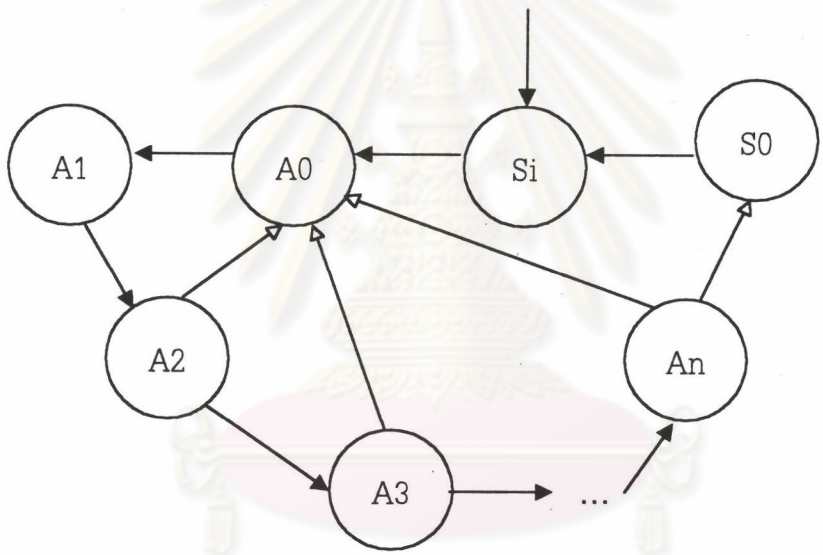
Sm โปรเซสที่หลับ (Sleep) อยู่ในหน่วยความจำหลัก

Ss โปรเซสที่หลับ และถูกเปลี่ยนออกจากหน่วยความจำหลัก

Z โปรเซสที่จบการทำงาน (Exit) แต่โปรเซสแม่ยังไม่สังเกตพบ

เทคนิคการบรรจุแผ่นเมื่อใช้นี้ ทำให้ขนาดของหน่วยความจำหลักไม่มีอิทธิพลต่อการทำงานของโปรเซสต่างๆ ในระบบ เพียงแต่มีข้อจำกัดของจำนวนที่อยู่เสมือน (virtual address) ในหน่วยความจำสำรอง การเลือกว่าจะนำหน้าใดเก็บในหน่วยความจำสำรองใช้อัลกอริทึมแบบการแทนที่แบบไม่ได้ใช้นานที่สุด (LRU:least-recently used) คือการเลือกหน้าที่มีการเรียกใช้งานน้อยที่สุด ออกจากหน่วยความจำหลัก ดังรูป 2.15

เมื่อโปรเซสเรียกใช้หน้าส่วนที่ไม่อยู่ในหน่วยความจำหลักเรียกว่า เกิดการขาดแผ่น (page fault) โปรเซสนั้นจะถูกหยุดการทำงานชั่วคราวจนกว่าหน้าที่ต้องการหรือหน้าที่ถูกต้องจะถูกบรรจุ หรือถูกเรียกเข้าทำงานในหน่วยความจำหลัก เมื่อโปรเซสได้รับอนุญาตให้ทำงานได้ โปรเซส นั้นจะเริ่มทำงาน ณ จุดที่เกิดเหตุการณ์



รูปที่ 2.15 สถานะของหน้า ในขั้นตอนวิธีแบบไม่ใช้นานที่สุด

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

คำอธิบาย

Si	สภาวะเริ่มต้น หน้าถูกเรียกเข้าสู่หน่วยความจำหลัก
So	หน้าถูกสับออกจากหน่วยความจำหลัก
A0-An	หน้าในสภาวะต่างๆ โดยหน้าที่ถูกเรียกใช้จะเปลี่ยนกลับสู่สภาวะ A0
——→	หน้าที่ถูกเรียก จะเข้าสู่สภาวะ A0
——→	หน้าที่ไม่ถูกเรียก จะเข้าสู่สภาวะ Ai+1

2.4 แนวทางการจัดการด้านสมรรถนะของระบบคอมพิวเตอร์

2.4.1 สมรรถนะของหน่วยประมวลผลกลาง

ในการเรียกใช้ หรือร้องขอการทำงานของทรัพยากรระบบใดๆ เมื่อมีการใช้งานจนถึงความจุสูงสุดของทรัพยากรนั้น ระบบจะเริ่มจัดลำดับการขอเข้าใช้งานทรัพยากรระบบนั้น เมื่อผู้ใช้ต่างเรียกใช้ทรัพยากรระบบเดียวกัน ลักษณะแบบนี้ เรียกว่าเกิดเหตุการณ์คอขวด (Bottleneck) ขึ้น ในระบบคอมพิวเตอร์ซึ่งใช้ระบบปฏิบัติการยูนิกซ์ มีศักยภาพการทำงานของทรัพยากรระบบ ซึ่ง ณ จุดหนึ่งอาจเกิดเหตุการณ์คอขวดขึ้นได้ ทั้งในส่วนของหน่วยประมวลผลกลาง หน่วยความจำหลัก และหน่วยความจำสำรองชนิดจานแม่เหล็ก แต่สิ่งที่ควรให้ความสำคัญเป็นลำดับแรก คือการพิจารณาถึงการเกิดเหตุการณ์คอขวดของหน่วยประมวลผลกลาง เพราะทำให้สามารถดูแลระบบคอมพิวเตอร์ให้ทำงานได้อย่างไม่มีปัญหาใดๆ

2.4.1.1 ลักษณะของระบบคอมพิวเตอร์ที่เกิดคอขวดของหน่วยประมวลผลกลาง

เมื่อส่วนของหน่วยประมวลผลกลางเกิดเหตุการณ์คอขวดขึ้น จะเกิดอาการบางลักษณะบางครั้งอาจเป็นความรู้สึกส่วนตัว เช่น รู้สึกว่าเครื่องทำงานช้า แต่ลักษณะบางอย่างจะเห็นได้ชัดคือ

2.4.1.1.1 ระบบคอมพิวเตอร์ทำงานตลอดเวลา ในระยะเวลาหนึ่ง

2.4.1.1.2 ลำดับการเข้าใช้งานในหน่วยประมวลผลกลางยาว

2.4.1.1.3 การทำงานของโปรแกรมผู้ใช้งานใช้เวลานาน เวลาการทำงานของโปรแกรมผู้ใช้งาน คือเวลาที่ใช้ในการทำงานของโปรแกรมประยุกต์ที่เราเขียนขึ้นใช้เองหรือจากการจัดซื้อมาใช้งานภายในหน่วยงานก็ได้

2.4.1.1.4 การทำงานของโปรแกรมระบบใช้เวลานาน เวลาการทำงานของโปรแกรมระบบ คือเวลาที่โปรเซสใช้ไปในการทำงานต่างๆ ของระบบ โดยที่โปรเซสนั้นเป็นโปรเซสที่เกิดจากการทำงานของโปรแกรมประยุกต์ เช่น การทำงานของเทอร์มินอล หรือส่วนรับเข้า และส่งออกอื่นๆ รวมทั้งการทำงานของโปรเซสของระบบ เช่น การจัดการทำงาน

