บทที่ 3

ระบบคอมพิวเตอร์แบบพร้อมใช้สูง

ระบบคอมพิวเตอร์แบบพร้อมใช้งานสูงเป็นระบบคอมพิวเตอร์ที่ออกแบบมาเพื่อแก้ไขปัญหาที่เกิดขึ้นในบทที่ 2 อันเนื่องมาจากใช้เวลามากในการโรลล์แบคเวิร์ดหรือโรลล์ฟอร์เวิร์ดหลังจากที่ระบบเกิดความล้มเหลวขึ้น ระบบนี้ประกอบไปด้วยส่วนสำคัญดังนี้คือ

3.1 ระบบการตรวจจับความล้มเหลว

3.2 ระบบเครือข่าย

3.3 รีดันแดนซ์อะเรย์ออฟอินเอ็กซ์เพนซีฟดิสก์ (เรดดิสก์) ระดับ 5

3.4 การจัดระบบโปรแกรมประยุกต์ในเครื่องคอมพิวเตอร์

3.5 สภาวะแวดล้อมของระบบ

ส่วนประกอบทั้ง 5 ส่วนทำงานประสานกันเพื่อทำให้ระบบดังกล่าวมีความพร้อมใช้งานสูงเพื่อที่สามารถควบคุมสายการผลิตให้เป็นไปได้อย่างมีประสิทธิภาพ

3.1 ระบบการตรวจจับความล้มเหลว

ระบบการตรวจจับความล้มเหลวนั้นเป็นหัวใจของระบบคอมพิวเตอร์แบบพร้อมใช้งานสูง เพราะว่า ถ้าเราสามารถทราบถึงความล้มเหลวได้เร็วเท่าใด ค่าเสียหายที่เกิดจากความล้มเหลวนั้นจะลดน้อยลงไปด้วย แต่อย่างไรก็ตามต้องตรวจสอบยืนยันให้แน่นอนว่าความล้มเหลวนั้น เกิดขึ้นจริงกับคอมพิวเตอร์ฮาร์ดแวร์

ระบบการตรวจจับความล้มเหลวมีองค์ประกอบที่สำคัญดังนี้คือ

3.1.1 โปรแกรมตรวจจับความล้มเหลว

โปรแกรมตรวจจับความล้มเหลวของระบบ ใช้ในการตรวจจับความล้มเหลวของระบบที่เกิดจากส่วนของฮาร์ดแวร์ เช่น หน่วยประมวลผลกลาง แผงวงจรหลัก (main board) เป็นต้น ซึ่งเป็นผลให้เครื่องคอมพิวเตอร์หยุดทำงานก่อให้เกิดความเสียหาย ต่อระบบสายการผลิตที่ยังดำเนินอยู่

หลักการในการตรวจจับของโปรแกรม แบ่งออกเป็น 2 ส่วนคือ

ก. การตรวจจับภายในเครื่องคอมพิวเตอร์ (process-to-process checking)

การตรวจจับภายในเครื่องคอมพิวเตอร์ เป็นการตรวจจับว่าโปรแกรมที่ทำการตรวจจับระหว่างเครื่องนั้น ยังทำงานอยู่หรือไม่

ข. การตรวจจับระหว่างเครื่องคอมพิวเตอร์ (host-to-host checking)

การตรวจจับระหว่างเครื่องคอมพิวเตอร์ เป็นการตรวจจับว่าโฮสต์ต่างๆ ในระบบยังคงทำงานอยู่หรือไม่

3.1.2 องค์ประกอบของโปรแกรมการตรวจจับความล้มเหลว

ก. โปรเซสควบคุม (control process) ทำหน้าที่

1. ติดต่อกับผู้ควบคุมระบบเพื่อรับคำสั่งต่างๆ
2. แสดงสถานะภาพของระบบตรวจจับความล้มเหลว
3. สร้างโปรเซสตรวจสอบ
4. ตรวจสอบเครื่องคอมพิวเตอร์ที่สงสัยว่าเกิดความล้มเหลว เพื่อยืนยันความล้มเหลวนั้น

ข. โปรเซสตรวจสอบตัวที่หนึ่ง (primary check process) ทำหน้าที่

1. ตรวจสอบไปยังโปรเซสควบคุมว่ายังทำงานอยู่หรือไม่
2. ตรวจสอบไปยังโปรเซสตรวจสอบตัวที่สองว่ายังทำงานอยู่หรือไม่
3. ตรวจสอบไปยังโฮสต์ที่กำหนดว่ายังทำงานอยู่หรือไม่
4. ติดต่อกับผู้ควบคุมระบบ เพื่อรับคำสั่งในการทำงาน

ค. โปรเซสตรวจสอบตัวที่สอง (secondary check process) ทำหน้าที่

1. ตรวจสอบไปยังโปรเซสควบคุมว่ายังทำงานอยู่หรือไม่
2. ตรวจสอบไปยังโปรเซสตรวจสอบตัวที่หนึ่งว่ายังทำงานอยู่หรือไม่
3. ตรวจสอบไปยังโฮสต์ที่กำหนดว่ายังทำงานอยู่หรือไม่
4. ติดต่อกับผู้ควบคุมระบบ เพื่อรับคำสั่งในการทำงาน

3.1.3 การทำงานของโปรแกรมการตรวจจับความล้มเหลว

การทำงานจะเริ่มจากการสร้าง โปรเซสควบคุมซึ่งผู้ควบคุมระบบเป็นผู้สั่งการ หลังจากนั้นโปรเซสควบคุมจะสร้างโปรเซสตรวจสอบทั้งสอง เมื่อโปรเซสตรวจสอบเริ่มทำงานจะทำการตรวจสอบไปยังจุดที่ถูกกำหนด แล้วส่งผลการสอบมาให้โปรเซสควบคุม โปรเซสควบคุมจะนำผลการตรวจสอบที่ได้รับ มาทำการตรวจสอบว่าเกิดความล้มเหลวขึ้นในระบบหรือไม่ ถ้ามีความล้มเหลวเกิดขึ้นจะทำการตรวจสอบเพื่อยืนยันความล้มเหลวนั้น แล้วแจ้งให้ผู้ควบคุมระบบทราบเพื่อดำเนินการต่อไป แต่ถ้าไม่มีความล้มเหลวเกิดขึ้นจะแสดงสถานะการทำงานของระบบและผลการตรวจสอบ

3.1.4 เครื่องมือที่ใช้ในการสร้างโปรแกรมการตรวจจับความล้มเหลว

เครื่องที่ใช้ในการสร้างโปรแกรมการตรวจจับความล้มเหลว อาศัยสิ่งที่มีอยู่แล้วในระบบยูนิกซ์ เช่น เมซเซสคิว อาร์พีซี เป็นต้น ซึ่งจะกล่าวในหัวข้อต่อไปนี้

ก. เมซเซสคิว (message queue)

เมซเซสคิวใช้ในการติดต่อระหว่างโปรเซสที่อยู่ในโฮสต์เดียวกัน เพื่อทำการส่งข้อมูลต่างๆ ในการทำงาน รวมถึงการใช้เป็นเครื่องมือเพื่อตรวจว่าโปรเซสนั้นๆ ยังทำงานอยู่หรือไม่

ข. รีโมทโพรซีเยอร์คอลล์ หรือ อาร์พีซี (remote procedure call)

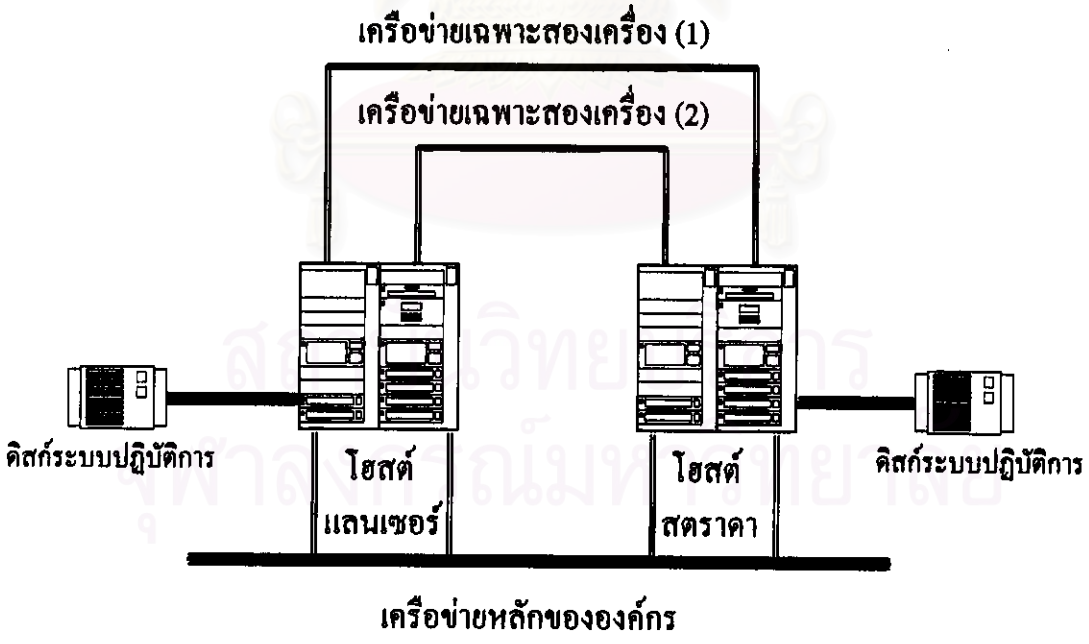
อาร์พีซีใช้ในการตรวจสอบว่าโฮสต์ที่กำหนดนั้นยังทำงานอยู่หรือไม่ โดยโปรเซสตรวจสอบใช้อาร์พีซีเป็นเครื่องมือในการตรวจการทำงานของโฮสต์ที่กำหนด