2.4.1.2 การจัดการกับเหตุการณ์คอขวดของหน่วยประมวลผลกลาง

เมื่อเกิดเหตุการณ์คอขวดขึ้นที่หน่วยประมวลผลกลาง แนวทางเพื่อแก้ไขปัญหาได้หลายแบบ บางครั้งอาจต้องเปลี่ยนอุปกรณ์ ซึ่งการแก้ไขปัญหาได้อย่างถูกต้องนั้นต้องใช้เวลาและความรู้ความเข้าใจเกี่ยวกับสภาวะแวดล้อมของระบบเครื่องคอมพิวเตอร์เป็นอย่างดี อย่างไรก็ตามมีแนวทางเพื่อพิจารณาเลือกใช้ในการทำงานดังต่อไปนี้

2.4.1.2.1 ปรับปรุงคุณภาพการทำงานของโปรแกรมประยุกต์ ในแนวทางเพื่อลดความต้องการการทำงานจากหน่วยประมวลผลกลาง

2.4.1.2.2 ลดระดับความสำคัญของโปรเซสที่ไม่จำเป็น

2.4.1.2.3 แบ่งแยกการทำงานไปยังระบบคอมพิวเตอร์อื่น

2.4.1.2.4 หยุดการทำงานที่ไม่จำเป็น เช่น โปรเซสการทำงานเกี่ยวกับระบบเครือข่าย

2.4.1.2.5 นำการทำงานที่เป็นการประมวลผลแบบแบทช์ เปลี่ยนเวลามาทำในช่วงที่มีการทำงานน้อย

2.4.1.2.6 เปลี่ยนรุ่นของหน่วยประมวลผลกลาง

2.4.2 สมรรถนะของหน่วยความจำหลัก

ข้อจำกัดอย่างหนึ่งของระบบคอมพิวเตอร์ คือขนาดของหน่วยความจำหลัก ซึ่งจะกระทบถึงสมรรถนะการใช้งานของระบบคอมพิวเตอร์เมื่อมีการทำงานของโปรแกรมขนาดใหญ่เกินกว่าขนาดของหน่วยความจำหลัก เพราะระบบจะเริ่มกระบวนที่นำหน้าในหน่วยความจำหลักเก็บลงในหน่วยความจำสำรองชนิดจานแม่เหล็กหรือสลับกัน เรียกว่าการสับค่าหน้า มีผลทำให้ประสิทธิภาพการทำงานในระบบลดลง สมรรถนะของหน่วยความจำหลักเป็นส่วนที่เข้าใจได้ง่าย เพราะพิจารณาถึงขนาดของหน่วยความจำหลักมีใช้งานอย่างเพียงพอหรือไม่เพียงพอ 2 กรณีเท่านั้น การสับค่าหน้าเป็นกระบวนการจัดการหน่วยความจำหลัก อย่างหนึ่ง เรียกว่าการทำงานของหน่วยความจำเสมือนทำให้สามารถเรียกการทำงานของโปรแกรมโดยไม่มีขีดจำกัดเกี่ยวกับขนาดของหน่วยความจำหลักโดยการแบ่งโปรแกรมออกเป็นส่วนย่อยๆ เรียกว่า หน้า แล้วใช้วิธีเรียกการทำงานเฉพาะหน้าที่ต้องการใช้เข้าสู่หน่วยความจำหลัก สำหรับการจัดการหน่วยความจำแบบเปลี่ยนทั้งโปรเซส จะใช้ใน 2 กรณีคือ

- โปรเซสที่อยู่ในสภาวะหลับ (sleep) นานเกินกว่า 20 วินาที ระบบจะถือว่าโปรเซสนั้นปล่อยเวลาให้สูญเปล่า และต้องถูกเปลี่ยนออก
- ในสถานการณ์ที่เนื้อที่ในหน่วยความจำหลักเหลือน้อยมาก โปรเซสต้องถูกเปลี่ยนออกเพื่อเพิ่มเนื้อที่ให้แก่หน่วยความจำหลัก เรียกว่าการเปลี่ยนออกในสภาวะร้ายแรง (Desperation Swapping)

การจัดการหน่วยความจำทั้ง 2 แบบทำให้การทำงานของระบบลดลง แต่ไม่มีผลกระทบมากนัก นอกจากการจัดการหน่วยความจำ ไม่อาจทำให้สภาพการใช้งานหน่วยความจำหลักอยู่ในสภาวะปกติได้ สมรรถนะของระบบจะลดลงจนเริ่มสังเกตได้ การจัดการหน่วยความจำช่วยป้องกันระบบจากความเสียหายเมื่อเนื้อที่หมดลง แต่ไม่สามารถเพิ่มเนื้อที่ของหน่วยความจำหลักได้

เมื่อใดก็ตามที่การทำงานของหน่วยความจำหลักมีปัญหา การทำงานของส่วนรับเข้าและส่งออก รวมทั้งการทำงานของหน่วยประมวลผลกลางก็จะมีปัญหาตามไปด้วย

ในทางปฏิบัติเมื่อสมรรถนะของระบบลดลงจนผู้ใช้งานในระบบเริ่มบ่น ควรตรวจสอบว่าขณะนั้นระบบมีการทำงานอะไรบ้าง ซึ่งถ้าพบว่าระบบทำการสับค่าหน้าอยู่ตลอดเวลา แสดงว่าปัญหาเกิดขึ้นจากขนาดของหน่วยความจำหลักไม่เพียงพออย่างแน่นอน

สำหรับส่วนของการเปลี่ยนโปรเซส ถ้าตรวจพบว่ามีการเปลี่ยนโปรเซสในขณะที่ทำการการสับค่าหน้า แสดงว่าเกิดการเปลี่ยนออกในสภาวะร้ายแรง ต้องตรวจสอบเพื่อการแก้ไขโดยเร็ว สำหรับการแก้ไขเลือกใช้ได้หลายแนวทาง เช่น ซื้อหน่วยความจำเพื่อเพิ่มเข้าในระบบ หรือใช้แนวทางในการปรับปรุงวิธีการทำงานเพื่อวัตถุประสงค์ ดังต่อไปนี้

2.4.2.1 การใช้งานในหน่วยความจำหลักอย่างมีประสิทธิภาพ

2.4.2.1.1 การใช้ส่วนเนื้อความของโปรแกรมร่วมกัน

การใช้ส่วนเนื้อความของโปรแกรมร่วมกันเป็นการกำหนดมาตรฐานของระบบ หมายถึงผู้ใช้งานสามารถเรียกใช้โปรแกรมการทำงานเดียวกันได้โดยไม่ต้องเสียเนื้อที่ในหน่วยความจำหลักเพิ่ม เพราะแบ่งส่วนของโปรแกรมออกเป็นส่วนเนื้อความซึ่งสามารถใช้ร่วมกัน และส่วนข้อมูลซึ่งแยกตามผู้ใช้ ตัวเชื่อมจะทำหน้าที่นี้โดยกำหนดให้ส่วนของเนื้อความมีสถานะอ่านได้เท่านั้นและกำหนดขนาดของส่วนเนื้อความเป็นค่าทวีคูณของหน้า

การเรียกใช้เฉพาะหน้าที่ต้องการ และการใช้ส่วนเนื้อความร่วมกัน เป็นเทคนิคที่นำมาใช้แทนการกำหนดบิทพิเศษ เรียกว่า สติกกี้บิท (Sticky Bit) ให้กับคำสั่งที่มีการเรียกใช้งานบ่อยๆ ซึ่งอาจคงมีใช้อยู่ในบางระบบ ช่วยลดช่วงเวลาการเริ่มทำงานเพราะการกำหนดบิทพิเศษนี้มีผลทำให้คำสั่งนั้นๆ ยังคงอยู่ในส่วนของพื้นที่ที่ใช้ในการสับเปลี่ยน โดยที่ส่วนเคอร์เนลไม่ต้องสำเนาคำสั่งหรือโปรแกรมนั้นอีกครั้งหนึ่งก่อนเริ่มการทำงาน การกำหนดค่าบิทพิเศษนี้ควรใช้กับคำสั่งที่มีการเรียกใช้งานมากๆ เช่น คำสั่ง vi ทั้งนี้เพื่อหลีกเลี่ยงการเสียเนื้อที่ที่ใช้เพื่อการสับเปลี่ยนมากเกินไป

2.4.2.1.2 การเรียกใช้ฟังก์ชันการทำงานจากคลังชุดคำสั่ง

การเรียกใช้ฟังก์ชันการทำงานจากคลังชุดคำสั่งเป็นการนำฟังก์ชันทั้งหลายเก็บในหน่วยความจำเสมือนของระบบ เมื่อโปรแกรมใดก็ตามเรียกใช้ ตัวเชื่อมจะกำหนดค่าไปยังการทำงานของฟังก์ชันที่เรียกใช้นั้น มีผลให้การใช้งานของส่วนหน่วยความหลักและหน่วยความจำสำรองชนิดจานแม่เหล็กอย่างมีประสิทธิภาพ เพราะโปรแกรมงานใช้เนื้อที่น้อยลงจากการอ้างถึงฟังก์ชันการทำงานซึ่งใช้ร่วมกัน

อย่างไรก็ตามถ้าในระบบมีการเรียกใช้ฟังก์ชันเหล่านี้เพียงเล็กน้อย วิธีนี้ไม่ช่วยให้การทำงานดีขึ้น เพราะการเก็บคลังชุดคำสั่งไว้ทั้งชุด ในขณะที่เรียกใช้เพียงเล็กน้อย เป็นการใช้เนื้อที่ในหน่วยความจำหลักโดยเปล่าประโยชน์

2.4.2.1.3 การใช้เทคนิคการเขียนโปรแกรม

กรณีเมื่อความสามารถของระบบลดลง เนื่องจากพบว่าโปรแกรมที่กำลังทำงานมีการใช้งานในหน่วยความจำหลักมาก ควรแนะนำให้ผู้ที่พัฒนาโปรแกรมตรวจสอบการทำงานของโปรแกรมว่ามีข้อผิดพลาดอย่างไร หรือไม่ และควรแก้ไขอย่างไรเพื่อให้มีการใช้งานหน่วยความจำหลักอย่างมีประสิทธิภาพ เช่น ใช้เทคนิคการอ้างถึงแบบกลุ่มซึ่งช่วยเพิ่มประสิทธิภาพการทำงานของการสับค่าหน้า เพราะการอ้างอิงมีลักษณะเป็นแบบสุ่ม ระบบไม่อาจทำนายว่าหน้าใดของหน่วยความจำหลัก จะเป็นที่ต้องการต่อไปแต่การอ้างถึงแบบกลุ่มช่วยให้เรียกใช้หน้าของหน่วยความจำในบริเวณเดียวกัน

2.4.2.1.4 การใช้การทำงานบัฟเฟอร์แคช

บัฟเฟอร์แคช คือ กลุ่มของบัฟเฟอร์มีหน้าที่เป็นตัวกลางในการจัดเก็บข้อมูลไว้ชั่วคราวก่อนที่จะเคลื่อนย้ายไปทำงานยังตัวขับอุปกรณ์ ช่วยให้ประสิทธิภาพการเข้าถึงข้อมูลในจานแม่เหล็กดีขึ้น เพราะหน่วยรับเข้าและส่งออก สามารถส่งผ่านข้อมูลได้รวดเร็วขึ้น ลดเวลาในการค้นหาข้อมูลในจานแม่เหล็ก และโปรแกรมที่ทำงานสามารถใช้ข้อมูลจากส่วนของบัฟเฟอร์นี้ได้ ถ้าเป็นข้อมูลที่ต้องการ ความสัมพันธ์ระหว่างขนาดของบัฟเฟอร์แคชและขนาดของหน่วยความจำหลักจะแปรผกผันกัน กล่าวคือถ้ากำหนดให้บัฟเฟอร์แคชมีขนาดใหญ่ ขนาดของหน่วยความจำจะลดลง แต่ประสิทธิภาพการทำงานของจานแม่เหล็กเพิ่มขึ้น แต่ในทางกลับกันถ้ากำหนดให้ขนาดของบัฟเฟอร์แคชเล็ก ขนาดของหน่วยความจำหลักจะมากขึ้น แต่ประสิทธิภาพการทำงานของจานแม่เหล็กลดลง

การพิจารณาต้องขึ้นกับลักษณะของงานในระบบ คือถ้าเป็นงานที่เรียกใช้แฟ้มข้อมูลมาก หรือเรียกใช้แฟ้มข้อมูลขนาดใหญ่ ต้องคำนึงถึงประสิทธิภาพการทำงานของจานแม่เหล็ก ยกเว้นกรณีที่แฟ้มข้อมูลที่ใช้งาน มีขนาดใหญ่กว่าขนาดของบัฟเฟอร์แคชมาก ควรเลือกประสิทธิภาพของหน่วยความจำหลัก และกำหนดขนาดของบัฟเฟอร์แคชให้เล็ก ส่วนกรณีที่ลักษณะงานในระบบเป็นงานที่เกี่ยวกับการคำนวณ มากกว่าการเรียกใช้ส่วนรับเข้า และส่งออก ควรคำนึงถึงประสิทธิภาพการทำงานของหน่วยความจำหลัก

2.4.2.1.5 การใช้การทำงานบัฟเฟอร์ของสตรีม

สตรีมเป็นช่วยอำนวยความสะดวกของส่วนรับเข้าและส่งออก ที่เป็นแบบอักขระ เช่นการทำงานของจอภาพ โดยระบบได้จัดสรรบัฟเฟอร์ของสตรีมเป็น 9 ระดับ

ตามขนาดของบัฟเฟอร์นั้นคือ สตรีมบัฟเฟอร์ขนาดตั้งแต่ 4, 16, 64, 128, 256, 512, 1024, 2048 และขนาด 4096 ไบต์