3.1.5 โครงร่างของระบบการตรวจจับความล้มเหลว

ถ้าเรามองโครงของระบบคอมพิวเตอร์โดยทั่วไปแล้ว จะเห็นว่าเราสามารถแบ่งออกได้เป็น 2 ส่วน คือ โครงร่างภายนอกและโครงร่างภายใน เช่น โครงร่างของฮาร์ดแวร์และโครงร่างของซอฟต์แวร์ เป็นต้น ระบบการตรวจจับความล้มเหลวนั้น จะใช้การมองโครงร่างเหมือนกับการมองระบบคอมพิวเตอร์ทั่วไปๆ ซึ่งสามารถแบ่งออกได้ดังนี้คือ

ก. โครงร่างภายนอกระบบการตรวจจับความล้มเหลว

จากรูปที่ 3.1 แสดงให้เห็นถึงรูปลักษณะภายนอกของระบบตรวจจับความล้มเหลว ระหว่างโฮสต์ 2 โฮสต์ โดยอาศัยเครือข่ายหลักและเครือข่ายเฉพาะเครื่อง เป็นช่องทางในการตรวจสอบระหว่างกันและกัน

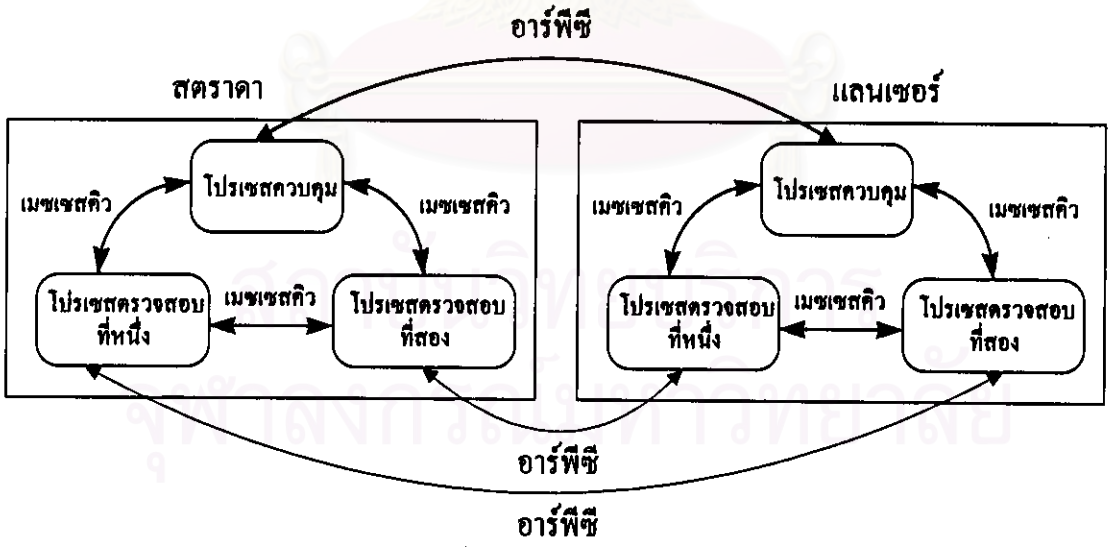


รูปที่ 3.1 โครงร่างภายนอกของระบบตรวจจับความล้มเหลว

ข. โครงร่างภายในของระบบการตรวจจับความล้มเหลว

จากรูปที่ 3.2 แสดงให้เห็นถึงโปรเซสต่างๆ และเครื่องมือที่โปรเซสใช้ในการตรวจจับความล้มเหลวของระบบ เช่น การส่งเมซเซสจากโปรเซสควบคุมไปยังโปรเซสตรวจสอบเพื่อทดสอบว่าโปรเซสตรวจสอบยังทำงานอยู่หรือไม่ เมซเซสจะถูกส่งผ่านเครื่องมือที่เรียกว่า "เมซเซสคิว" จากโปรเซสควบคุมไปยังโปรเซสตรวจสอบในการตรวจสอบระหว่างเครื่องคอมพิวเตอร์ โปรเซสตรวจสอบในแต่ละโฮสต์จะทำหน้าที่ในการตรวจสอบ ตัวอย่างเช่น โฮสต์แลนเซอร์ต้องการตรวจสอบว่าโฮสต์สตราดายังทำงานอยู่หรือไม่ สามารถทำได้โดยโปรเซสตรวจสอบของโฮสต์แลนเซอร์ส่งอาร์พีซีไปยังโปรเซสตรวจสอบของโฮสต์สตราดา เมื่อโปรเซสตรวจสอบของโฮสต์สตราดาเมื่อได้รับอาร์พีซีจะทำงานตามที่อาร์พีซีกำหนดแล้วส่งผลลัพธ์กลับไปให้โปรเซสตรวจสอบของโฮสต์แลนเซอร์ทราบ การตรวจสอบนั้นอาศัยการโต้ตอบกันไปมาระหว่างโปรเซสตรวจสอบที่อยู่ต่างโฮสต์กัน คือถ้าโปรเซสตรวจสอบของโฮสต์ใดๆ ไม่มีการโต้ตอบไปยังโปรเซสตรวจสอบของอีกโฮสต์หนึ่ง เมื่อมีการร้องขอการทำงานผ่านอาร์พีซี แสดงว่าโฮสต์นั้นอาจหยุดทำงานไปแล้ว

การตรวจสอบว่าโฮสต์ที่กำหนดนั้นเกิดความล้มเหลวขึ้น อาศัยการตอบกลับของโฮสต์หลังจากที่โปรเซสตรวจสอบส่งอาร์พีซีไป ซึ่งทั้งนี้เงื่อนไขการที่จะตัดสินว่า โฮสต์นั้นเกิดความล้มเหลวคือ โปรเซสตรวจสอบที่หนึ่งและสอง ต้องมีผลการตรวจสอบที่สอดคล้องกัน จึงจะสรุปว่าโฮสต์นั้นเกิดความล้มเหลวขึ้นจริง



รูปที่ 3.2 โครงร่างภายในของระบบตรวจจับความล้มเหลว

3.1.6 อัลกอริทึมหลักของระบบตรวจจับความล้มเหลว

อัลกอริทึมที่ใช้ในระบบตรวจจับความล้มเหลวจะมี 4 ส่วนคือ

ก. อัลกอริทึมการเริ่มต้นทำงาน

ผู้ควบคุมระบบเป็นผู้สั่งการให้ระบบตรวจจับความล้มเหลวทำงาน โดยเริ่มจากการสร้างโปรเซสควบคุม สร้างโปรเซสตรวจที่หนึ่งและสอง จากนั้นโปรเซสควบคุมทำการตรวจจับภายในเครื่องคอมพิวเตอร์ และสั่งให้โปรเซสตรวจสอบที่หนึ่งและสองทำการตรวจจับระหว่างเครื่องคอมพิวเตอร์ด้วย ดังแสดงในรูปที่ 3.3

ข. อัลกอริทึมการตรวจจับภายในเครื่องคอมพิวเตอร์ (process-to-process checking)

อัลกอริทึมนี้ใช้ในการตรวจสอบว่าโปรเซสตรวจสอบและโปรเซสควบคุมยังทำงานอยู่หรือไม่ โดยการตรวจสอบเริ่มจากสร้างโปรเซสควบคุมและโปรเซสตรวจสอบหนึ่งและสอง จากนั้นโปรเซสควบคุมส่งข้อมูลผ่านเมซเซสคิวไปให้โปรเซสตรวจสอบหนึ่งและสอง เมื่อโปรเซสตรวจสอบหนึ่งและสอง ได้รับข้อมูลแล้วจะส่งข้อมูลตอบกลับมาให้โปรเซสควบคุมโดยผ่านเมซเซสคิวเช่นกัน

ตัวอย่างการตรวจสอบว่าโปรเซสใดๆ หยุดทำงานนั้นสามารถทำได้โดย (ขอยกตัวอย่างกรณีที่โปรเซสควบคุมตรวจสอบไปยังโปรเซส ตรวจสอบหนึ่งและสองว่ายังทำงานอยู่หรือไม่) หลังจากที่โปรเซสควบคุมส่งข้อมูลผ่านเมซเซสคิวแล้วจะรอรับคำตอบจากโปรเซสตรวจสอบหนึ่งและสองจนถึงเวลาที่กำหนด ถ้าโปรเซสตรวจสอบตัวใดไม่ส่งคำตอบมาแสดงว่าโปรเซสตรวจสอบนั้นอาจหยุดทำงานไปแล้ว ซึ่งต้องมีการตรวจสอบซ้ำอีกครั้งเพื่อให้แน่ใจว่า แล้วจึงดำเนินการต่อไป เช่น สร้างโปรเซสตรวจสอบตัวนั้นขึ้นใหม่ หรือแจ้งต่อผู้ควบคุมระบบเพื่อสั่งการ เป็นต้น ดังแสดงในรูปที่ 3.4