จำนวนบัฟเฟอร์ในแต่ละระดับจะเก็บในรูปของค่าพารามิเตอร์ระบบ NBLKn โดย n คือขนาดของบัฟเฟอร์ การกำหนดค่าจำนวนของบัฟเฟอร์ขนาดใหญ่ (2 KB หรือ 4 KB) มาก ทำให้เนื้อที่ในหน่วยความจำหลักของระบบหมดลงอย่างรวดเร็ว แต่ถ้ากำหนดเป็นค่าน้อย โปรแกรมการทำงานของระบบเครือข่าย ทำงานช้าลงเพราะต้องรอคอยบัฟเฟอร์ให้ว่างก่อนที่จะเริ่มส่งข้อมูล และในระบบมีจำนวนจอภาพมาก ต้องกำหนดค่าของจำนวนบัฟเฟอร์ขนาดเล็กเพิ่มขึ้น

2.4.2.1.6 การปรับค่าในตารางควบคุมการทำงานของระบบ

เมื่อมีการปรับขนาดของบัฟเฟอร์ มีผลกระทบต่อขนาดของตารางควบคุมการทำงานของระบบคือตารางควบคุมโปรเซส ตารางควบคุมไอโหนด และตารางควบคุมแฟ้มข้อมูล อย่างไรก็ตามตารางควบคุมการทำงานเหล่านี้มีขนาดเล็ก ซึ่งไม่ช่วยให้ขนาดของหน่วยความจำหลักเพิ่มขึ้นมากนัก

2.4.2.2 การปรับส่วนเคอร์เนลเพื่อลดเนื้อที่การใช้งานในหน่วยความจำหลัก

คือการปรับค่าพารามิเตอร์ของระบบที่เกี่ยวข้องกับกระบวนการสับค่าหน้าหรือกระบวนการเปลี่ยนโปรเซส ระบบปฏิบัติการยูนิกซ์ อนุญาตให้ผู้บริหารระบบสามารถเปลี่ยนแปลงค่าพารามิเตอร์ต่างๆ ของระบบที่ประกอบกันขึ้นเป็นส่วนของเคอร์เนล ดังได้กล่าวแล้วว่าทุกครั้งที่มีการเปลี่ยนแปลงค่าพารามิเตอร์ของระบบ ระบบจะดำเนินการสร้างส่วนเคอร์เนลใหม่เสมอ

2.4.2.2.1 การปรับเปลี่ยนขั้นตอนวิธีของการสับค่าหน้า

การเปลี่ยนโปรเซสเป็นการทำงานโดยปกติของระบบ เพราะโปรเซสใดก็ตามที่ไม่ทำงานประมาณ 20 วินาที โปรเซสนั้นต้องถูกระบบเปลี่ยนออกเพื่อให้โอกาสโปรเซสต่อไปเข้าใช้งานแทน การสับค่าหน้าจะเริ่มต้นทำงานเมื่อถึงขีดจำกัดหนึ่งที่ระบบกำหนดไว้ ทั้งนี้เพื่อรักษาสมดุลย์ของเนื้อที่ใช้งานในหน่วยความจำหลัก การสับค่าหน้าจะหยุดการทำงานเมื่อเนื้อที่ในหน่วยความจำหลักเพียงพอ แต่ถ้าระบบไม่สามารถรักษาสมดุลย์ไว้ได้และถึงขีดจำกัดใหม่การเปลี่ยนออกในสภาวะร้ายแรงจะถูกนำมาใช้ โดยโปรเซสที่กำลังทำงานต้องถูกเปลี่ยนออกโดยทันที วิธีการเพื่อป้องกันคือเปลี่ยนค่าขีดจำกัดนั้น เพื่อให้กระบวนการสับค่าหน้าทำงานก่อนที่จะเกิดปัญหา

2.4.2.2.2 การจัดการเกี่ยวกับเนื้อที่ที่ใช้เก็บโปรเซสที่ถูกเปลี่ยนออก

โดยปกติจำนวนเนื้อที่การทำงานส่วนนี้ถูกกำหนดไว้ในแฟ้มข้อมูลเพื่อการเริ่มต้นระบบ เมื่อเข้าสู่การทำงานในสภาวะหลายผู้ใช้ภายใต้ไดเรคทอรี่ /etc/rc2.d แต่ในระบบปฏิบัติการยูนิกซ์บางรุ่น มีคำสั่งเพื่อการเปลี่ยนแปลงขนาดของจานแม่เหล็กที่ใช้สำหรับเก็บโปรเซส การทำงานที่ถูกเปลี่ยนออก

การเปลี่ยนแปลงแก้ไขค่าพารามิเตอร์เหล่านี้ต้องอาศัยความรู้เกี่ยวกับระบบ และความเข้าใจถึงผลกระทบที่จะเกิดขึ้นในภายหลัง เพราะการเปลี่ยนแปลงค่าจาก ค่ากำหนดมาตรฐานเพื่อความเหมาะสมกับระบบนั้นเป็นการปรับส่วนโครงสร้างสำคัญที่ใช้ควบคุมการทำงานในส่วนหัวใจของระบบ พารามิเตอร์เหล่านี้จะแตกต่างกันไปในระบบปฏิบัติการยูนิกซ์แต่ละตระกูล ผู้บริหารระบบต้องเรียนรู้ และทำความเข้าใจการทำงานก่อนจะแก้ไขหรือเปลี่ยนแปลงค่าต่างๆ เหล่านี้ รายละเอียดของพารามิเตอร์ในภาคผนวก ก.

2.4.3 สมรรถนะของหน่วยความจำสำรองชนิดจานแม่เหล็ก

การพิจารณาถึงประสิทธิภาพการทำงานของจานแม่เหล็กแบ่งเป็น

- การคิดปริมาณงานจากแต่ละโปรเซส

ความเร็วของการที่โปรเซสใดๆ สามารถอ่านหรือบันทึกข้อมูลลงในจานแม่เหล็ก การคำนวณปริมาณงานในแบบนี้ โดยใช้การจับเวลาการทำงานของการทำสำเนาแฟ้มข้อมูลขนาดใหญ่ หรือเขียนโปรแกรมสร้างแฟ้มข้อมูลขนาดใหญ่ โดยให้เกิดความสูญเสียน้อยที่สุด

- การคิดปริมาณงานโดยรวม

ความเร็วโดยรวมที่ทุกโปรเซสในระบบสามารถส่งผ่านข้อมูล จากจานแม่เหล็ก การคำนวณปริมาณงานในแบบนี้ใช้การเขียนโปรแกรมเพื่อจำลองการเข้าถึงข้อมูลในแต่ละส่วนของระบบแฟ้มข้อมูลอย่างต่อเนื่องกัน

- ประสิทธิภาพการเก็บข้อมูล

คือประสิทธิภาพการใช้งานด้านการเก็บข้อมูลบนจานแม่เหล็ก ซึ่งเป็นวัตถุประสงค์แรกในการนำจานแม่เหล็กมาใช้งาน และเนื่องจากราคาของจานแม่เหล็กจะแพงขึ้นตามขนาดของจานแม่เหล็กดังนั้นจึงต้องใช้งานให้เต็มประสิทธิภาพเทียบกับค่าใช้จ่ายที่ต้องสูญเสียไป