ค. อัลกอริทึมการตรวจจับระหว่างเครื่องคอมพิวเตอร์

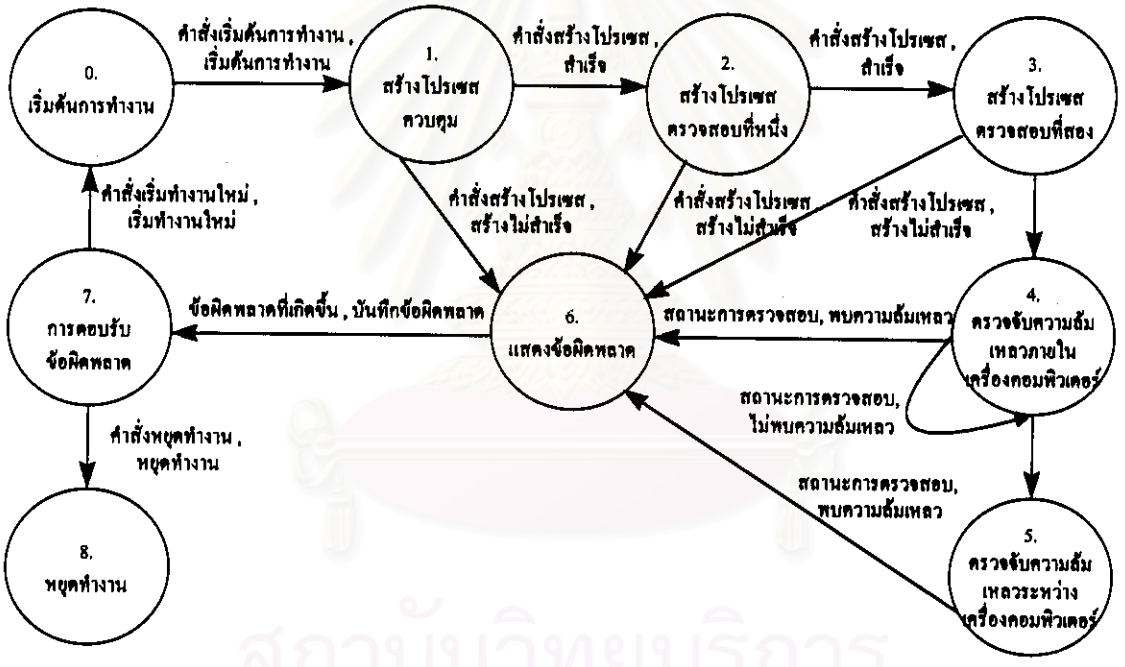
ในรูปที่ 3.5 แสดงถึงอัลกอริทึมการตรวจจับระหว่างเครื่องคอมพิวเตอร์ ซึ่งอัลกอริทึมนี้จะใช้โปรเซสตรวจสอบหนึ่งและสองของทั้งสองโฮสต์เป็นตัวทำงานคือ โปรเซสตรวจสอบที่หนึ่งและสองของโฮสต์แลนเซอร์ส่งอาร์พีซีไปที่โปรเซสหนึ่งและสองของโฮสต์สตราดาเมื่อโปรเซสตรวจสอบของโฮสต์สตราดาได้รับอาร์พีซี จะทำงานตามที่อาร์พีซีกำหนดแล้วส่งผลลัพธ์กลับไปให้โปรเซสตรวจสอบของโฮสต์แลนเซอร์ ในทำนองเดียวกันนั้น จะถูกกระทำบนโฮสต์สตราดาด้วย เพื่อเป็นการตรวจสอบซึ่งกันและกัน การตรวจสอบจะใช้การโต้ตอบกันไปมาของโปรเซสตรวจสอบของโฮสต์ทั้งสอง คือถ้าโปรเซสตรวจสอบบนโฮสต์ใดไม่มีการโต้ตอบก็แสดงว่าโฮสต์นั้นอาจเกิดความล้มเหลวขึ้น เมื่อเกิดกรณีนี้จะทำการส่งเมซเซสให้โปรเซสควบคุมทราบ เพื่อทำการตรวจสอบต่อไป

ง. อัลกอริทึมเพื่อตัดสินว่าเครื่องคอมพิวเตอร์เกิดความล้มเหลวขึ้นจริง

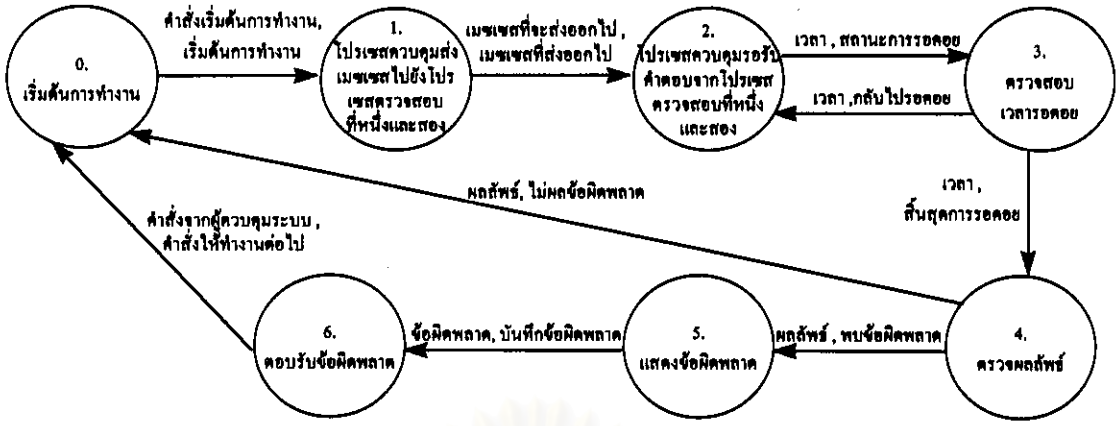
ในรูปที่ 3.6 เมื่อโปรเซสตรวจสอบพบว่าเกิดความล้มเหลวของระบบขึ้น จะส่งข้อมูลไปที่

โปรเซสควบคุม เมื่อโปรเซสควบคุมรับข้อมูลแล้วจะตรวจสอบซ้ำอีกครั้ง โดยการส่งอาร์พีซีผ่านเครือข่ายเฉพาะสองเครื่องที่หนึ่ง ไปยังโปรเซสควบคุมของโฮสต์ที่สงสัยว่าจะเกิดความล้มเหลวขึ้น ถ้าไม่มีการได้ตอบจากโปรเซสควบคุมของโฮสต์ที่สงสัย จะส่งอาร์พีซีผ่านเครือข่ายเฉพาะสองเครื่องที่สองไปยังโปรเซสควบคุมของโฮสต์ที่สงสัยอีกครั้งหนึ่ง ถ้าครั้งนี้ไม่มีการตอบสนองจากโปรเซสควบคุมของโฮสต์ที่สงสัย นั่นแสดงว่าโฮสต์ที่สงสัยนั้นเกิดความล้มเหลวขึ้นจริง จะแจ้งให้ผู้ควบคุมระบบทราบเพื่อดำเนินการต่อไป

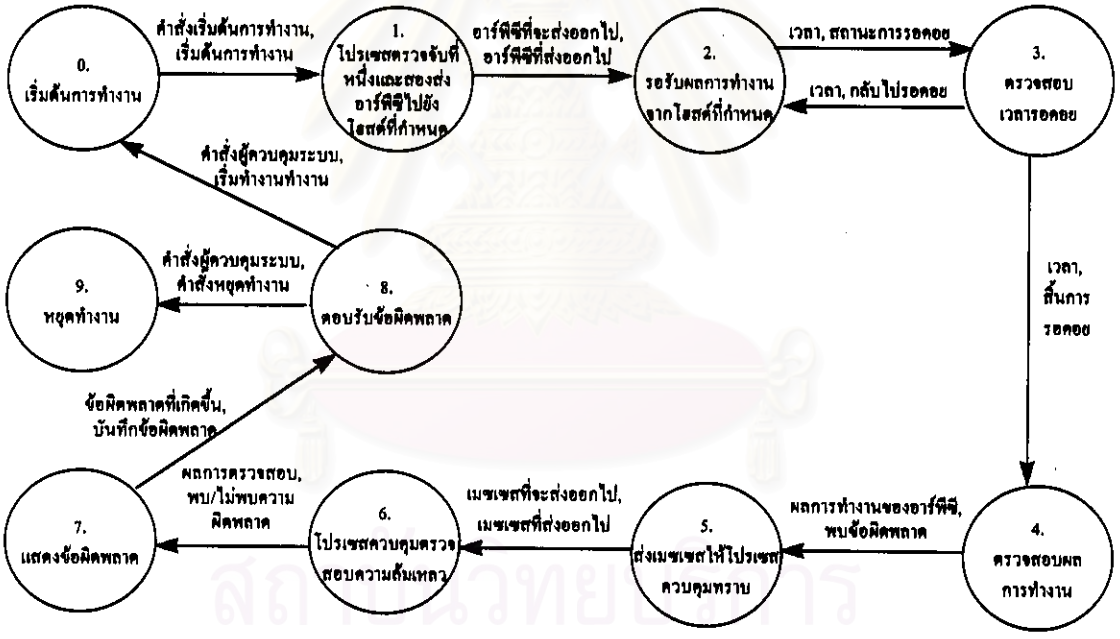
เหตุผลที่ต้องที่การส่งอาร์พีซีผ่านเครือข่ายเฉพาะสองเครื่องที่หนึ่งและสองนั้นเพราะว่าในบางกรณีเครือข่ายหลักขององค์กรอาจเกิดความขัดข้อง ซึ่งเป็นผลให้การตรวจจับพบว่าโฮสต์เกิดความล้มเหลวขึ้น ซึ่งสาเหตุไม่ได้มาจากระบบคอมพิวเตอร์ในส่วนของฮาร์ดแวร์ แต่เป็นเพราะระบบเครือข่ายหลักขององค์กรเกิดขัดข้องขึ้น อาจทำให้การตัดสินใจทำงานในขั้นตอนต่อไปผิดพลาดได้



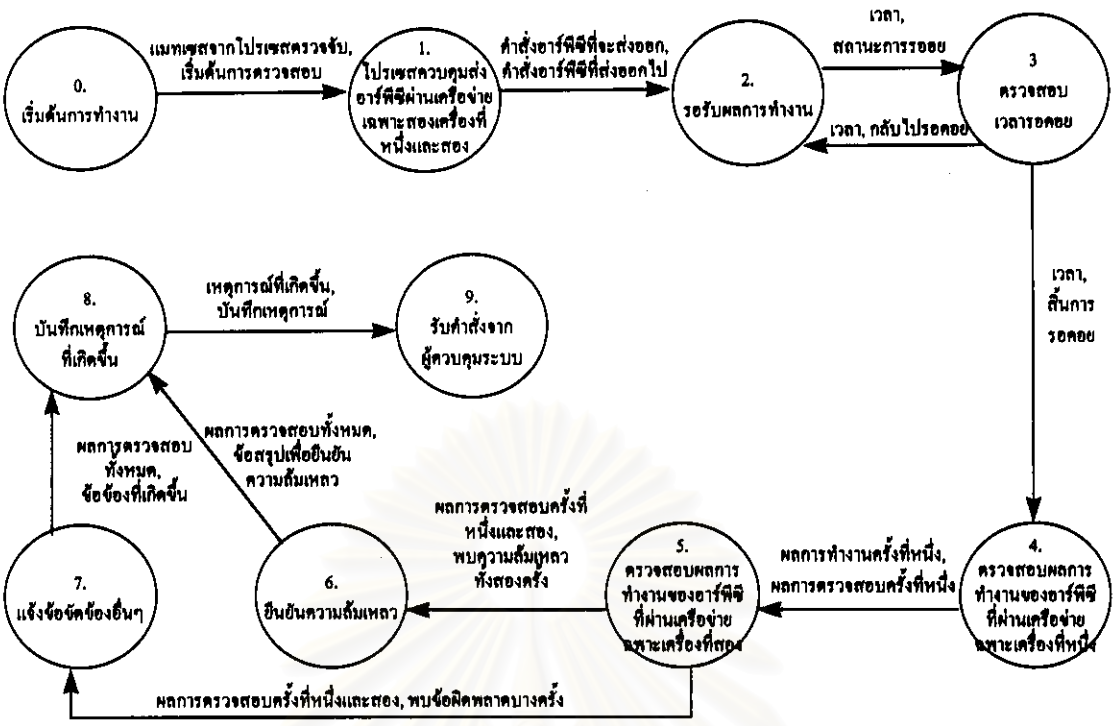
รูปที่ 3.3 แผนผังสถานะแสดงถึงอัลกอริทึมการเริ่มต้นการทำงาน



รูปที่ 3.4 แผนผังสถานะของอัลกอริทึมการตรวจจับภายในเครื่องคอมพิวเตอร์



รูปที่ 3.5 แผนผังสถานะแสดงถึงอัลกอริทึมการตรวจสอบระหว่างเครื่องคอมพิวเตอร์



รูปที่ 3.6 แผนผังสถานะแสดงถึงอัลกอริทึมเพื่อตัดสินว่าเครื่องคอมพิวเตอร์เกิดความล้มเหลวขึ้นจริง

3.2 ระบบเครือข่าย

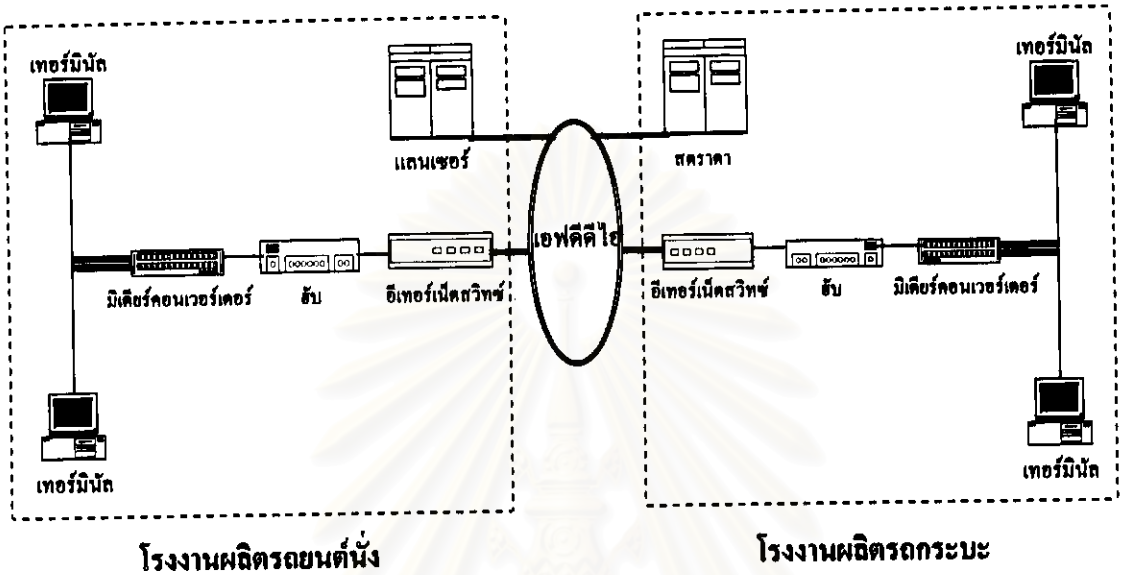
ระบบเครือข่ายที่ใช้กับระบบคอมพิวเตอร์แบบพร้อมใช้งานสูงนั้น ต้องทนต่อความล้มเหลวได้สูงเท่ากับระบบคอมพิวเตอร์ เพราะว่าถ้าระบบเครือข่ายเกิดความเสียหายขึ้นแล้วไม่มีประโยชน์อะไรที่จะรักษาเครื่องคอมพิวเตอร์ให้คงอยู่ งานที่กำลังทำอยู่จะหยุดไปด้วย ดังนั้นระบบเครือข่ายมีความสำคัญไม่ยิ่งหย่อนไปกว่าระบบคอมพิวเตอร์ ต่อไปจะได้กล่าวถึงระบบเครือข่ายที่ได้ออกแบบมา

3.2.1 โครงร่างของระบบเครือข่าย

ระบบเครือข่ายนี้ใช้เครือข่ายหลักเป็นเอฟดีดีไอ โดยต่อเชื่อมโฮสต์ทั้งสองตัวเข้ากับเอฟดีดีไอริงค์ การเชื่อมต่อระหว่างโฮสต์ไปยังเทอร์มินัลที่ใช้งาน สามารถทำได้โดยผ่านอีเทอร์เน็ตสวิทซ์ที่เชื่อมต่อกับฮับ จากนั้นเพื่อให้สามารถใช้สายออปติกไฟเบอร์เชื่อมต่อไปยังเทอร์มินัลได้ ต้องใช้อุปกรณ์ที่ชื่อว่า มิเดียคอนเวอร์เตอร์ ทำการเปลี่ยนสัญญาณไฟฟ้าให้เป็นสัญญาณแสง แล้วส่งผ่านไปยังเทอร์มินัลเพื่อใช้งานต่อไป

เหตุผลที่ใช้สายออปติกไฟเบอร์ต่อไปยังเทอร์มินัลคือ ในสายการผลิตนั้นมีสัญญาณรบกวนที่มาจากสนามแม่เหล็ก ที่เกิดจากการเชื่อมโลหะ หม้อแปลงไฟฟ้า เป็นต้น ด้วยเหตุนี้จึงไม่

สามารถที่จะใช้สายเคเบิลที่เป็นทองแดงนำไฟฟ้าได้ และเหตุผลอีกประการคือ ระยะของการเดินสายระหว่างจุดต่อจุดนั้นใช้การเดินแบบสตาร์ คือ จากพอร์ทแต่ละพอร์ทของฮับตรงไปยังจุดที่มีเทอร์มินัลที่ใช้งานจึงทำให้ระยะของการเดินสายไกล เกินความสามารถของสายเคเบิลปกติที่จะรองรับได้ และอาจทำให้สัญญาณเกิดการสูญเสียในสายได้ ดังแสดงในรูปที่ 3.7