ถ้าระบบมีการทำงานของแต่ละโปรเซสอยู่ในเกณฑ์ดีระบบนั้นมีแนวโน้มที่จะให้ปริมาณงานโดยรวมดีไปด้วย แต่อย่างไรก็ตามการทำงานของแต่ละโปรเซสเป็นการเข้าถึงแฟ้มข้อมูลเดียว หรือบนระบบแฟ้มข้อมูลเดียว ส่วนการทำงานโดยรวมเป็นการทำงานกับหลายระบบแฟ้มข้อมูลพร้อมๆ กัน การพิจารณาจึงขึ้นอยู่กับลักษณะงานในระบบ ในกรณีที่ระบบมีการเรียก

ใช้แฟ้มข้อมูลขนาดใหญ่ควรใช้การทำงานแบบแต่ละโปรเซส ระบบที่มีการทำงานการพัฒนาโปรแกรมหรือการใช้คำสั่งโดยทั่วไปควรใช้แบบโดยรวม แต่ถ้าในระบบมีการทำงานทั้ง 2 แบบ ควรแบ่งระบบแฟ้มข้อมูลเพื่อรองรับการทำงานทั้งแบบแต่ละโปรเซส และแบบโดยรวมแยกกันจะยิ่งเพิ่มสมรรถนะการทำงานของจานแม่เหล็ก

สำหรับประสิทธิภาพการเก็บข้อมูลของจานแม่เหล็กนั้น ให้ผลในทางตรงข้ามกับปริมาณ การทำงานของจานแม่เหล็ก ถ้าปริมาณการทำงานเพิ่มขึ้นจะให้ผลด้านการเก็บข้อมูลลดลง เช่นการเพิ่มขนาดของบล็อคข้อมูล จะให้ผลด้านการส่งผ่านข้อมูลมากขึ้นแต่สูญเสียเนื้อที่ในการเก็บข้อมูลไป แต่ถ้ากำหนดให้ขนาดของบล็อคเล็กลง ทำให้การส่งผ่านข้อมูล หรือปริมาณงานน้อยลง แต่ความสามารถในการเก็บข้อมูลมากขึ้น อย่างไรก็ตามการนำจานแม่เหล็กมาใช้งานต้องสูญเสียเนื้อที่ส่วนหนึ่งไปในการจัดรูปแบบเพื่อการใช้งาน เนื้อที่ส่วนที่จำเป็นต้องสูญเสียไปนี้มีขนาดมากหรือน้อยขึ้นกับวิธีการจัดรูปแบบซึ่งจะแตกต่างกันไปตามบริษัทผู้ผลิต ในบางครั้งพบว่าสูญเสียเนื้อที่ในราว 1 ใน 3 ของเนื้อที่ทั้งหมด เหล่านี้ทำให้ประสิทธิภาพด้านการจัดเก็บข้อมูลน้อยลง การนำจานแม่เหล็กมาใช้งานเพื่อให้เกิดประโยชน์สูงสุด คือความสมดุลย์ระหว่างความสามารถในการเก็บข้อมูล และปริมาณงานที่ได้จากการทำงานของจานแม่เหล็ก

คุณลักษณะของจานแม่เหล็กที่มีผลต่อสมรรถนะการทำงานของจานแม่เหล็กคือระยะการค้นหา ความเร็วในการหมุนรอบ และอัตราการส่งผ่านข้อมูล ค่าเหล่านี้แสดงถึงความเร็วในการค้นพบข้อมูลบนจานแม่เหล็ก และความเร็วในการได้รับข้อมูล คุณสมบัติของระยะการค้นหาน้อยทำให้สมรรถนะการทำงานของระบบดีขึ้น เพราะเวลาที่ใช้ในการส่งผ่าน และเวลาที่ใช้ในการค้นหาข้อมูลมีสัดส่วนประมาณ 1:10

เมื่อสร้างและเริ่มใช้งานระบบแฟ้มข้อมูลบนจานแม่เหล็ก การตรวจสอบการใช้งานและเนื้อที่ของจานแม่เหล็กอยู่เสมอช่วยรักษาระดับความสามารถในการทำงานของจานแม่เหล็กได้ โดยมีแนวทางดังนี้

2.4.3.1 การควบคุมการแตกกระจาย ของเนื้อที่การเก็บข้อมูลบนจานแม่เหล็ก

การจัดเก็บข้อมูลบนจานแม่เหล็กในทางกายภาพนั้น บางครั้งระบบไม่สามารถหาเนื้อที่ของ บล็อคข้อมูลว่างๆ ที่ติดกันสำหรับการจัดเก็บได้ ทั้งนี้เพราะการปฏิบัติการของแฟ้มข้อมูล อันได้แก่การเพิ่ม การแก้ไข หรือการลบ ทำให้การจัดเรียงภายในแฟ้มข้อมูลไม่เป็นระเบียบ เกิดการแตกกระจายของเนื้อที่การเก็บข้อมูลบนจานแม่เหล็ก มีผลให้ความสามารถของระบบลดลงเพราะระบบไม่อาจเรียกข้อมูลจากแฟ้มข้อมูลได้จากการทำงานแบบเรียงลำดับ ต้องอ้างถึงข้อมูลในส่วนอื่นๆ ของจานแม่เหล็กทำให้การเข้าถึงข้อมูลโดยใช้เวลามากขึ้น ดังนั้นการ

I17493316

ปรับปรุงเนื้อที่ว่างซึ่งไม่ติดต่อกันเหล่านี้เป็นแนวทางหนึ่งในดำเนินการเพื่อการเพิ่มสมรรถนะการทำงานของจานแม่เหล็ก

2.4.3.2 การสร้างสมดุลย์การเก็บข้อมูลของแต่ละระบบแฟ้มข้อมูล

การตรวจสอบการใช้งานของแต่ละระบบแฟ้มข้อมูล จะสามารถทราบถึงการทำงานของแต่ละตัวขับอุปกรณ์ เป็นข้อมูลช่วยในการสร้างสมดุลย์การทำงานในแต่ละระบบแฟ้มข้อมูล โดยการจัดระบบแฟ้มข้อมูลใหม่ เพื่อกระจายข้อมูลไปยังการทำงานในส่วนต่างๆ ของจานแม่เหล็กอย่างเท่าเทียม

2.4.3.3 ตรวจสอบแฟ้มข้อมูลและไดเรคทอรี่ที่มีขนาดใหญ่เกินปกติ

การดำเนินงานของระบบในแต่ละวันจะเกิดแฟ้มข้อมูลขึ้นในระบบมากมาย ซึ่งแฟ้มข้อมูลเหล่านี้จะมีขนาดเพิ่มขึ้น มีผลทำให้ระบบเสียเนื้อที่สำหรับการทำงาน เช่น แฟ้มข้อมูล /usr/adm/wtmp ซึ่งเก็บข้อมูลการเข้าสู่ระบบ หรือแฟ้มข้อมูล /usr/spool/uucp ซึ่งเก็บประวัติการส่งผ่านแฟ้มข้อมูลระหว่างระบบคอมพิวเตอร์ ดังนั้นจึงควรตรวจสอบขนาดของแฟ้มข้อมูลไม่ให้มีขนาดใหญ่เกินไป ตัวอย่างการใช้คำสั่งเพื่อจำกัดขนาดของแฟ้มข้อมูลเหล่านี้

```
$ tail -50 /usr/adm/wtmp > /tmp/wtmp
$ mv /tmp/wtmp /usr/adm/wtmp
```

คำสั่งแรกเป็นการนำข้อมูลที่เกิดขึ้น 50 บรรทัดหลังสุด เก็บลงในไดเรคทอรี่ชั่วคราว คำสั่งที่สองนำแฟ้มข้อมูลคืนเก็บในไดเรคทอรี่เดิม ซึ่งขนาดของแฟ้มข้อมูลจะลดลง ซึ่งถ้าปฏิบัติกับทุกๆ แฟ้มข้อมูลจะเป็นวิธีหนึ่งในการดูแลรักษาเนื้อที่ว่างของระบบแฟ้มข้อมูลนั้นได้

2.4.3.4 ตรวจสอบเปอร์เซนต์การใช้งานของเนื้อที่บนจานแม่เหล็ก

การตรวจสอบเปอร์เซนต์การใช้งานของเนื้อที่บนจานแม่เหล็ก เพื่อตรวจสภาพความจุของเนื้อที่บนจานแม่เหล็ก ว่ามีการใช้งานไปแล้วเท่าใด และมีเนื้อที่ที่สามารถใช้งานได้เหลืออยู่กี่เปอร์เซนต์ ซึ่งสามารถเลือกใช้คำสั่งจากตัวอย่างในตารางที่ 2.3 และตารางที่ 2.4 ระบบแฟ้มข้อมูลของรูท (/) และระบบแฟ้มข้อมูล /usr ควรมีเนื้อที่เหลือสำหรับการใช้งานมากกว่า 50 เปอร์เซนต์ เพราะระบบแฟ้มข้อมูลทั้งสองใช้ในการเก็บข้อมูล ที่เกิดจากการทำงานของโปรแกรมระบบ เช่น การส่งแฟ้มข้อมูลผ่านเครือข่าย การสร้างแฟ้มข้อมูลชั่วคราวเพื่อการพิมพ์ และการทำบัญชีระบบ

ตารางที่ 2.2 ตัวอย่างการทำงานของคำสั่ง df -t

```
$ df -t

/          (/dev/dsk/c1t1d0s0):    44844 blocks     3214 i-nodes
                  total:    80108 blocks    4992 i-nodes
/usr       (/dev/dsk/c1t1d0s2):    51852 blocks     5771 i-nodes
                  total:   300312 blocks   18752 i-nodes
/usr2      (/dev/dsk/c1t1d0s8):    40080 blocks     7226 i-nodes
                  total:   180728 blocks   22576 i-nodes
/usr3      (/dev/dsk/c1t4d0s8):    18526 blocks    41959 i-nodes
                  total:   629376 blocks   65488 i-nodes
/usr10     (/dev/dsk/c6t2d0s8):   194266 blocks    55414 i-nodes
                  total:  1014300 blocks   65488 i-nodes
/usr12     (/dev/dsk/c6t2d0s9):   337852 blocks    32803 i-nodes
                  total:  1013248 blocks   63328 i-nodes
```

ตารางที่ 2.3 ตัวอย่างการทำงานของคำสั่ง sysadm diskuse

```
# sysadm  diskuse

Running subcommand 'diskuse' from menu 'filemgmt',
FILE MANAGEMENT

FILE SYSTEM USAGE AS OF 03/01/96 17:05:36

File        Free        Total       Percent
System      Blocks      Blocks      Full
------      ------      ------      -------
/           44844       80108       44%
/usr        51856       300312      82%
/usr10      194266      1014300     80%
/usr12      337852      1013248     66%
/usr2       40080       180728      77%
/usr3       18526       629376      97%
```

2.4.3.5 ตรวจสอบแฟ้มข้อมูลที่ไม่มีการเคลื่อนไหว

การลบแฟ้มข้อมูลที่ไม่มีการเคลื่อนไหว หรือไม่มีการใช้งาน โดยใช้คำสั่ง find ช่วยในการตรวจสอบที่อยู่ของแฟ้มข้อมูลที่มีเงื่อนไขตามที่กำหนดในคำสั่ง คำสั่งนี้จะสำรวจ

แฟ้มข้อมูลในแต่ละไดเรคทอรี่ โดยสามารถระบุเงื่อนไขในการค้นหาได้ เงื่อนไขในการค้นหาสามารถกำหนดตามการอนุญาตของแฟ้มข้อมูล ตามวันที่ที่แก้ไขแฟ้มข้อมูล ตามขนาดของแฟ้มข้อมูล ตามชื่อผู้สร้างแฟ้มข้อมูลนั้น ตามชนิดของแฟ้มข้อมูล ตามชื่อของแฟ้มข้อมูลนั้น หรือตามจำนวนการเชื่อมโยงก็ได้ และเมื่อค้นพบแฟ้มข้อมูลตามเงื่อนไขแล้ว สามารถกำหนดให้ทำงานบางอย่างได้

ในกรณีนี้ต้องการแฟ้มข้อมูลซึ่งไม่มีการเคลื่อนไหว คือใช้เงื่อนไขของเวลาในการเรียกใช้ข้อมูล ตัวอย่างคำสั่งในการค้นหาแฟ้มข้อมูลที่ไม่เคลื่อนไหวตลอดระยะเวลา 6 เดือน เมื่อค้นพบแฟ้มข้อมูลดังกล่าวให้ระบบเปิดโอกาสในการตัดสินใจในการเก็บแฟ้มข้อมูลนั้นไว้ก่อน หรือทำการ ลบแฟ้มข้อมูลนั้นออกจากระบบ

```
$ find / -atime +180 -ok rm {} \;
```

2.4.3.6 ตรวจสอบผู้ใช้ที่ใช้เนื้อที่บนจานแม่เหล็กมาก

เพื่อให้ข่าวสารแก่ผู้ใช้งานปลายทาง ในการตรวจสอบแฟ้มข้อมูลที่สร้างขึ้นภายใต้ไดเรคทอรี่ของตนเองว่ามีแฟ้มข้อมูลขยะที่สามารถลบออกจากระบบได้หรือไม่ ตัวอย่างคำสั่งที่ใช้ในการทำงาน

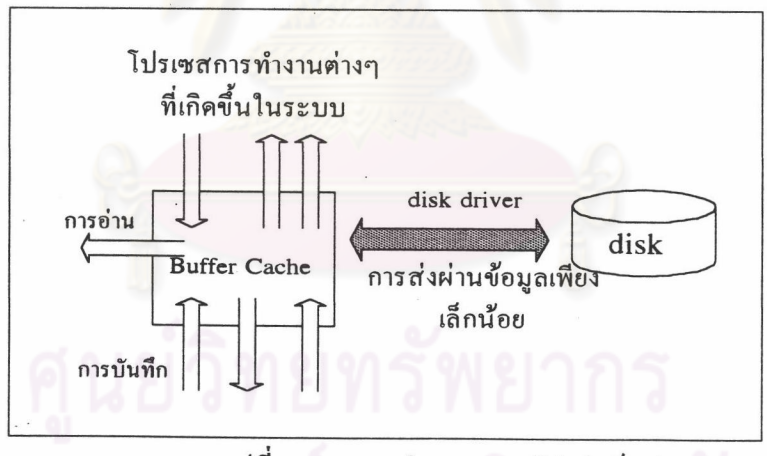
```
$ du /user/home
$ find /user/home -size +10 -print
# find / -name a.out -o -name core -atime +10 -exec rm {} \;
# find / -type d size +40 -print
```

คำสั่งแรกเป็นการแสดงจำนวนบล็อคของการเก็บข้อมูลภายใต้ไดเรคทอรี่ /user/home คำสั่งที่ 2 แสดงรายการของแฟ้มข้อมูลที่มีขนาดมากกว่า 10 บล็อค ภายใต้ไดเรคทอรี่ /user/home คำสั่งที่ 3 เป็นการทำงานในสถานะผู้ใช้พิเศษโดยค้นหาแฟ้มข้อมูลชื่อ a.out หรือแฟ้มข้อมูลชื่อ core ซึ่งไม่มีการเปลี่ยนแปลงใดๆ ในระยะเวลา 10 วัน เมื่อพบแฟ้มข้อมูลซึ่งเข้ากับเงื่อนไขดังกล่าวระบบจะลบแฟ้มข้อมูลนั้นทันที สำหรับคำสั่งที่ 4 เป็นคำสั่งเพื่อตรวจสอบไดเรคทอรี่ ที่มีขนาด 40 บล็อค การค้นหาข้อมูลในไดเรคทอรี่ที่มีขนาดใหญ่อาจมีผลต่อความสามารถของระบบได้ การลบแฟ้มข้อมูลออกจากไดเรคทอรี่ ไม่ทำให้ขนาดของไดเรคทอรี่นั้นลดลง เพราะเมื่อแฟ้มข้อมูลถูกลบออกจากไดเรคทอรี่ ค่าไอโหนดของแฟ้มข้อมูลนั้นถูกเปลี่ยนเป็นค่า Null คือมีขนาดเท่าเดิมแต่ไม่มีค่าข่าวสารของแฟ้มข้อมูลใดๆ เมื่อลบแฟ้มข้อมูลออกจากไดเรคทอรี่อีก เนื้อที่ของส่วนนี้ก็จะเพิ่มขนาดขึ้น การแก้ไขในกรณีที่เป็นไดเรคทอรี่ โดยย้ายแฟ้มข้อมูลภายใต้ไดเรคทอรี่

นั้นเก็บลงในไดเรคทอรี่ที่สร้างขึ้นใหม่ แล้วลบไดเรคทอรี่เดิมออกจากระบบ แต่กรณีเป็นระบบแฟ้มข้อมูลทั้งระบบควรจัดระบบการทำงานใหม่เหมือนกับการทำงานในหัวข้อ การควบคุมการแตกกระจายของข้อมูล

2.4.3.7 การตรวจสอบการทำงานของบัฟเฟอร์แคช

ระบบแฟ้มข้อมูลมีหน้าที่ในการจัดการทำงานของบัฟเฟอร์ ของส่วนอุปกรณ์รับเข้า และส่งออก เรียกว่าบัฟเฟอร์แคชดังได้กล่าวแล้วข้างต้น ซึ่งเป็นแนวทางหนึ่งในการเพิ่มประสิทธิภาพการทำงานของหน่วยความจำหลัก ช่วยให้ระบบปฏิบัติการยูนิกซ์อ่าน และบันทึกข้อมูลการทำงานในระบบอย่างมีประสิทธิภาพ โดยจัดให้การอ่านและบันทึกข้อมูลในจานแม่เหล็กมีการทำงานในลักษณะแบบแบทช์ โดยใช้วิธีการอ่านก่อน (Read-ahead) และการบันทึกทีหลัง (Write-behind) การอ่านก่อนมีสมมุติฐานว่าผู้ใช้งานในระบบมีแนวโน้มที่จะอ่านข้อมูลซึ่งต่อเนื่องกันไปคราวละมากๆ ทำให้ระบบนำข้อมูลที่คาดว่ากำลังจะถูกอ่าน ส่งผ่านเข้าสู่บัฟเฟอร์แคช เมื่อโปรแกรมเรียกโปรเซสการอ่านข้อมูล ระบบจะตรวจสอบว่าข้อมูลที่ต้องการนั้น เก็บอยู่ในบัฟเฟอร์แคชหรือไม่ กรณีที่ระบบตรวจพบข้อมูลในบัฟเฟอร์แคช สามารถส่งผ่านข้อมูลนั้นให้แก่โปรแกรมที่ต้องการอย่างรวดเร็ว โดยไม่ต้องเสียเวลารอคอยการทำงานของหัวอ่าน บันทึก



รูปที่ 2.16 การทำงานของบัฟเฟอร์แคช

กรณีของการบันทึกทีหลัง เมื่อโปรแกรมเรียกการบันทึกข้อมูล ระบบจะทำการบันทึกข้อมูลนั้นลงในส่วนของบัฟเฟอร์แคชนี้ ดังนั้นการบันทึกข้อมูลจากแต่ละโปรแกรมที่ทำงานในระบบจะเก็บรวมในส่วนนี้ จนขณะหนึ่งที่เหมาะสมระบบจึงนำข้อมูลนี้ส่งผ่านการทำงานให้แก่ตัวขับอุปกรณ์เพื่อเก็บข้อมูลนั้นลงในจานแม่เหล็ก จะเห็นว่าการทำงานของบัฟเฟอร์แคช เป็นการแปลงการทำงานของส่วนรับเข้า และส่งออกจากโปรเซสที่เกิดขึ้นในระบบ รวบรวมและส่งผ่านการทำงานให้กับตัวขับอุปกรณ์ในคราวเดียว ดังรูปที่ 2.16

นอกจากที่ได้กล่าวแล้ว ต้องพิจารณาถึงระยะเวลาในการนำข้อมูลจากบัฟเฟอร์ บันทึกเก็บลงในจานแม่เหล็กเพื่อปรับปรุงข้อมูลที่เก็บอยู่นั้น ถ้ากำหนดให้ค่าของระยะเวลาน้อย ระบบจะบันทึกข้อมูลทุกช่วงระยะเวลานั้น ซึ่งอาจทำให้การทำงานของแคชไม่ช่วยเพิ่มประสิทธิภาพใดๆ แก่ระบบ และอาจเป็นการเพิ่มงานให้แก่ระบบโดยไม่จำเป็น เพราะระบบสามารถเก็บข้อมูลลงในจานแม่เหล็กได้โดยตรง แต่ถ้ากำหนดให้ระยะเวลามีค่ามาก ความเสี่ยงในการสูญเสียข้อมูลในบัฟเฟอร์แคชจะเพิ่มขึ้นเพราะอาจเกิดความล้มเหลวขึ้นในระบบในขณะใดๆ ก็ได้ ทำให้ความน่าเชื่อถือของระบบลดลง การกำหนดระยะเวลาของการปรับปรุงข้อมูลในจานแม่เหล็กนี้ กำหนดโดยค่าพารามิเตอร์ระบบ (ภาคผนวก ก.) คือค่า BDFLUSHR และค่า NAUTOUP

2.4.4 สมรรถนะของระบบเครือข่าย

สมรรถนะการทำงานของระบบเครือข่ายเกี่ยวข้องกับการทำงานของระบบเครื่องคอมพิวเตอร์มากกว่า 1 ระบบ ปัญหาที่เกิดขึ้นคือปัญหาของการสื่อสารข้อมูล ระหว่างระบบเครื่องคอมพิวเตอร์ท้องถิ่น และระบบเครื่องคอมพิวเตอร์ทางไกล คือ นอกจากต้องพิจารณาระบบคอมพิวเตอร์แต่ละระบบแล้ว ยังต้องดูแลถึงระบบของการสื่อสารข้อมูลระหว่างกันอีกด้วย

2.4.4.1 การเชื่อมโยงระหว่างระบบคอมพิวเตอร์ทำได้ใน 3 ลักษณะคือ

2.4.4.1.1 การเชื่อมโยงโดยตรง (Direct Mode) เป็นการเชื่อมโดยตรงจากสายอนุกรม ของระบบท้องถิ่นไปยังสายอนุกรมของระบบทางไกล การเชื่อมโยงลักษณะนี้สามารถติดต่อสื่อสาร ด้วยความเร็วประมาณ 19,200 บิท ต่อวินาที (bit per second) ใช้กับระบบเครื่องคอมพิวเตอร์ซึ่งอยู่ห่างกันไม่เกิน 15 เมตร เพราะอาจเกิดสัญญาณรบกวน

2.4.4.1.2 การเชื่อมโยงแบบเครือข่ายท้องถิ่น (Local Area Network) เป็นการต่อเชื่อมโยงของระบบคอมพิวเตอร์ผ่านอุปกรณ์ชุมสาย (Hub) ภายในอาคารเดียวกัน สามารถติดต่อสื่อสารด้วยความเร็วมากประมาณ 10 ล้านบิท ต่อวินาที

2.4.4.1.3 การเชื่อมโยงแบบเครือข่ายระยะกว้าง (Wide Area Network) เป็นการต่อเชื่อมโยงของระบบคอมพิวเตอร์ที่อยู่ห่างไกลกัน เช่น ระหว่างจังหวัด หรือระหว่างประเทศ โดยผ่านอุปกรณ์แปลงสัญญาณข้อมูล ความเร็วของการติดต่อสื่อสาร ขึ้นอยู่กับความเร็วของอุปกรณ์แปลงข้อมูล

2.4.4.2 ปัญหาการใช้งานของระบบเครือข่าย มี 2 รูปแบบ

2.4.4.2.1 การทำงานบนระบบเครือข่ายมีมากเกินไป คือจำนวนข้อมูลที่ต้องการส่งมีมากเกินกว่าความจุของระบบเครือข่าย มีผลทำให้ส่งข้อมูลได้ช้า การแก้ไขต้องจัดระบบ โครงสร้างเครือข่ายใหม่ เช่น การแบ่งแยกให้เป็นระบบเครือข่ายย่อย เป็นต้น

2.4.4.2.2 ข้อมูลที่ได้รับจากระบบเครือข่ายไม่ถูกต้อง เกิดจากการทำงานของเครือข่ายขัดข้อง หรือทำงานเป็นช่วงๆ และเพื่อให้ได้รับข้อมูลที่ถูกต้อง ระบบเครือข่ายจะต้องส่งข้อมูลนั้นหลายครั้ง ทำให้งานอื่นทำงานช้าลงไปด้วย การแก้ไขต้องระบุให้ได้ว่าส่วนใดของระบบเครือข่ายทำงานไม่ถูกต้องและเปลี่ยนเฉพาะส่วนนั้น

ปัญหาของสมรรถนะการทำงานของระบบเครือข่าย สังเกตจากความเร็วของการทำงานผ่านเครือข่าย เปรียบเทียบกับการทำงานของระบบปกติ เช่นการเรียกใช้แฟ้มข้อมูล หรือการเข้าสู่ระบบ ถ้าใช้เวลาในการทำงานต่างกันมากแสดงถึงปัญหาการทำงานของระบบเครือข่าย สำหรับแนวทางเพื่อการแก้ไข ควรคำนึงถึงลักษณะงานที่ใช้ผ่านเครือข่าย โดยควรหลีกเลี่ยงการส่งผ่านแฟ้มข้อมูลขนาดใหญ่ การเรียกใช้การสำรองข้อมูลผ่านเครือข่าย

2.4.5 สมรรถนะของอุปกรณ์

ระบบยูนิกซ์ให้ความสำคัญกับการทำงานของอุปกรณ์จอภาพมาก เพราะมีลักษณะการทำงานแบบโต้ตอบทันที ปัญหาที่เกิดขึ้นจึงมักเกี่ยวเนื่องการรบกวนจากสายอุปกรณ์ อย่างไรก็ตามในกรณีที่ผู้ใช้งานในระบบรายงานว่า การโต้ตอบของระบบเครื่องคอมพิวเตอร์ช้าลง การพิจารณาโดยตรวจสอบจากปริมาณงาน และพฤติกรรมของระบบว่ามีการสับค่าหน้ามากน้อยเพียงใด การทำงานนั้นเป็นการทำงานของผู้ใช้งานในระบบท้องถิ่นหรือระบบทางไกล

นอกจากนั้นการทำงานของจอภาพในระบบยูนิกซ์ ซิสเต็มไฟว์ มีการทำงานโดยอาศัยสตรีม ดังนั้นถ้าการทำงานของสตรีมมีปัญหา มีผลถึงการทำงานของสายอุปกรณ์ด้วย การพิจารณาถึงสมรรถนะของสายอุปกรณ์ในส่วนที่เกี่ยวข้องกับสตรีมควรพิจารณาถึงค่าพารามิเตอร์ที่ใช้ควบคุมการทำงานของสตรีมดังต่อไปนี้

2.4.5.1 จำนวนของบัฟเฟอร์ขนาดเล็กมีจำนวนเพียงพอสอดคล้องกับจำนวนจอภาพในระบบหรือไม่ ค่านี้กำหนดโดยค่าของพารามิเตอร์ระบบ NBLK4 NBLK16 NBLK64 และ NBLK128 ค่าเหล่านี้ต้องตรวจสอบทุกครั้งที่มีการเพิ่มจำนวนจอภาพ หรืออุปกรณ์โมเด็ม

2.4.5.2 การทำงานของจอภาพเสมือนซึ่งเป็นการทำงานผ่านเครือข่าย ต้องการโครงสร้างของสตรีม 1 ค่า ต่อการทำงานของแต่ละจอภาพ ซึ่งกำหนดโดยค่าพารามิเตอร์ระบบ NSTREAM ดังนั้นค่านี้ต้องเพียงพอสำหรับการใช้งานของผู้ใช้ในระบบท้องถิ่น และระบบทางไกล

2.4.5.3 จำนวนลำดับการทำงานของสตรีม กำหนดโดยค่าพารามิเตอร์ NQUEUE มีเพียงพอหรือไม่ โดยทั่วไปค่าสตรีม 1 ค่าต้องการลำดับการทำงาน 4 ค่า ใช้สำหรับสตรีม 2 ค่า และสำหรับตัวขับอุปกรณ์ 2 ค่า