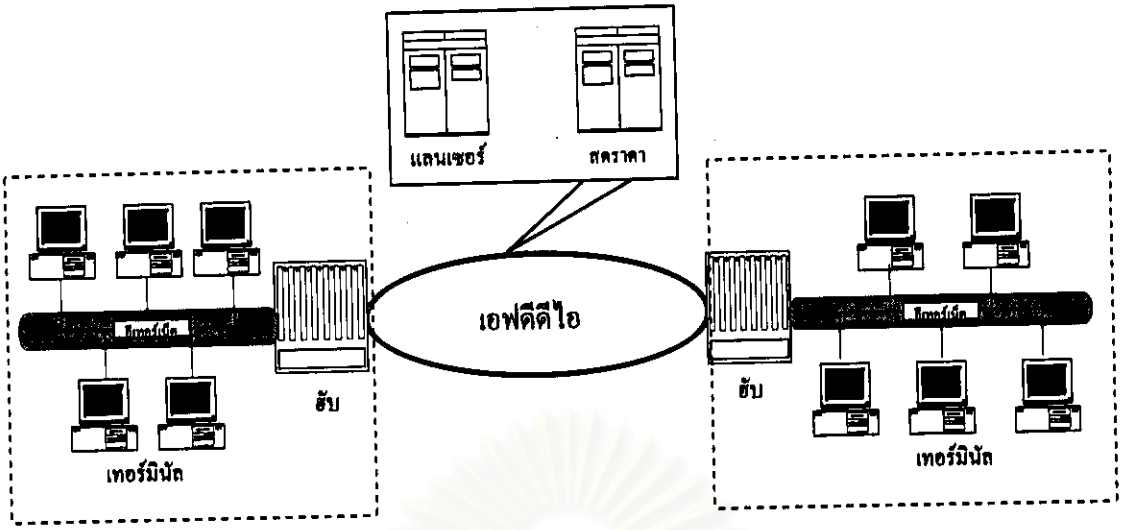


รูปที่ 3.7 โครงร่างระบบเครือข่าย

3.2 2 แผนภาพของระบบเครือข่าย

จากรูปที่ 3.8 จะเห็นว่าเราต่อโฮสต์ทั้งสองตัวเข้ากันด้วยเอฟดีดีไอ เพื่อที่ว่าในอนาคตเกิดโฮสต์ใดเกิดความล้มเหลวขึ้น เทอร์มินัลที่เชื่อมต่อกับเครือข่ายนี้สามารถใช้งานได้อย่างมีประสิทธิภาพ ไม่เกิดปัญหาในเรื่อง การจราจรในเครือข่ายที่หนาแน่น (traffic jam)

เหตุผลที่ไม่ใช้เอฟดีดีไอกับทุกๆ เทอร์มินัลเพราะว่าเป็นการสิ้นเปลืองเกินไปในขณะนี้ แต่อย่างไรก็ตามโทโพโลจีที่ประยุกต์ในการเดินสายได้รองรับไว้แล้วคือ เมื่อมีเทคโนโลยีที่สามารถเพิ่มประสิทธิภาพของเครือข่ายให้สามารถใช้ความเร็วเท่ากับความเร็วในระบบเครือข่ายเอฟดีดีไอ เราสามารถประยุกต์ใช้ได้โดยไม่ต้องเปลี่ยนแปลงโครงร่างของระบบเครือข่ายนี้



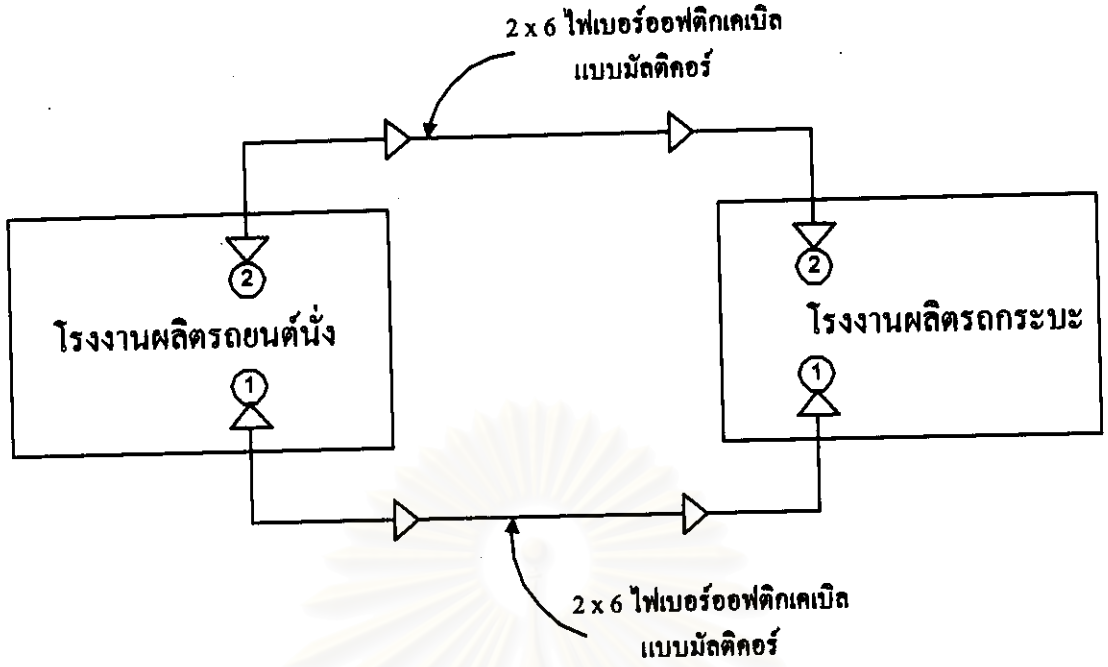
รูปที่ 3.8 แผนภาพของระบบเครือข่าย

3.2.3 สายสัญญาณและการเดินสายสัญญาณในระบบเครือข่าย

สายสัญญาณที่ใช้ในระบบเครือข่ายนี้จะใช้ สายออบติกไฟเบอร์เป็นหลัก สายออบติกไฟเบอร์ที่ใช้นั้นเป็นแบบ มัลติคอร์ คือจะมีสายออบติกไฟเบอร์หลายๆ เส้นในหนึ่งเส้นใหญ่ ประโยชน์ของจุดนี้คือเราสามารถมีสายออบติกไฟเบอร์สำรองได้ทุกๆ จุด จึงเป็นการยืนยันได้ว่าระบบเครือข่ายนี้มีความพร้อมใช้งานสูงเหมาะสมที่จะใช้กับระบบคอมพิวเตอร์แบบพร้อมใช้งานสูง

การเดินสายสัญญาณใช้การเดินสายในลักษณะที่สายหนึ่งและสอง เดินแยกออกจากกันคนละแนว ทั้งนี้เพื่อลดความเสี่ยงอันเนื่องมาจากอุบัติเหตุหรือเหตุอื่น ทำให้สายขาดเป็นผลให้ระบบเครือข่ายเกิดปัญหาไม่สามารถใช้งานได้ ดังแสดงในรูปที่ 3.9

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.9 แผนผังการเดินสายไฟเบอร์ออปติกที่ใช้ในระบบเครือข่าย

3.3 รีดันแดนซ์อะเรย์ออฟอินเอ็กซ์เพนซีฟดิสก์ (เรด) ระดับ 5

เรดระดับ 5 คล้ายกับเรดระดับ 4 คือในเรดระดับ 4 ใช้การเก็บพาร์ริตีของข้อมูล (พาร์ริตี (parity) คือ ชุดของข้อมูลสำรองที่ถูกสร้างขึ้นเพื่อที่จะใช้ในการคืนสภาพ (recovery data) เมื่อมีดิสก์ใดๆ ในโลจิคัลโวลุ่มเดียวกันเกิดความเสียหาย) บนดิสก์ตัวใดตัวหนึ่งในโลจิคัลโวลุ่มเดียวกัน แต่เรดระดับ 5 จะเก็บพาร์ริตีของข้อมูลกระจายลงบนดิสก์ทุกๆ ตัวในโลจิคัลโวลุ่มเดียวกัน เพื่อที่ช่วยลดการอ่านและเขียนบนดิสก์ที่เก็บพาร์ริตีให้น้อยลงไม่ให้เกิดสภาวะขอขวด

การสร้างพาร์ริตีของข้อมูลสามารถทำได้โดย การใช้เอ็กซ์คูลซีฟออร์ฟังก์ชัน (Exclusive OR function) เพื่อคำนวณหาพาร์ริตีของข้อมูล พาร์ริตีที่คำนวณหานั้นเป็นลักษณะบิทต่อบิท (bit-by-bit) ของข้อมูลในโลจิคัลโวลุ่มนั้น ตัวอย่างการคำนวณหาพาร์ริตีของข้อมูลสามารถทำได้ดังนี้คือ สมมุติว่าเราต้องการคำนวณหาพาร์ริตีของ P(0-3) จะได้ว่า

$$P(0\text{-}3) = Chunk0 \oplus Chunk1 \oplus Chunk2 \oplus Chunk3$$

จะเท่ากับ

$$X = P(0\text{-}3) \oplus Chunk0 \oplus Chunk1 \oplus Chunk2$$

ดังนั้นด้วยฟังก์ชันนี้ถ้ามีดิสก์ใดๆ ในโลจิคัลโวลุ่มเดียวกันเกิดเสียหายเราสามารถแก้ไขให้กลับคืนมาได้โดยการนำข้อมูลที่เหลืออยู่กับพาร์ริตีของข้อมูลที่หายไปมาผ่านฟังก์ชันจะได้ข้อมูลเดิมคืนมา

3.3.1 หลักการทำงานเบื้องต้น

ในเรดระดับ 5 การอ่านหรือเขียนข้อมูลมีการทำงานที่แตกต่างกัน ทั้งนี้เพราะว่าในการเขียนต้องมีการคำนวณหาพาร์ริตีใหม่แล้วบันทึกลงไป ส่วนการอ่านต้องตรวจสอบว่าข้อมูลที่อ่านขึ้นมานั้นถูกต้องหรือไม่ โดยการเปรียบเทียบกับพาร์ริตีของข้อมูลนั้น

ก. การอ่านข้อมูล มีขั้นตอนดังต่อไปนี้

1. โปรแกรมควบคุม (control program) ทำการเปลี่ยนเวอร์ชัวร์ดิสก์แอดเดรส (virtual disk address) ที่ต้องการอ่านข้อมูลให้เป็นแอดเดรสที่เก็บจริงในดิสก์ของระบบ (physical disk address)

2. โปรแกรมควบคุม ทำการอ่านข้อมูลจากดิสก์ตามแอดเดรสจริงที่หามาได้ ซึ่งการอ่านนี้อาจจะอ่านแบบพร้อมกันหลายๆ ดิสก์ได้ ถ้าแอดเดรสนั้นๆ ไม่อยู่บนดิสก์ที่ซ้ำซ้อนกัน

3.โปรแกรมควบคุม ตอบกลับหลังจากการอ่านเสร็จสิ้นลง

ข. การเขียนข้อมูล มีขั้นตอนดังต่อไปนี้ (ตามรูปที่ 3.10)

1. โปรแกรมควบคุมทำการอ่านข้อมูลชุดเดิมเข้าสู่บัฟเฟอร์ภายใน (internal buffers)

2. โปรแกรมควบคุมอ่านพาร์ริตีของข้อมูลชุดเดิมเข้าสู่บัฟเฟอร์ภายในซึ่งพาร์ริตีของข้อมูลที่อ่านขึ้นไป ถูกกำหนดโดยฟิสิคัลแอดเดรสของข้อมูลชุดเดิมนั้นอยู่

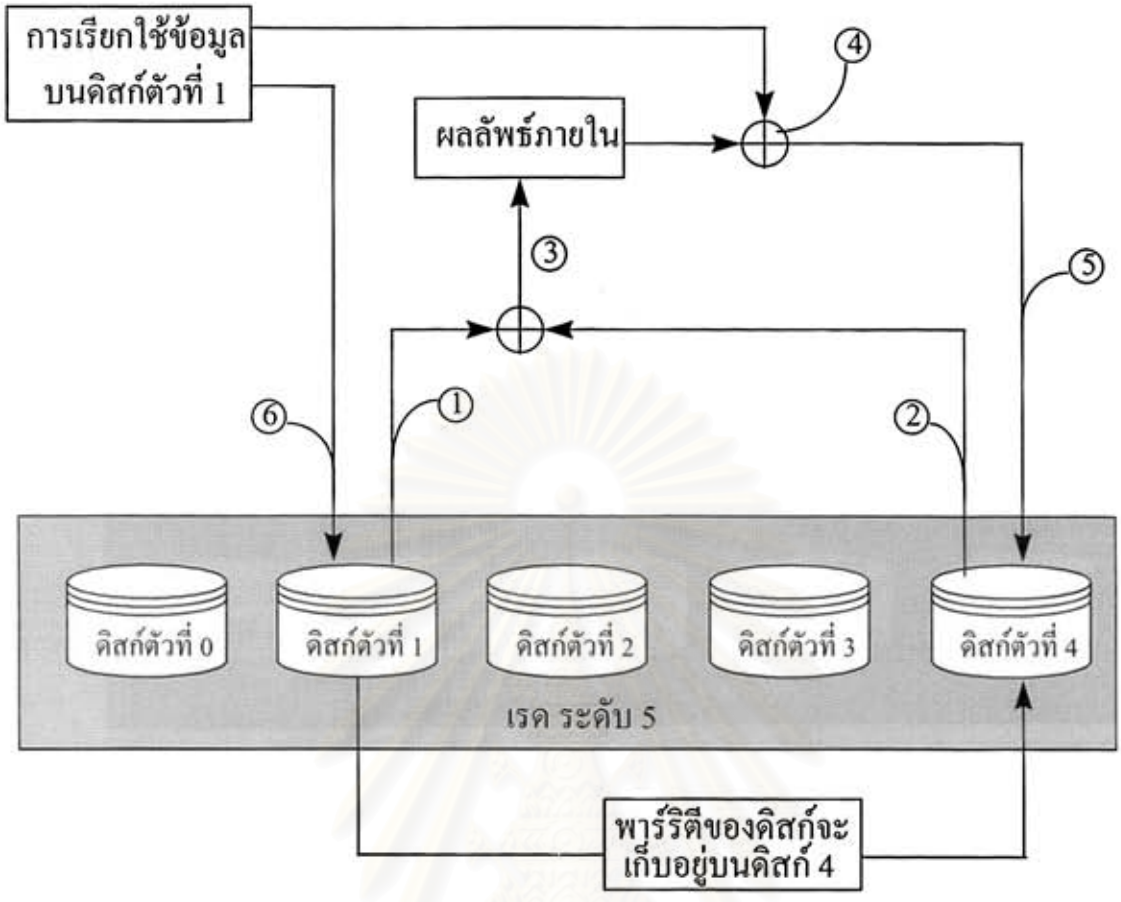
3. โปรแกรมควบคุมย้ายพาร์ริตีของชุดข้อมูลเดิมออกจากที่เก็บเดิม โดยการคำนวณพาร์ริตีของชุดพาร์ริตีนี้ด้วย (ทำดับเบิ้ลพาร์ริตี)

4. โปรแกรมควบคุมเก็บบันทึกพาร์ริตีของข้อมูลเดิมและพาร์ริตีของตัวเองไปยังฟิสิคัลแอดเดรสอื่น

5. โปรแกรมควบคุมทำการคำนวณหาพาร์ริตีของข้อมูลชุดใหม่แล้วบันทึกลงที่เดิม

6. โปรแกรมควบคุมทำการบันทึกข้อมูลชุดใหม่ลงบนดิสก์ที่กำหนด

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



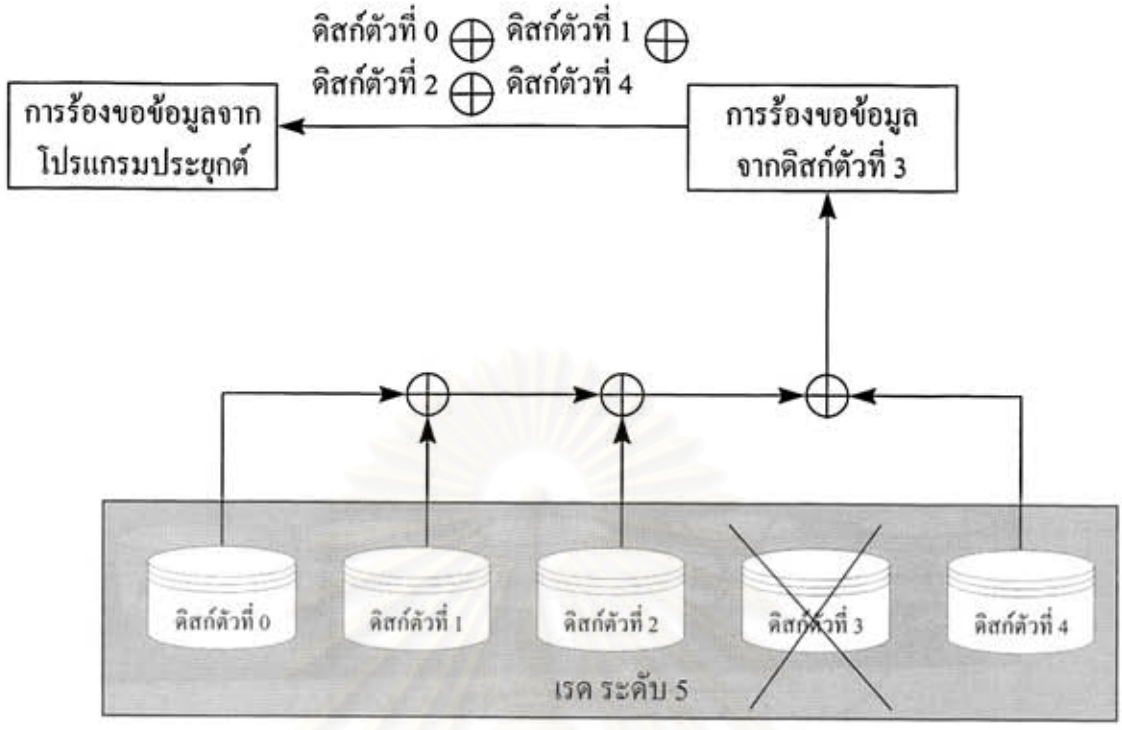
รูปที่ 3.10 ขั้นตอนการเขียนข้อมูลของเรดระดับ 5

3.3.2 การกระทำเมื่อเกิดความล้มเหลวเกิดขึ้น

ได้กล่าวไปแล้วในข้างต้นว่าเรดระดับ 5 นั้นสามารถทำงานได้ถึงแม้ว่ามีดิสก์ในระบบเกิดความเสียหาย มันจะนำข้อมูลที่เหลืออยู่มาผ่านขั้นตอนต่างๆ เพื่อคืนสภาพของข้อมูล ทำให้เราไม่ต้องหยุดทำงานในขณะที่เกิดดิสก์ในระบบเกิดเสียหาย ซึ่งขึ้นอยู่กับว่าเวลานั้นเป็นการอ่านหรือเขียนข้อมูล

ก. เมื่อต้องการอ่านข้อมูล

เมื่อมีดิสก์ใดๆ ในระบบเกิดความล้มเหลวขึ้น เรดระดับ 5 จะสามารถทำการแก้ไขโดยการนำข้อมูลในดิสก์ที่เหลืออยู่มาทำการสร้างข้อมูลในดิสก์ที่หายไปโดยการใช้ฟังก์ชันเอ็กซ์คูลซีฟออร์ ตามรูปที่ 3.11 จะเป็นการกระทำการอ่านข้อมูลในกรณีที่ดิสก์ในระบบเกิดความเสียหาย



รูปที่ 3.11 ขั้นตอนการอ่านข้อมูลของเรดระดับ 5 กรณีที่ดิสก์ในระบบเกิดความเสียหาย

ข. เมื่อต้องการเขียนข้อมูล

เมื่อต้องการเขียนข้อมูลลงบนดิสก์แต่เผอิญว่าดิสก์นั้นเกิดความเสียหาย เรดระดับ 5 จะทำขั้นตอนดังต่อไปนี้

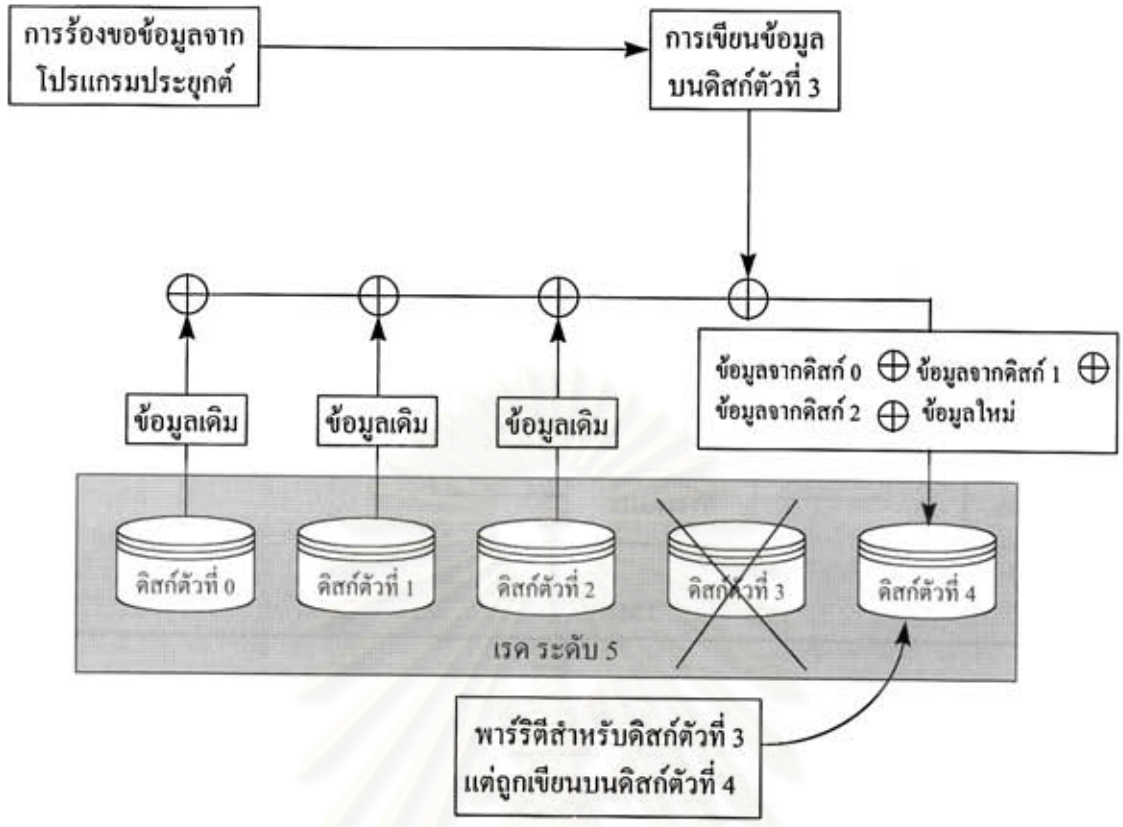
1. อ่านข้อมูลในตำแหน่งที่สมนัยกับตำแหน่งที่ต้องการเขียนข้อมูลบนดิสก์ทุกตัวที่เหลืออยู่ (ไม่รวมพาร์ริตีของข้อมูลนั้น) เข้าสู่บัฟเฟอร์ภายใน

2. ปรับปรุงพาร์ริตีของข้อมูลที่อ่านมาในข้อ 1 โดยใช้ฟังก์ชันเอ็กซ์คูลซีฟออร์ แล้วเขียนลงในดิสก์

3. ทำการบันทึกพาร์ริตีที่ทำการปรับปรุงแล้ว

ขั้นตอนดังที่กล่าวมา จะแสดงในรูปที่ 3.12

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.12 ขั้นตอนการเขียนข้อมูลของเรดระดับ 5 กรณีที่ดิสก์ในระบบเกิดความเสียหาย

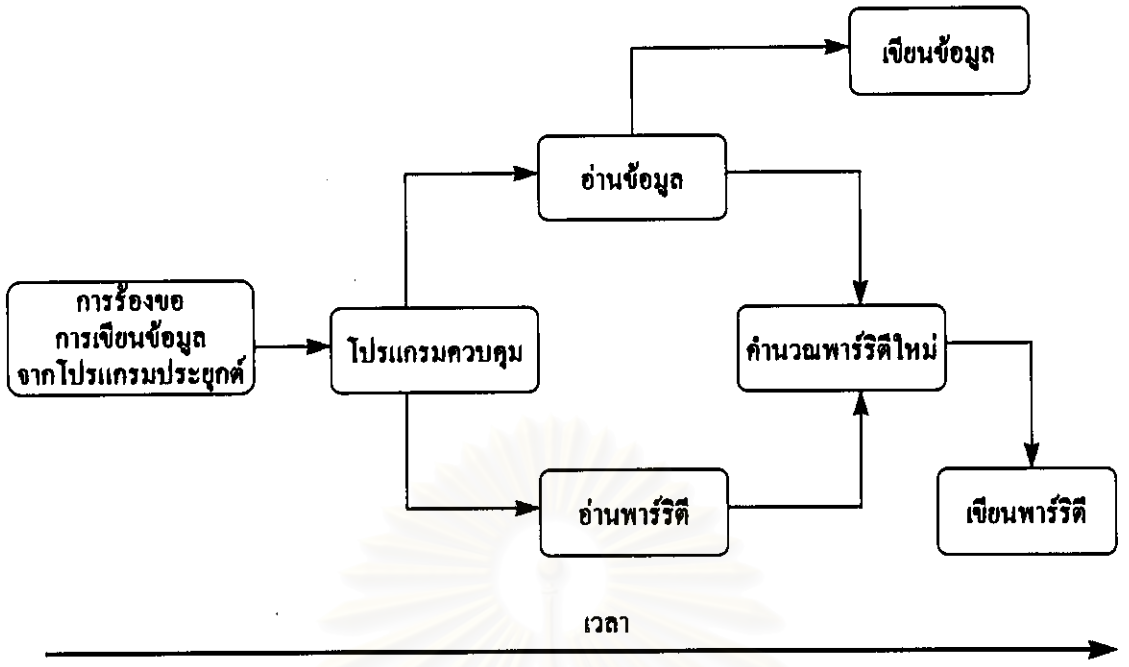
3.3.3 ประสิทธิภาพของเรดระดับ 5

ประเด็นนี้เป็นจุดที่น่าสนใจเพราะว่า มีผลกระทบต่อประสิทธิภาพของระบบโดยรวม กล่าวคือ เมื่อคอมพิวเตอร์ทำการประมวลผลต้องมีการอ่านและเขียนลงบนดิสก์มากครั้งและต่อเนื่องซึ่งในเรื่องนี้เรดระดับ 5 จะมีคุณลักษณะเด่นที่สำคัญ 2 ประการคือ

ก. เพราะว่าการเขียนข้อมูลลงบนเวอร์ชัวร์ดิสก์นั้นจะเป็นการเขียนลงบนหลายๆ ฟิสิคัลดิสก์พร้อมกัน นั่นหมายถึงเราสามารถเพิ่มประสิทธิภาพของอ่านและเขียนดิสก์ได้ อีกประการคือเราสามารถสร้างความสมดุลย์ของดิสก์ในระบบได้อีกด้วย

ข. จากเหตุผลประการแรกเราสามารถเพิ่มอัตราการส่งผ่านข้อมูล (transfer rate) เข้าสู่ระบบคอมพิวเตอร์ได้มากขึ้น ซึ่งเป็นผลให้ประสิทธิภาพของระบบดีขึ้นด้วย

จากรูปที่ 3.13 จะแสดงถึงขั้นตอนการเขียนข้อมูลลงบนดิสก์ของเรดระดับ 5 จะเห็นได้ว่าการอ่านข้อมูลและการอ่านพาร์ริตีของข้อมูลนั้นสามารถทำได้พร้อมกัน จนถึงการการเขียนข้อมูลและการเขียนพาร์ริตีของข้อมูลนั้นสามารถทำพร้อมกันเช่นกัน ทั้งนี้เป็นเพราะว่าในเรดระดับ 5 นั้นดิสก์ที่เก็บพาร์ริตีของข้อมูลกับดิสก์ที่เก็บข้อมูลนั้นจะเป็นดิสก์คนละดิสก์กัน ดังนั้นจึงสามารถอ่านและเขียนข้อมูลได้พร้อมกัน



รูปที่ 3.13 ขั้นตอนเวลาของการเขียนข้อมูลบนเรดระดับ 5

3.4 การจัดระบบโปรแกรมประยุกต์ในเครื่องคอมพิวเตอร์

การจัดโปรแกรมประยุกต์ในเครื่องคอมพิวเตอร์ที่จะใช้ในระบบพร้อมใช้งานสูงนั้นจะทำการจัดโดยใช้ลักษณะการทำงานของโปรแกรมประยุกต์เป็นหลัก คือ

ก. โปรแกรมประยุกต์ที่ทำงานแบบแบทช์

ข. โปรแกรมประยุกต์ที่ทำงานแบบโต้ตอบกับผู้ใช้

เหตุผลที่ใช้ลักษณะการทำงานของโปรแกรมประยุกต์เป็นหลักในการจัดโปรแกรมประยุกต์ในแต่ละเครื่องเพราะว่า โปรแกรมที่มีความสำคัญในการควบคุมสายการผลิตคือ ระบบโปรแกรม ALC ที่ใช้ในการควบคุมการประกอบส่วนต่างๆ ของรถยนต์ การที่ให้ความสำคัญกับระบบโปรแกรม ALC ก็เพราะว่า ในเวลา 1 วันของการทำงานจะถูกแบ่งออกเป็น 4 ช่วงๆ ละ 2 ชั่วโมง เหตุที่ต้องแบ่งเวลาออกเป็นช่วงๆ เพราะว่า

1. สามารถตรวจสอบปริมาณการผลิตได้อย่างชัดเจน

2. สามารถตรวจสอบวัตถุดิบที่ใช้ในการผลิตในช่วงเวลาต่างๆ เพื่อที่จะจัดส่งวัตถุเข้าสู่สายการผลิต และควบคุมการใช้วัตถุในแต่ช่วงเวลาเพื่อป้องกันการสูญหาย

3. สามารถที่จะทำให้ทราบปริมาณของรถยนต์ที่ประกอบเสร็จ เพื่อที่จะประมาณการส่งมอบให้แก่ลูกค้าได้อย่างถูกต้อง (เป็นการเพิ่มมูลค่าของสินค้าประการหนึ่ง คือ เมื่อผลิตเสร็จเราสามารถขายสินค้าได้ทันทีโดยไม่ต้องสต๊อกสินค้านั้นไว้)

4. สามารถทราบถึงปริมาณรถยนต์ที่ต้องเก็บในคลังเก็บ เพื่อคำนวณหาจำนวนสินค้าคงคลัง ควบคุมการนำรถยนต์เข้าเก็บในคลังเพื่อป้องกันการสูญหาย

5. ควบคุมเวลาในการประกอบชิ้นส่วน ในแต่ละจุดให้เป็นไปอย่างมีประสิทธิภาพ

6. สามารถประมาณกำลังผลิตที่ผลิตได้ กับความต้องการของตลาด เพื่อการวางแผนการขายได้อย่างมีประสิทธิภาพ

ด้วยการแบ่งเวลาออกเป็นช่วงนี้สายการผลิตจะดำเนินไปตามขั้นตอนของการผลิต โดยมีการควบคุมในเรื่องของ ลำดับขั้นตอนการทำงาน (work flow) และเวลาในการประกอบชิ้นส่วนแต่ละจุดคือ ตัวถังรถยนต์จะไหลไปตามจุดภายในสายการผลิตเพื่อจะประกอบชิ้นส่วนต่างๆ เช่น ระบบส่งกำลัง ระบบช่วงล่าง และการตกแต่งภายในห้องโดยสาร เป็นต้น การทำงานที่กล่าวมาทั้งหมดจะถูกสั่งการจากระบบโปรแกรม ALC ในการที่จะประกอบชิ้นส่วนแต่ละชิ้น การพ่นสีและอื่นๆ ของรถยนต์แต่ละคัน แต่ละรุ่น ดังนั้นเมื่อใดก็ตามระบบโปรแกรม ALC ไม่สามารถทำงานได้ซึ่งเท่ากับว่า สายการผลิตจะหยุดลงโดยปริยาย ก่อให้เกิดความสูญเสียต่อสายการผลิตเป็นอย่างมาก

อีกประการหนึ่งคือ ลักษณะการทำงานของระบบโปรแกรม ALC จะเป็นแบบโต้ตอบกับผู้ใช้ ดังนั้นขนาดของทรานแซกชันที่เกิดขึ้นในระบบโปรแกรม ALC จะมีขนาดเล็ก ใช้เวลาในการทำโรลล์แบคเวิร์ดหรือโรลล์ฟอร์เวิร์ดน้อยกว่า ระบบโปรแกรมอื่นๆ ที่มีการทำงานแบบแบทช์ ซึ่งมีทรานแซกชันขนาดใหญ่

การจัดระบบโปรแกรมประยุกต์ในเครื่องคอมพิวเตอร์แต่ละเครื่อง จะแยกโปรแกรมประยุกต์ที่เป็นแบบแบทช์ และ แบบโต้ตอบกับผู้ใช้ให้อยู่คนละอินสแตนท์ (instance) การที่ให้คนละอินสแตนท์นั้นหมายถึง การที่ให้แต่ละกลุ่มของระบบโปรแกรมประยุกต์ใช้ ส่วนของแชร์เมมโมรีเซกเมนต์ (shared memory segment) แยกของออกจากกัน ดังนั้นความอิสระของกลุ่มระบบโปรแกรมจึงมีมาก และอีกทั้งยังง่ายต่อการควบคุม สะดวกในการทำโรลล์แบคเวิร์ดและโรลล์ฟอร์เวิร์ด ของระบบโปรแกรมประยุกต์ในแต่ละกลุ่มเมื่อเกิดความล้มเหลวขึ้น การจัดระบบโปรแกรมประยุกต์แสดงในตารางที่ 3 .1

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ชื่อเครื่อง	อินสแตนท์	รหัสโปรแกรม
แลนเซอร์	หนึ่ง	ALC
	สอง	BOM
		PA
		PB
		PP
		PC
		PS
		PR
		INV
สตราดา	หนึ่ง	ALC
	สอง	BOM
		PA
		PB
		PP
		PC
		PS
		PR
		INV

ตารางที่ 3.1 การจัดระบบโปรแกรมประยุกต์ในเครื่องคอมพิวเตอร์แต่ละเครื่อง

3.5 สภาวะแวดล้อมของระบบ

เนื่องจากระบบนี้จะใช้งานกับโรงงานอุตสาหกรรมดังนั้น ระบบไฟฟ้าจึงเป็นเรื่องสำคัญ เพราะว่า อาจมีเหตุการณ์บางอย่างที่ทำให้กระแสไฟฟ้าที่จ่ายให้กับระบบคอมพิวเตอร์เกิดขัดข้อง ไม่สามารถทำงานได้ แต่สายการผลิตยังดำเนินอยู่ ซึ่งก่อให้เกิดผลการเสียกับสายการผลิตอีกเช่นกัน ถึงแม้ว่าเรามีระบบสำรองไฟฟ้า (Un-interrupt Power Supply) อยู่แล้ว แต่ระบบสำรองไฟฟ้าที่มีอยู่นั้นไม่สามารถใช้ได้เป็นเวลานาน ถ้าต้องใช้เวลาในการแก้ไขปัญหาที่เกิดขึ้น

ดังนั้นจึงควรมีเครื่องกำเนิดไฟฟ้า (Power generator) ที่สามารถรองรับการใช้ในระบบได้เป็นเวลานานกว่า 2 ชั่วโมง เพื่อที่ควบคุมการทำงานของสายการผลิตในหนึ่งช่วงการทำงานได้

เมื่อเรานำส่วนประกอบทั้ง 5 มาประกอบกันเราจะได้ระบบพร้อมใช้งานสูง ซึ่งจะทำให้ระบบคอมพิวเตอร์ที่ใช้ในการควบคุมสายการผลิตมีความเชื่อถือได้มากขึ้น และอีกทั้งยังสามารถประยุกต์ใช้สิ่งต่างๆ ที่มีอยู่ในระบบปฏิบัติการ ระบบดิสก์ และระบบเครือข่าย ได้อย่างมีประสิทธิภาพและคุ้มค่า ซึ่งสามารถช่วยลดค่าใช้จ่ายในการผลิตได้อีกด้วย อย่างไรก็ดีระบบคอมพิวเตอร์แบบพร้อมใช้งานสูงนี้จะต้องคำนึงผลกระทบในด้านต่างๆ ความเป็นไปได้ในการประยุกต์ใช้งาน และการลงทุนในการสร้างระบบดังกล่าวขึ้นมา ซึ่งจะได้กล่าวในบทต่อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย