บทที่ 3
โครงสร้างและสถาปัตยกรรมของระบบปฏิบัติการ

โครงสร้างของระบบปฏิบัติการและสถาปัตยกรรมของระบบปฏิบัติการเป็นการพัฒนาที่ควบคู่กันมาโดยสืบเนื่องมาจากโครงสร้างคอมพิวเตอร์ ด้วยเหตุที่ว่าโปรแกรมเมอร์จะต้องพัฒนาระบบปฏิบัติการให้สามารถเข้ากับการทำงานของระบบคอมพิวเตอร์และรองรับการใช้งานของผู้ใช้ได้ ซึ่งในบทเรียนนี้จะกล่าวถึง 1) โครงสร้างระบบคอมพิวเตอร์ 2) โครงสร้างของระบบปฏิบัติการ 3) สถาปัตยกรรมของระบบปฏิบัติการและ 4) โครงสร้างพื้นฐานของไมโครโปรเซสเซอร์ 5) สถาปัตยกรรมแบบ CISC และ RISC

3.1 โครงสร้างระบบคอมพิวเตอร์

โครงสร้างระบบคอมพิวเตอร์ประกอบไปด้วยระดับชั้นการทำงานต่างๆ ซึ่งผู้สอนได้ศึกษาโครงสร้างระบบคอมพิวเตอร์จากนักวิชาการดังนี้ 1) พิเชษฐ์ ศิริรัตนไพศาลกุล (2547: 25-35) 2) ธีรวัฒน์ ประกอบผลและจันทนา ผ่องเพ็ญศรี (2551: 4-5) 3) ฝ่ายตำราวิชาการคอมพิวเตอร์ (2557: 56-58) แล้วนำมากล่าวว่าโดยสรุปว่าโครงสร้างระบบคอมพิวเตอร์แบ่งได้เป็น 6 ชั้น (Level) โดยชั้นบนเป็นระดับชั้นใกล้ชิดกับผู้ใช้มากที่สุด ชั้นล่างเป็นระดับที่ใกล้ชิดกับเครื่องและวงจรอิเล็กทรอนิกส์มากที่สุด ซึ่งอธิบายได้ดังนี้

3.1.1 ชั้นที่ 0 จะเป็นวงจรพื้นฐานต่างๆ ของเครื่องที่เรียกว่า Digital Logic Level ซึ่งประกอบด้วยวงจรลอจิกเกตต่างๆ รีจิสเตอร์ต่างๆ

3.1.2 ชั้นที่ 1 เรียกว่า ไมโครสถาปัตยกรรม(Micro Architecture Level) ประกอบไปด้วยรีจิสเตอร์ต่างๆ รวมกันเป็นหน่วยความจำภายใน วงจรที่ประมวลผลทางคณิตศาสตร์และลอจิกหรือ ALU โดยมีการส่งข้อมูลมาประมวลผลร่วมกันได้ เช่น การบวกข้อมูลที่เก็บอยู่ในรีจีสเตอร์ ชั้นนี้การควบคุมการประมวลผลและการทำงานต่างๆ จะต้องใช้โปรแกรมในระดับภาษาเครื่อง ซึ่งบางครั้งจะเรียกว่า Micro program

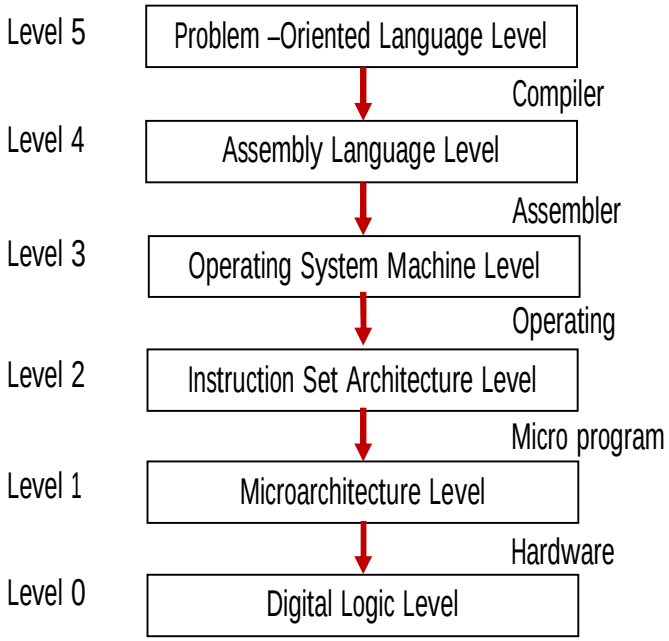
3.1.3 ชั้นที่ 2 เรียกว่า ชุดคำสั่งสถาปัตยกรรม (Instruction Set Architecture Level) หรือ ISA Level สามารถเข้าใจชุดคำสั่งต่างๆ ของไมโครโปรเซสเซอร์ได้ การสั่งงานให้ระบบทำงานต่างๆ จะสั่งงานผ่านคำสั่งของเครื่อง

3.1.4 ชั้นที่ 3 เรียกว่า ระบบปฏิบัติการ (Operating System Machine Level) เป็นชั้นของระบบปฏิบัติการ ซึ่งจะเชื่อมโยงระหว่างโปรแกรมต่างๆ กับอุปกรณ์ทางฮาร์ดแวร์

3.1.5 ชั้นที่ 4 เรียกว่า ภาษาแอสแซมบลี (Assembly Language Level) เป็นชั้นของภาษาเครื่อง การควบคุมการทำงานในชั้นนี้จะต้องใช้ภาษาแอสแซมบลี โดยตัวที่ทำหน้าที่แปลงโปรแกรมไปควบคุมระดับชั้น 1,2 และ 3 จะเรียกว่า แอสแซมเบลอร์(Assambler)

3.1.6 ชั้นที่ 5 เรียกว่า เชิงภาษา (Problem Oriented Language Level) เป็นชั้นที่โปรแกรมเมอร์สามารถเขียนโปรแกรมควบคุมด้วยภาษาระดับสูงได้ ตัวที่จะทำการแปลงภาษาโปรแกรมไปให้กับระดับชั้น 4 และ 3 เรียกว่า คอมไพเลอร์ (Compilers)

จากคำอธิบายโครงสร้างคอมพิวเตอร์สามารถนำมาแสดงได้ ดังภาพที่ 3.1



ภาพที่ 3.1 แสดงระดับชั้นของคอมพิวเตอร์ 6 ชั้น

ที่มา : ธีรวัฒน์ ประกอบผลและจันทนา ผ่องเพ็ญศรี (2551: 5)

จากแผนงานระดับชั้นของคอมพิวเตอร์จะพบว่าผู้ที่เขียนโปรแกรมซึ่งทำงานอยู่ในระดับบนสุดไม่จำเป็นต้องทราบวงจรการทำงานของเครื่องและไม่จำเป็นต้องทราบการเขียนโปรแกรมภาษาแอสแซมบลีสามารถเขียนโปรแกรมให้กับคอมพิวเตอร์ได้

3.2 โครงสร้างของระบบปฏิบัติการ

โครงสร้างของระบบปฏิบัติการมีปัจจัยสำคัญในการดำเนินงาน ซึ่งผู้สอนได้ศึกษาเกี่ยวกับโครงสร้างของระบบปฏิบัติการจากนักวิชาการ ดังนี้ 1) พิรพร หมุนสนิท (2553: 33-34) 2) พีระพนธ์ โสพัศสถิต (2552: 26-35) 3) ฝ่ายตำราวิชาการคอมพิวเตอร์ (2557: 62-65) แล้วนำมากล่าวโดยสรุปได้ว่า โครงสร้างของระบบปฏิบัติการมีปัจจัยสำคัญในการดำเนินงาน ได้แก่ 1) สภาพแวดล้อมของระบบปฏิบัติการ (Operating System Environment) 2) ระบบดำเนินการแบบทันที (System Call) 3) บริการของระบบปฏิบัติการ

3.2.1 สภาพแวดล้อมของระบบปฏิบัติการ

การใช้งานระบบปฏิบัติการจำเป็นต้องมีสภาพแวดล้อม (Environment) ที่เหมาะสม ซึ่งระบบปฏิบัติการที่แตกต่างกันย่อมมีความต้องการสภาพแวดล้อมที่ไม่เหมือนกัน ขึ้นอยู่กับโครงสร้างของระบบปฏิบัติการว่าต้องการสภาพแวดล้อมแบบใด จึงจะสามารถทำงานได้อย่างเต็มประสิทธิภาพ ฮาร์ดแวร์และซอฟต์แวร์ที่อยู่ภายในระบบถือว่าเป็นส่วนหนึ่งของสภาพแวดล้อมที่ช่วยผลักดันให้ระบบปฏิบัติการทำงานได้อย่างดีเยี่ยม ในการออกแบบระบบปฏิบัติการจึงต้องพิจารณาจากสภาพแวดล้อมของระบบก่อนเป็นอันดับแรก เพื่อให้ทราบถึงข้อจำกัดและสิ่งต่างๆ ที่อาจส่งผลกระทบต่อระบบได้ ระบบปฏิบัติการที่มีความต้องการทรัพยากรค่อนข้างสูง จำเป็นต้องออกแบบให้มีองค์ประกอบสภาพแวดล้อมต่างๆ ที่มีประสิทธิภาพสูง เช่นกัน โดยอาจมีวัตถุประสงค์ในการออกแบบแตกต่างกันไปส่งผลให้ทรัพยากรต่างๆ มีความแปรผันตามสภาพแวดล้อมนั้นด้วย เช่น ต้องการหน่วยความจำที่มีขนาดใหญ่มาก ต้องการใช้งานอุปกรณ์เฉพาะทางและมีโปรเซสที่ต้องประมวลผลจำนวนมาก ระบบปฏิบัติการลักษณะนี้จำเป็นต้องอาศัยการทำงานของโปรเซสเซอร์ (Processor) จำนวนมากหรือที่ เรียกว่า มัลติโปรเซสเซอร์ (Multiprocessor) ซึ่งต้องการจัดการเฉพาะทาง นอกจากนี้การเปลี่ยนแปลงของเทคโนโลยีต่างๆ ยังเป็นตัวกำหนดแนวทางในการพัฒนาระบบปฏิบัติการได้อีกด้วย เช่น การออกแบบระบบปฏิบัติการในอุปกรณ์เคลื่อนที่ต่างๆ ที่มีระบบการทำงานที่เรียกว่า“ระบบสมองกลฝังตัว” (Embedded System) ซึ่ง Embedded System เป็นอีกระบบหนึ่งที่มีความหลากหลายของสภาพแวดล้อมเมื่อนำไปใช้งานจริง ซึ่งอาจถูกนำไปใช้กับอุปกรณ์ขนาดเล็กที่มีทรัพยากรจำกัด ในขณะเดียวกันอาจถูกนำไปใช้ในอุปกรณ์เฉพาะทางที่ต้องการความแม่นยำและปรับเปลี่ยนอยู่ตลอดเวลา ซึ่งทำให้สภาพแวดล้อมของระบบปฏิบัติการเปลี่ยนแปลงไป นอกจากข้อจำกัดด้านเทคโนโลยีแล้วความต้องการหรือ

วัตถุประสงค์ของระบบยังเป็นตัวกำหนดแนวทางในการเลือกใช้ระบบปฏิบัติการ เนื่องจากความต้องการหรือวัตถุประสงค์ที่แตกต่างกัน ทำให้ระบบปฏิบัติการเดียวกันทำงานได้ประสิทธิภาพที่แตกต่างกัน เช่น ระบบวิเคราะห์ทางธุรกิจออนไลน์ที่จำเป็นต้องมีความแม่นยำและมีความรวดเร็วในการประมวลผล รวมถึงต้องมีความปลอดภัยสูงด้วย ดังนั้นสภาพแวดล้อมต้องเอื้ออำนวยต่อระบบปฏิบัติการเพื่อให้สามารถดำเนินการได้ตามที่ต้องการ

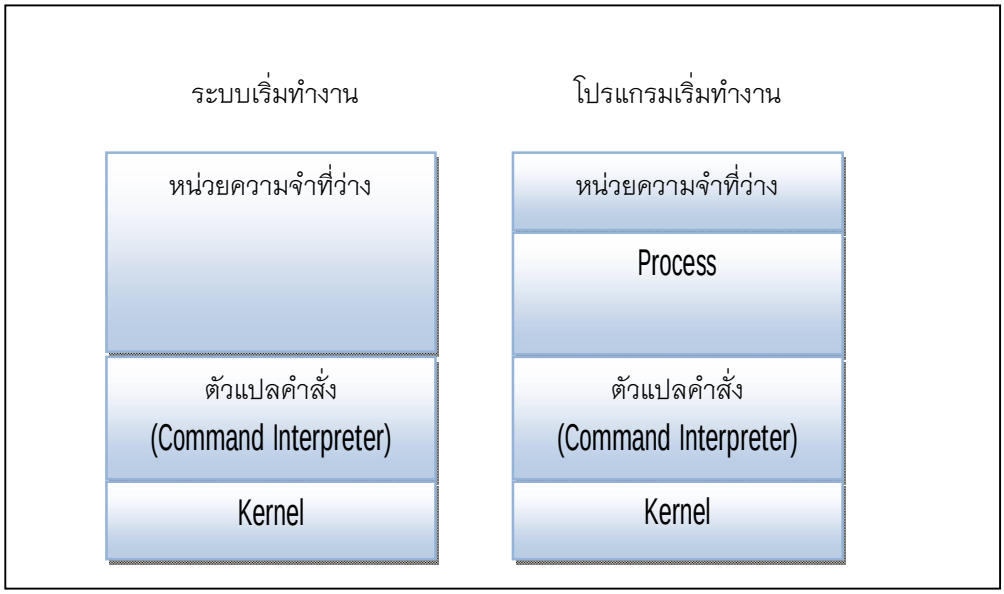
3.2.2 ระบบดำเนินการแบบทันที

ในระหว่างที่ระบบปฏิบัติการกำลังดำเนินการอยู่ อาจได้รับสัญญาณที่แจ้งมาให้ทราบว่ามีงานใหม่ต้องการให้ระบบดำเนินการแบบทันทีหรือมีงานใหม่ที่มีลำดับความสำคัญมากกว่าต้องการดำเนินการ ทำให้ระบบปฏิบัติการจำเป็นต้องพิจารณางานที่เข้ามาใหม่ก่อน สัญญาณที่ระบบปฏิบัติการได้รับเป็นสัญญาณสำหรับเรียกใช้งานระบบที่เรียกว่า “System Call” ซึ่งเป็นสัญญาณสำหรับแจ้งให้ซีพียู (CPU) เปลี่ยนจากงานเดิมมาทำงานชิ้นใหม่ก่อน แล้วจึงกลับไปทำงานเดิมต่อเมื่อเสร็จสิ้นจากงานใหม่ที่ร้องขอมา การทำงานในลักษณะนี้ช่วยให้ระบบคอมพิวเตอร์มีความยืดหยุ่นในการใช้งานมากขึ้น กลุ่มคำสั่งของ System Call แบ่งได้เป็น 5 กลุ่ม ได้แก่ 1) คำสั่งควบคุม โปรเซส (Process Control) 2) คำสั่งจัดการไฟล์ข้อมูล (File Management) 3) คำสั่งจัดการอุปกรณ์ (Device Management) 4) คำสั่งบำรุงรักษาข้อมูล (Information Maintenance) และ 5) คำสั่งการติดต่อสื่อสาร (Communication)

3.2.2.1 คำสั่งควบคุมโปรเซส (Process Control)

เป็นกลุ่มคำสั่งสำหรับควบคุมการทำงานของโปรเซส (Process) ทั้งหมดในระบบ แม้ว่าจะมีการเปลี่ยนแปลงการทำงานของ โปรเซส อยู่ตลอดเวลา เช่น สั่งให้โปรแกรมหยุดทำงาน สร้างหรือทำลาย โปรเซส และจัดการระยะเวลาในการประมวลผลโปรเซส ซึ่งในแต่ละระบบปฏิบัติการ เมื่อได้รับสัญญาณเรียกใช้งานระบบที่เกี่ยวข้องกับการควบคุมโปรเซสอาจแสดงผลออกมาได้แตกต่างกันตามรูปแบบการทำงาน สำหรับระบบปฏิบัติการที่ใช้ส่วนต่อประสานกับผู้ใช้แบบกราฟิก (GUI) สัญญาณเรียกใช้ระบบอาจแสดงในรูปแบบของหน้าต่างแบบ Pop Up ซึ่งจะปรากฏขึ้นในระหว่างที่กำลังดำเนินการโปรแกรมใดๆ อยู่เพื่อแจ้งให้ทราบถึงข้อผิดพลาดหรือคำสั่งใหม่ โดยผู้ใช้จะเลือกตัดสินใจและรับรู้ได้ว่าระบบจะดำเนินการต่อไปอย่างไร แต่สำหรับระบบแบช (Batch System) นั้น จะต้องทำงานตามลำดับคำสั่งโดยรอให้คำสั่งก่อนหน้าเสร็จก่อนจึงจะดำเนินการคำสั่งต่อไปได้ ระบบปฏิบัติการแต่ละประเภทจะมีการควบคุมการทำงานของโปรเซสแตกต่างกันไปตาม

ความสามารถในการประมวลผล เช่น MS-DOS เป็นระบบปฏิบัติการสำหรับใช้ได้งานเดียว (Single Tasking) ซึ่งสามารถประมวลผลได้ครั้งละหนึ่งโปรเซสเท่านั้น เมื่อระบบปฏิบัติการเริ่มทำงาน หน่วยความจำจะว่างและพร้อมที่จะรับคำสั่งจากโปรแกรม หากมีการเรียกใช้โปรแกรมระบบจะไม่สร้างโปรเซสใหม่ขึ้น แต่จะนำโปรแกรมมาจัดเก็บไว้ในหน่วยความจำแทน ซึ่งเป็นการทำงานอย่างง่ายเนื่องจากระบบรองรับแค่เพียงงานเดียวเท่านั้น ดังแสดงภาพที่ 3.2



ภาพที่ 3.2 แสดงการทำงานของ MS-DOS
ที่มา : พิรพร หมุนสนิทและคณะ (2553: 37)

ยูนิกซ์(Unix) เป็นระบบปฏิบัติการสำหรับใช้ได้หลายงาน (Multi Tasking) ซึ่งสามารถดำเนินการได้พร้อมกันหลายโปรเซส เมื่อระบบปฏิบัติการได้รับการร้องขอจากผู้ใช้จะรับคำสั่งและนำโปรแกรมที่ต้องการไปจัดเก็บไว้ในหน่วยความจำเพื่อใช้งาน จากนั้นจะดำเนินการกับโปรแกรมตามที่ผู้ใช้ร้องขอ โดยเมื่อผู้ใช้สั่งการเสร็จสิ้นระบบจะต้องกลับมายังโปรแกรมอื่นที่ผู้ใช้ต้องการใช้งานต่อไปและไม่จำเป็นต้องรอให้โปรเซสดังกล่าวเสร็จสิ้น ซึ่งการแสดงผลบนหน้าจอจะเปลี่ยนไปตามโปรแกรมที่ผู้ใช้ต้องการ ส่วนโปรแกรมที่สั่งการไปแล้วจะติดต่อสื่อสารผ่านทางไฟล์ข้อมูลแทนหน้าจอทำให้สามารถใช้งานได้หลายโปรแกรมพร้อมกัน ดังแสดงภาพที่ 3.3

ภาพที่ 3.3 แสดงการทำงานของ UNIX
ที่มา : พิรพร หมุนสนิทและคณะ (2553: 37)

3.2.2.2 คำสั่งจัดการไฟล์ข้อมูล (File Management)

เป็นกลุ่มคำสั่งที่ใช้สำหรับเข้าถึงและจัดการกับไฟล์หรือแฟ้มข้อมูล โดยใช้ในการเรียกใช้งานระบบเพื่อดำเนินการกับไฟล์ข้อมูลในลักษณะต่างๆ ตามความต้องการของผู้ใช้ การจัดการกับไฟล์ข้อมูลช่วยให้สามารถวางระบบและจัดระเบียบกับไฟล์ข้อมูลที่มีอยู่จำนวนมากได้ เช่น

1) เปิด (Open) และปิด (Close) ไฟล์ข้อมูล เป็นคำสั่งพื้นฐานในการเริ่มต้นและสิ้นสุดการใช้งานไฟล์ข้อมูลที่ต้องการซึ่งเริ่มจากการเปิด (Open) ไฟล์ข้อมูลเมื่อใช้งานเสร็จจึงปิด (Close) ไฟล์ข้อมูล

2) สร้าง (Create) และลบ (Delete) ไฟล์ข้อมูล เป็นคำสั่งในการจัดการไฟล์ข้อมูล โดยการสร้างขึ้นใหม่ หรือลบไฟล์ข้อมูลที่ไม่ต้องการออกจากระบบ

3) ย้าย (Move) และคัดลอก (Copy) คำสั่งย้าย (Move) เป็นคำสั่งสำหรับใช้เคลื่อนย้ายไฟล์ข้อมูลมาที่อยู่เดิมนำไปเก็บไว้ที่ใหม่ ส่วนคำสั่งคัดลอก (Copy) เป็นการทำสำเนาไฟล์ข้อมูลขึ้นใหม่อีกหนึ่งไฟล์ ซึ่งมีรายละเอียดภายในเหมือนกันแต่ใช้ชื่อไฟล์ข้อมูลที่ต่างกันหรืออาจระบุไว้ว่าเป็นไฟล์ข้อมูลสำเนา

3.2.2.3 คำสั่งจัดการอุปกรณ์ (Device Management)

เป็นกลุ่มคำสั่งที่ใช้สำหรับจัดการกับทรัพยากรในระบบที่เป็นอุปกรณ์ทั้งหมด อาจเป็นอุปกรณ์ภายในเครื่องหรือภายนอกที่เชื่อมต่อกับระบบคอมพิวเตอร์ อุปกรณ์เหล่านี้ถูกเรียกใช้งานตามคำร้องขอของผู้ใช้เป็นหลัก ในกรณีที่มีผู้ใช้มากกว่าหนึ่งคนต้องการใช้งานอุปกรณ์เดียวกัน ระบบจะต้องพิจารณาผู้ใช้ที่ร้องขอมาเป็นอันดับแรกก่อนแล้ว

จึงจัดสรรให้กับผู้ใช้ในลำดับถัดไปหลังจากผู้ใช้ก่อนหน้าใช้งานเสร็จสิ้นแล้ว คำสั่งในการจัดการกับอุปกรณ์ เช่น

1) ร้องขอ (Request) และปล่อยคืน(Release) อุปกรณ์เป็นคำสั่งในการร้องขอใช้งานอุปกรณ์ที่ต้องการ โดยผู้ที่ร้องขอเป็นอันดับแรกจะได้ใช้งานก่อน เมื่อใช้งานเสร็จจะปล่อยคืนอุปกรณ์เพื่อให้ผู้อื่นใช้งานต่อไป

2) อ่านและตั้งค่าข้อมูลเฉพาะของอุปกรณ์ (Get and Set Device Attributes) เป็นการอ่านข้อมูลหรือคุณสมบัติของอุปกรณ์เพื่อให้ทราบรายละเอียดต่างๆ

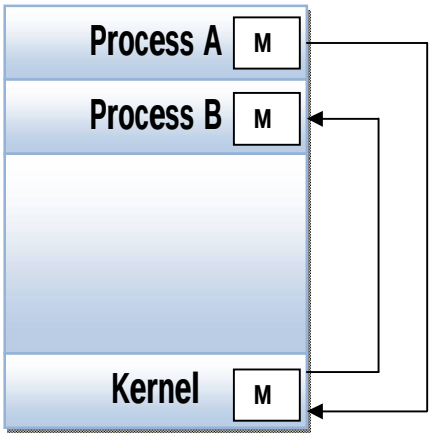
3.2.2.4 คำสั่งบำรุงรักษาข้อมูล(Information Maintenance) เป็นกลุ่มคำสั่งที่มีความเกี่ยวข้องกับข้อมูลของระบบซึ่งเป็นรายละเอียดในการใช้งานข้อมูลต่างๆ โดยข้อมูลเหล่านี้จะแสดงถึงสถานภาพของระบบที่มีความสำคัญต่อผู้ใช้หากต้องการจะรับรู้เพื่อให้เข้าใจถึงความเป็นไปของระบบ เช่น จำนวนผู้ใช้ระบบเวอร์ชันของระบบปฏิบัติการและพื้นที่ว่างของหน่วยความจำ ข้อมูลเหล่านี้บางส่วนอาจมีความจำเป็นต้องการใช้งานทั่วๆ ไป และเป็นข้อมูลสำคัญที่ใช้ในการบำรุงรักษาระบบหรือข้อมูลที่เกี่ยวข้องกัน แต่บางส่วนอาจมีความจำเป็นในทางเทคนิคที่ผู้ใช้งานกลุ่มต้องการระบบปฏิบัติการจะจัดเก็บข้อมูลนี้ไว้ซึ่งจะเป็นประโยชน์ต่อระบบและตัวผู้ใช้เอง

3.2.2.5 คำสั่งการติดต่อสื่อสาร (Communication) เป็นกลุ่มคำสั่งเกี่ยวข้องกับการติดต่อสื่อสารระหว่าง โปรเซส มี 2 วิธี ดังนี้

1) การรับ-ส่งข้อความ (Message Passing)

เป็นการติดต่อสื่อสารระหว่าง โปรเซส ด้วยการรับ-ส่งข้อความ สามารถติดต่อสื่อสารได้ 2 รูปแบบ คือ 1) การสื่อสารระหว่าง โปรเซส ในระบบคอมพิวเตอร์เดียวกัน 2) การสื่อสารระหว่างระบบคอมพิวเตอร์ผ่านทางระบบเครือข่ายการติดต่อสื่อสารหรือแลกเปลี่ยนข้อมูลระหว่าง โปรเซส ซึ่งไม่ว่าจะเป็นรูปแบบใดก็ตามจะอาศัยกล่องข้อความ (Mailbox) เป็นตัวกลางในการรับ-ส่งข้อความ โดยขั้นตอนการส่งข้อความจะเริ่มต้นด้วยการเปิดช่องทางการเชื่อมต่อและอาศัย Host Name ของเครื่องคอมพิวเตอร์ ซึ่งทุกเครื่องในระบบเครือข่ายจะต้องมี Host Name ประจำตัวอยู่ สำหรับการระบุตำแหน่งของเครือข่ายที่ชัดเจนอาจใช้หมายเลข IP ในการค้นหาเครือข่ายที่ถูกต้อง การรับ-ส่ง ข้อความเหล่านี้เป็นการติดต่อระหว่างโปรเซสด้วยกัน ทำให้แต่ละโปรเซสจำเป็นต้องมีโปรเซส Name เพื่อใช้เป็นชื่อในการระบุตัวตนช่วยให้การติดต่อสื่อสารมีความถูกต้อง กระบวนการรับ-ส่งข้อความประกอบด้วยต้นทางที่เรียกว่า “ไคลเอนต์ (Client)” ทำหน้าที่เป็นผู้ส่งข้อความและ

ปลายทางที่เรียกว่า “เซิร์ฟเวอร์(Server)” ทำหน้าที่เป็นผู้รับข้อความ โดยจะเริ่มต้นดำเนินการจากการเปิดช่องทางการเชื่อมต่อ ซึ่งต้นทางจะส่งสัญญาณไปแจ้งให้ปลายทางทราบว่าพร้อมที่จะดำเนินการและรอการตอบรับจากปลายทางจากนั้นจึงดำเนินการส่งข้อความ เมื่อข้อความเดินทางถึงผู้รับแล้ว ระบบจะดำเนินการอ่านข้อความ เขียนข้อความ รวมถึงการปิดเส้นทางการเชื่อมต่อลงเพื่อยืนยันการเสร็จสิ้นกระบวนการตามลำดับ การติดต่อสื่อสารโดยการรับ-ส่งข้อความแสดงดังภาพที่ 3.4

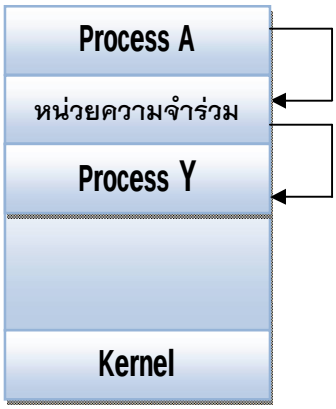


ภาพที่ 3.4 แสดงการรับ – ส่งข้อความ
ที่มา : พิรพร หมุนสนิทและคณะ (2553: 39)

จากภาพที่ 3.4 Process A ต้องการส่งข้อความไปยัง Process B ซึ่งอยู่ภายในระบบเดียวกัน โดยทำตามขั้นตอนดังนี้

1) Process A นำข้อความจาก Mailbox ส่งไปยัง Process B
2) การใช้หน่วยความจำร่วม (Share Memory)

เป็นการติดต่อสื่อสารกันระหว่างโปรเซสผ่านทางหน่วยความจำที่นำมาใช้ในการเก็บข้อมูล ซึ่งเป็นหน่วยความจำที่ถูกสร้างขึ้นโดยให้โปรเซสทั้งสองที่จะติดต่อสื่อสารกันแบ่งเป็นหน่วยความจำดังกล่าวร่วมกัน ในหน่วยความจำร่วมของทั้งสองโปรเซสจะเป็นส่วนที่ใช้ในการอ่านและเขียนข้อมูล ซึ่งอยู่นอกเหนือการควบคุมของระบบปฏิบัติการ ดังนั้นโปรเซสทั้งสองจะต้องรับผิดชอบการเขียนและอ่านข้อมูลด้วยตนเองรวมถึงต้องระวังไม่ให้การเขียนข้อมูลเกิดการซ้ำกันบนพื้นที่เดียวกัน ซึ่งการติดต่อสื่อสารด้วยการใช้หน่วยความจำร่วมแสดงดังภาพที่ 3.5



ภาพที่ 3.5 แสดงการรับ – ส่งข้อความการใช้หน่วยความจำร่วม
ที่มา : พิรพร หมุนสนิทและคณะ (2553: 39)

จากภาพที่ 3.5 จะเห็นได้ว่า Process X ต้องการติดต่อกับ Process Y โดยทำตามขั้นตอนดังนี้ 1) ดำเนินการใช้พื้นที่ในหน่วยความจำร่วมเพื่อใช้ในการเขียนและอ่านข้อมูล 2) Process Y เป็นผู้รับข้อมูลที่ถูกบันทึกลงอย่างถูกต้องในหน่วยความจำร่วมและนำไปใช้งานต่อไป ดังนั้น Kernel ที่เป็นศูนย์กลางในการควบคุมระบบปฏิบัติการจึงไม่มีส่วนในการติดต่อสื่อสารในครั้งนี้ การส่งต่อข้อมูลผ่านทางหน่วยความจำร่วม Process X และ Y จะเป็นผู้ดูแลและรับผิดชอบโครงการ

3.2.3 บริการของระบบปฏิบัติการ

การให้บริการของระบบปฏิบัติการถือเป็นอีกสิ่งหนึ่งที่สำคัญในโครงสร้างของระบบ เนื่องจากเป็นส่วนที่แสดงให้เห็นว่าระบบปฏิบัติการสามารถสนับสนุนงานใดได้บ้าง ทั้งนี้ระบบปฏิบัติการได้จัดเตรียมสภาพแวดล้อมและบริการต่างๆ สำหรับดำเนินการกับโปรแกรมไว้ให้ผู้ใช้งาน โดยแต่ละระบบปฏิบัติการอาจมีบริการที่แตกต่างกัน แต่ยังคงไว้ซึ่งบริการพื้นฐานที่เหมือนกัน โดยบริการต่างๆ จะคำนึงถึงความสะดวกสบายในการใช้งานของผู้ใช้เป็นหลักเพื่อให้ตรงตามวัตถุประสงค์ของระบบปฏิบัติการที่จะเป็นตัวประสานการใช้งานทรัพยากรในระบบอย่างสมบูรณ์ โดยบริการของระบบปฏิบัติการสรุปได้ดังต่อไปนี้

3.2.3.1 การประมวลผลโปรแกรม (Program Execution) เป็นบริการขั้นพื้นฐานของระบบปฏิบัติการ โดยทำหน้าที่สนับสนุนการเรียกโปรแกรมขึ้นมาใช้งาน ขั้นตอนการทำงานจะเริ่มต้นจากการดึงข้อมูลที่เกี่ยวข้องกับโปรแกรมมาเก็บไว้ที่หน่วยความจำหลักเพื่อให้โปรแกรมเริ่มทำงานและดำเนินการต่อไปเรื่อยๆ จนกว่างานทุกอย่างเสร็จเรียบร้อย

ตามที่ต้องการ หากมีการตรวจพบข้อผิดพลาดในระหว่างการใช้งานโปรแกรม ระบบจะทำการส่งข้อความแจ้งกลับมาทันที

3.2.3.2 การดำเนินการของอินพุต/เอาต์พุต (Input / Output Operation) ในระบบปฏิบัติการจำเป็นต้องมีการติดต่อกับอุปกรณ์หรือการนำเข้าและส่งออกข้อมูล จึงต้องมีบริการที่ทำหน้าควบคุมการทำงานของอินพุต/เอาต์พุตเพื่อรองรับและประสานการทำงานระหว่างโปรแกรม ข้อมูลและอุปกรณ์ เมื่อผู้ใช้สั่งการผ่านทางโปรแกรม คำสั่งดังกล่าวอาจมีความต้องการในการเรียกใช้อุปกรณ์หรือต้องการส่งข้อมูล ซึ่งผู้ใช้ไม่สามารถดำเนินกระบวนการเหล่านี้ได้ด้วยตนเอง ดังนั้นบริการนี้จึงเป็นตัวกลางในการจัดหาวิธีการหรืออุปกรณ์ให้ตามความเหมาะสม เช่น การสั่งพิมพ์เอกสารผ่านเครื่องพิมพ์และการใช้คำสั่งต่างๆ เพื่อให้แสดงผลบนหน้าจอ

3.2.3.3 การจัดการกับระบบไฟล์ข้อมูล (File-System Manipulation) เป็นบริการที่ใช้สำหรับจัดการกับระบบไฟล์หรือแฟ้มข้อมูลซึ่งเป็นสิ่งที่สำคัญอย่างมาก เนื่องจากระบบปฏิบัติการและโปรแกรมจะใช้ไฟล์ข้อมูลในการดำเนินการต่างๆ ไม่ว่าจะเป็นการแสดงผลและการบันทึก โดยระบบปฏิบัติการสามารถจัดการไฟล์ข้อมูลได้หลายลักษณะ เช่น สร้างไฟล์ข้อมูลใหม่ บันทึกไฟล์ข้อมูลและลบไฟล์ข้อมูล การจัดการไฟล์ข้อมูลในบางโปรแกรมอาจจำเป็นต้องมีการอ้างสิทธิ์ของผู้ใช้ในการเปิดไฟล์ข้อมูลเพื่อช่วยในการรักษาสิทธิ์และความปลอดภัยของไฟล์ข้อมูล หากไม่ได้รับสิทธิ์ก็จะไม่สามารถกระทำการใดๆ กับไฟล์ข้อมูลเหล่านั้นได้ ดังนั้น บริการนี้จึงมีความสำคัญอย่างยิ่งต่อระบบปฏิบัติการ อีกทั้งระบบจัดการไฟล์ข้อมูลที่มีประสิทธิภาพจะเป็นตัวสนับสนุนให้การดำเนินการของโปรแกรมเป็นไปอย่างราบรื่นถูกต้องและสมบูรณ์ด้วย

3.2.3.4 การติดต่อสื่อสาร (Communication) ระบบปฏิบัติการที่มีจำนวนโปรเซสดำเนินการอยู่มากกว่าหนึ่งโปรเซสเหล่านั้นอาจจำเป็นต้องมีการติดต่อสื่อสารระหว่างกันเพื่อให้ดำเนินการได้อย่างสมบูรณ์ การติดต่อสื่อสารนี้อาจเกิดขึ้นระหว่างโปรเซสที่ดำเนินการอยู่ภายในเครื่องคอมพิวเตอร์เดียวกันหรือติดต่อกับโปรเซสที่อยู่ในเครื่องอื่นก็ได้ โดยอาศัยระบบปฏิบัติการเป็นตัวกลางในการติดต่อสื่อสารและส่งข้อมูลในรูปแบบแพ็กเกจหรือข้อความผ่านทางระบบเครือข่าย การสื่อสารดังกล่าวอาจใช้หน่วยความจำร่วม (Shared Memory) หรืออาจใช้เทคนิคที่เรียกว่า “การรับ-ส่งข้อความ (Message Passing)” เพื่อช่วยในการโอนถ่ายแพ็กเกจข้อมูลทั้งภายในและภายนอกโปรเซส

3.2.3.5 การตรวจจับข้อผิดพลาด (Error Detection) ระบบปฏิบัติการนอกจากจะสนับสนุนและคอยประสานการทำงานแล้ว จำเป็นต้องมีระบบตรวจจับหรือ

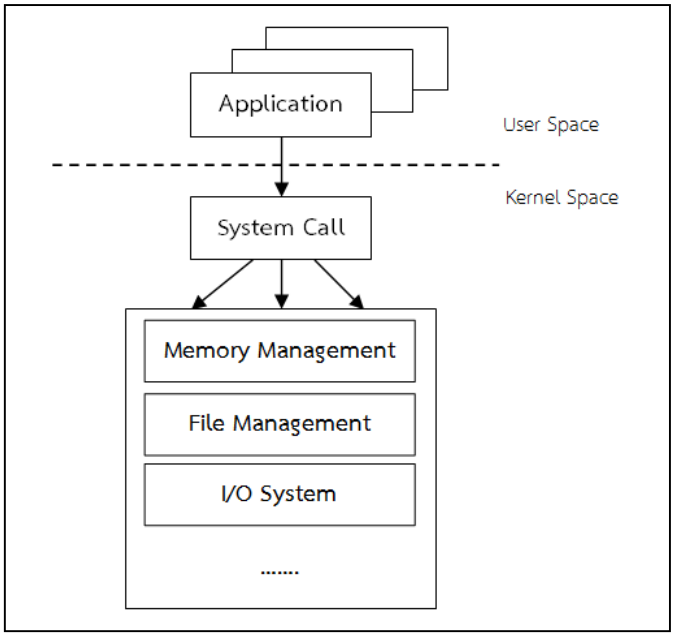
ให้บริการที่เกี่ยวข้องกับการตรวจสอบเพื่อหาข้อผิดพลาดที่อาจเกิดขึ้นและส่งผลต่อการทำงานโดยรวมของระบบ บริการนี้เป็นเสมือนกลไกในการตรวจจับข้อผิดพลาดต่างๆ ที่เกิดขึ้นได้เสมอทั้งจากอุปกรณ์และโปรแกรมซึ่งข้อผิดพลาดที่เกิดขึ้นแต่ละประเภท ระบบจะจัดเตรียมวิธีการที่สามารถแก้ไขปัญหาได้ในเบื้องต้นไว้รองรับหรือจัดการหาวิธีแก้ไขที่เหมาะสมที่สุดกับข้อผิดพลาดดังกล่าว เช่น การติดต่อสื่อสารระหว่าง โปรเซส ผ่านระบบเครือข่ายเกิดล้มเหลว ทำให้การแลกเปลี่ยนข้อมูลไม่สำเร็จ ระบบจึงต้องมีการตรวจจับข้อผิดพลาดดังกล่าวเพื่อแจ้งให้ผู้ใช้ทราบหรือดำเนินการแก้ไขปัญหาตามวิธีการที่เหมาะสม

3.3 สถาปัตยกรรมของระบบปฏิบัติการ

ระบบปฏิบัติการได้รับการพัฒนามาอย่างต่อเนื่องจนมีความสามารถในการสนับสนุน การใช้งานของผู้ใช้ ซึ่งมีหน้าที่ในการประสานงานกับองค์ประกอบต่างๆ ในระบบ มีบริการที่รองรับต่อรูปแบบการใช้งานที่มีความหลากหลายมากขึ้น ด้วยเหตุนี้ทำให้ระบบปฏิบัติการมีความซับซ้อนมากขึ้นตามประสิทธิภาพใช้งานที่สูงขึ้น สถาปัตยกรรมของระบบปฏิบัติการเป็นอีกสิ่งหนึ่งที่ช่วยให้นักออกแบบและนักพัฒนาสามารถสร้างสรรค์ระบบปฏิบัติการได้ สถาปัตยกรรมของระบบปฏิบัติการที่สำคัญมีดังนี้ (พิรพร หมุนสนิทและคณะ, 2553: 40-43)

3.3.1 สถาปัตยกรรม Monolithic Architecture

เป็นสถาปัตยกรรมของระบบปฏิบัติการในยุคแรกๆ ซึ่งไม่มีความซับซ้อนและมีรูปแบบในการนำเสนอที่ง่าย กล่าวคือ ทุกๆ องค์ประกอบ (Component) ในระบบปฏิบัติการจะถูกนำมาบรรจุไว้ที่ศูนย์กลางของระบบ (Kernel) ทั้งหมด ช่วยให้องค์ประกอบอื่นๆ สามารถติดต่อสื่อสารกันได้โดยตรง สถาปัตยกรรมแบบ Monolithic จะรวมองค์ประกอบต่างๆ ไว้ด้วยกันทั้งหมดทำให้ประสิทธิภาพการทำงานค่อนข้างสูง อีกทั้งแต่ละองค์ประกอบสามารถติดต่อกันได้ง่ายเพราะรวมอยู่เป็นกลุ่มทำให้ในบางกรณีการตรวจสอบหาข้อผิดพลาดหรือแหล่งกำเนิดของความผิดพลาดภายในกลุ่มขององค์ประกอบดังกล่าวทำได้ค่อนข้างยาก เนื่องจากทุกองค์ประกอบต่างประสานการทำงานเป็นกลุ่มและสามารถติดต่อและเข้าถึงได้ง่าย ทำให้ง่ายต่อการที่จะเกิดความเสียหายจากโค้ด (Code) ที่ไม่ประสงค์ดีต่อระบบสถาปัตยกรรม Monolithic แสดงดังภาพที่ 3.6



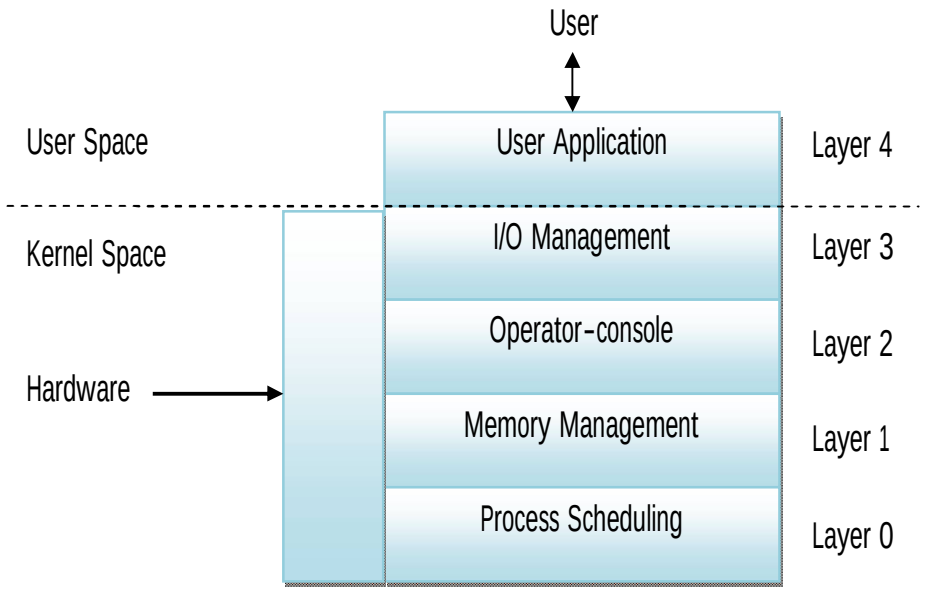
ภาพที่ 3.6 แสดง Monolithic Architecture
ที่มา : พิรพร หมุนสนิทและคณะ (2553: 40)

จากภาพที่ 3.6 แอปพลิเคชันในส่วนผู้ใช้จะติดต่อผ่านทางส่วนต่อประสาน ซึ่งมี System Call เป็นตัวส่งสัญญาณเรียกใช้งานระบบ เพื่อให้ระบบปฏิบัติการรับหน้าที่เปิดองค์ประกอบสำหรับให้บริการแอปพลิเคชั่นนั้น โดยแต่ละแอปพลิเคชั่นมีความต้องการองค์ประกอบหรือบริการจากระบบปฏิบัติการที่แตกต่างกัน องค์ประกอบต่างๆ ภายในระบบปฏิบัติการอาจมีจำนวนไม่เท่ากันขึ้นอยู่กับระบบปฏิบัติการนั้นว่ามีรูปแบบการทำงานอย่างไร แต่ส่วนใหญ่จะเป็นระบบการสนับสนุนขั้นพื้นฐาน เช่น การจัดการโปรเซส การจัดการหน่วยความจำและการจัดการไฟล์ข้อมูล ระบบปฏิบัติการที่มีลักษณะคล้ายกับสถาปัตยกรรม Monolithic เช่น MS-DOS, Windows 9x และ Linux

3.3.2 สถาปัตยกรรม Layered Architecture

เป็นสถาปัตยกรรมที่เหมาะสำหรับระบบปฏิบัติการขนาดใหญ่และมีความซับซ้อนมากขึ้น โครงสร้างของระบบปฏิบัติการถูกแบ่งออกเป็นระดับชั้น (Layer) ตามความสัมพันธ์ของแต่ละองค์ประกอบ โดยรวมองค์ประกอบที่มีหน้าที่และฟังก์ชันการทำงานคล้ายๆ กัน รวมเป็นกลุ่มเดียวกันแล้วแบ่งเป็นระดับชั้น ซึ่งแต่ละชั้นสามารถติดต่อกับชั้นที่อยู่สูงและต่ำกว่าได้ โดยชั้นที่อยู่ต่ำกว่าจะคอยให้บริการกับชั้นที่อยู่ระดับสูงกว่า การแบ่งเป็นระดับชั้นจะแตกต่างจากสถาปัตยกรรมแบบ Monolithic ที่รวมองค์ประกอบเข้าไว้ด้วยกันทั้งหมด ทำให้การพัฒนาและเปลี่ยนแปลงใดๆ สามารถทำได้โดยตรงในแต่ละระดับชั้น

โดยไม่ต้องแก้ไขทุกระดับชั้นเพราะแต่ละระดับชั้นมีความเป็นอิสระจากกัน แต่การให้บริการจะต้องให้ข้อมูลเดินทางผ่านทุกๆ ชั้น ซึ่งแต่ละชั้นต่างก็มีบริการเป็นของตนเองและมีหน้าที่เฉพาะตัว หากมีข้อผิดพลาดเกิดขึ้นสามารถตรวจหาได้ง่ายกว่าเมื่อพบว่าผิดพลาดที่ระดับชั้นใด ก็ทำการแก้ไขได้โดยตรง อย่างไรก็ตามสถาปัตยกรรมรูปแบบนี้ยังไม่สามารถป้องกันตนเองจาก Code บางชนิดที่ประสงค์ร้ายต่อระบบ ซึ่งอาจทำให้ระบบเกิดความเสียหายได้ แนวคิดในการแบ่งระดับชั้นได้นำมาสร้างเป็นแบบจำลองระบบปฏิบัติการ THE (Technische Hogeschool Eindhover) ดังแสดงในภาพที่ 3.7



ภาพที่ 3.7 แสดงระดับชั้นของ The Operating System
ที่มา : พิรพร หมุนสนิทและคณะ (2553:41)

จากภาพที่ 3.7 จะเห็นได้ว่าระดับชั้นจะถูกแบ่งเป็น 2 ฝั่ง คือ ฝั่งผู้ใช้ (User) และฝั่งระบบปฏิบัติการ (Kernel) ซึ่งระดับชั้นบนสุดจะอยู่ในส่วนของผู้ใช้ ส่วนที่เหลือจะอยู่ใน Kernel และเป็นระดับชั้นที่เกี่ยวข้องกับฮาร์ดแวร์ โดยแต่ละชั้นจะมีหน้าที่แตกต่างกันดังนี้

ชั้นที่ 0 Process Scheduling

ทำหน้าที่จัดตารางการทำงานของ CPU ว่าจะมีลำดับในการประมวลผลงานหรือข้อมูลต่างๆ ที่นำเข้ามาอย่างไรเพื่อให้ โปรเซส ต่างๆ ในระบบมีลำดับการทำงานที่ถูกต้อง

ชั้นที่ 1 Memory Management

ทำหน้าที่ในการจัดการกับหน่วยความจำในขณะที่ดำเนินการโปรเซสต่างๆ

ชั้นที่ 2 Operator-console

ทำหน้าที่ดำเนินการติดต่อสื่อสารระหว่างระบบปฏิบัติการกับส่วนใช้งาน (Console) ซึ่งจะใช้การส่งสัญญาณจังหวะ (Synchronization) ด้วยรูปแบบหรืออัลกอริธึมที่เหมาะสมเพื่อหลีกเลี่ยงการเกิดภาวการณ์ติดตาย (Deadlock)

ชั้นที่ 3 I/O Management

มีหน้าที่ในการจัดการกับอุปกรณ์อินพุต/เอาต์พุตสำหรับรับ-ส่งข้อมูลต่างๆ ภายในระบบปฏิบัติการ โดยจะจัดสรรทรัพยากรทุกอย่างที่จำเป็นต่อการใช้งานอุปกรณ์และสนับสนุนการทำงานของอุปกรณ์ดังกล่าว

ชั้นที่ 4 User Application

เป็นชั้นที่อยู่ในส่วนของผู้ใช้ ซึ่งเกี่ยวข้องกับการทำงานของโปรแกรมต่างๆ ซึ่งอาจมีกระบวนการพื้นฐานต่างๆ ที่เกิดขึ้นในระดับชั้นนี้ เช่น การคอมไพล์ (Compilation) การดำเนินการ (Execution) และการพิมพ์ (Printing)

การแบ่งระดับชั้นของสถาปัตยกรรมรูปแบบนี้ จะขึ้นอยู่กับความซับซ้อนของระบบปฏิบัติการ สำหรับระบบปฏิบัติการที่นำสถาปัตยกรรมรูปแบบนี้ไปใช้ในการแบ่งระดับชั้น เช่น Unix และ OS/2

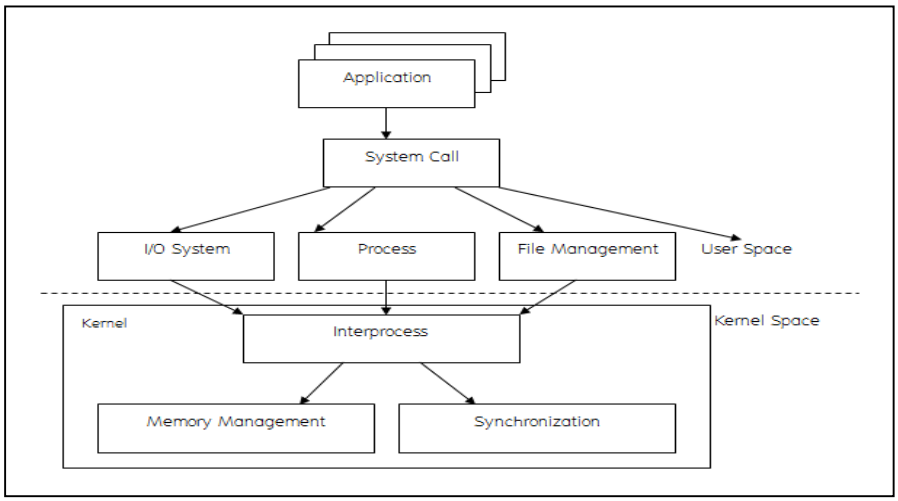
3.3.3 สถาปัตยกรรม Microkernel Architecture

เป็นสถาปัตยกรรมของระบบปฏิบัติการที่มีจำนวนการให้บริการต่างๆ น้อยกว่ารูปแบบอื่นโดยมีวัตถุประสงค์เพื่อรักษาขนาดของ Kernel ให้เหมาะสมภายใน Kernel ประกอบด้วย 3 บริการ ได้แก่

3.3.3.1 การจัดการหน่วยความจำ (Memory Management)

3.3.3.2 การติดต่อสื่อสารระหว่างโปรเซส (Interposes Communication)

3.3.3.3 การส่งสัญญาณจังหวะ (Synchronization) ส่วนบริการที่เหลือจะถูกบรรจุไว้นอก Kernel ซึ่งอยู่ในส่วนของผู้ใช้ดังแสดงในภาพที่ 3.8



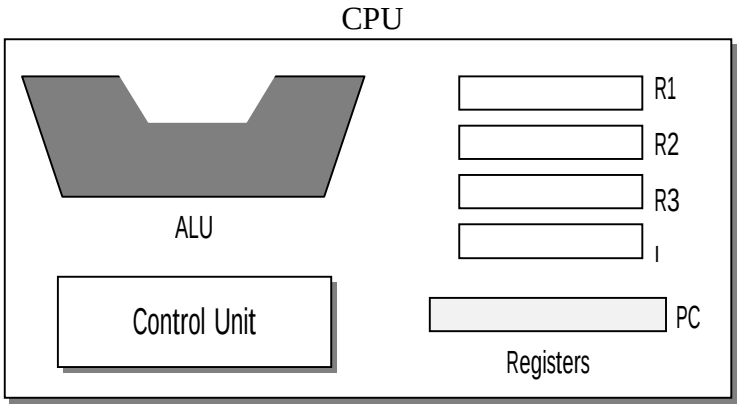
ภาพที่ 3.8 แสดงMicrokernel Architecture
ที่มา : พิรพร หมุนสนิทและคณะ (2553: 43)

สถาปัตยกรรม Microkernel Architecture มีความกะทัดรัดและมีขนาดของ Kernel ที่พอดีซึ่งสามารถขยายได้ตามความเหมาะสมทำให้รูปแบบการดำเนินการต่างๆ ถูกกระจายไปยังองค์ประกอบอื่นๆ ภายในระบบปฏิบัติการ โดยไม่จับกลุ่มอยู่ส่วนใดส่วนหนึ่งของ Kernel เมื่อเกิดความผิดพลาดขึ้นกับองค์ประกอบประเภทหนึ่งก็จะไม่ส่งผลกระทบต่อองค์ประกอบอื่นๆ ภายในระบบปฏิบัติการ อย่างไรก็ตาม การกระจายและจำนวนขององค์ประกอบที่มีอยู่มาก Kernel อาจเป็นตัวการทำให้ประสิทธิภาพการแสดงผลด้อยลงได้ สถาปัตยกรรมรูปแบบนี้ถูกนำไปใช้กับระบบปฏิบัติการต่างๆ ได้แก่ 1) Windows 2000 2) Windows XP และ 3) Mac OS X

3.4 โครงสร้างพื้นฐานของไมโครโปรเซสเซอร์

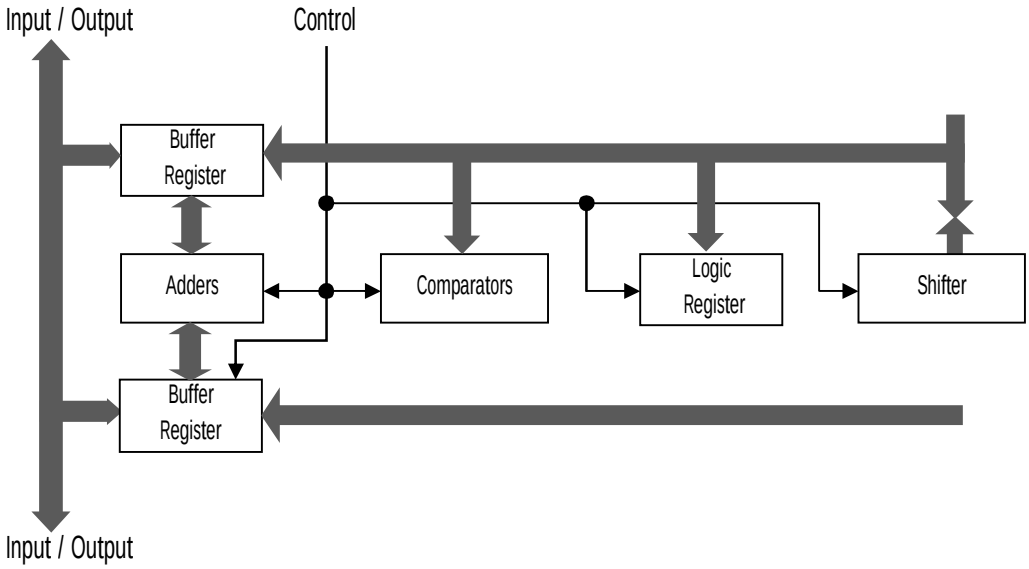
โครงสร้างพื้นฐานของไมโครโปรเซสเซอร์ประกอบด้วยหน่วยต่างๆ ซึ่งผู้สอนได้ศึกษาเกี่ยวกับโครงสร้างพื้นฐานภายในไมโครโปรเซสเซอร์จากนักวิชาการดังนี้ 1) ไพศาล โมลิสกุลมงคล (2547: 14-15) 2) ธีรวัฒน์ ประกอบผลและจันทนา ผ่องเพ็ญศรี (2551: 88-93) 3) ธีรณี อจลากุล (2556: 38-52) แล้วนำมาสรุปโดยกล่าวถึงโครงสร้างพื้นฐานของไมโครโปรเซสเซอร์ว่าประกอบด้วยส่วนประกอบหลัก คือ หน่วยประมวลผลทางคณิตศาสตร์และลอจิก (ALU: Arithmetic and Logic Unit) หน่วยควบคุม (Control Unit) และรีจิสเตอร์ (Register) ถ้าหากมองเป็นโครงสร้างอย่างง่ายจะได้ดังภาพที่ 3.9 หน่วย ALU ถือว่าเป็นหัวใจ

หลักของไมโครโปรเซสเซอร์ การคำนวณทั้งหมดจะเกิดขึ้นในหน่วยนี้ สำหรับรีจิสเตอร์จะมีได้หลายตัว ในตัวอย่างนี้จะให้ชื่อเป็น R1,R2 และ R3 เป็นรีจิสเตอร์สำหรับใช้งานทั่วไป รีจิสเตอร์ I สำหรับใช้เก็บคำสั่ง ส่วนรีจิสเตอร์ PC ใช้สำหรับชี้คำสั่งที่ซีพียูต้องการประมวลผล



ภาพที่ 3.9 แสดงโครงสร้างภายในซีพียู
ที่มา : ธีรวัฒน์ ประกอบผลและจันทนา ผ่องเพ็ญศรี (2551: 88)

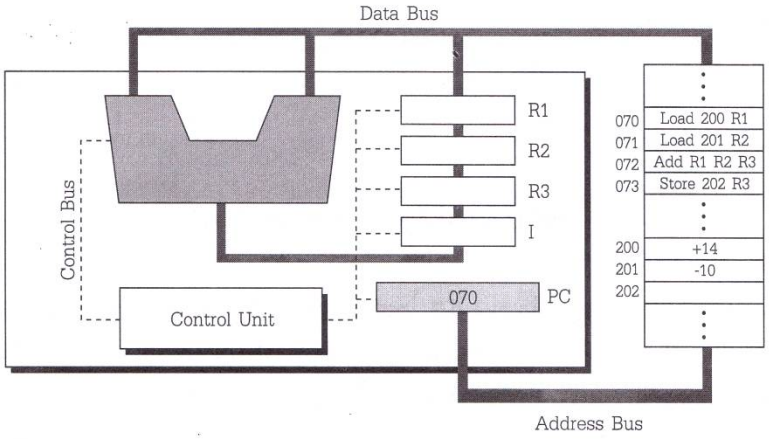
จากภาพที่ อธิบายได้ว่า ภายใน ALU จะประกอบด้วยหน่วยประมวลผลทางคณิตศาสตร์และหน่วยประมวลผลทางลอจิกต่างๆ วงจรภายในมีความซับซ้อนมาก ผู้ผลิตไมโครโปรเซสเซอร์มักจะไม่เปิดเผยรายละเอียดของวงจร ในส่วนนี้หากมองเป็นไดอะแกรมย่อยๆ จะได้ดังภาพที่ 3.10 หาก ALU ต้องการประมวลผลกับข้อมูลใด หน่วยควบคุมจะทำหน้าที่นำข้อมูลที่ต้องการส่งเข้ามาเก็บในรีจิสเตอร์บัฟเฟอร์ (Buffer Register) โดย Adder ทำหน้าที่เป็นวงจรบวกและวงจรลบ โดยตัวเปรียบเทียบ(Comparators) ทำหน้าที่เปรียบเทียบข้อมูลสองค่า ตัวรีจิสเตอร์ลอจิก (Logic Register) จะใช้สำหรับการกระทำทางลอจิกของข้อมูล ส่วน Shifter จะใช้สำหรับการเลื่อนและหมุนข้อมูล หากต้องการประมวลผลอย่างไร หน่วยควบคุมจะเป็นตัวส่งสัญญาณไปบอกกับ ALU ว่าให้ประมวลผลตามที่ต้องการ



ภาพที่ 3.10 แสดงโครงสร้างอย่างง่ายของ ALU
ที่มา : ธีรวัฒน์ ประกอบผลและจันทนา ผ่องเพ็ญศรี (2551: 88)

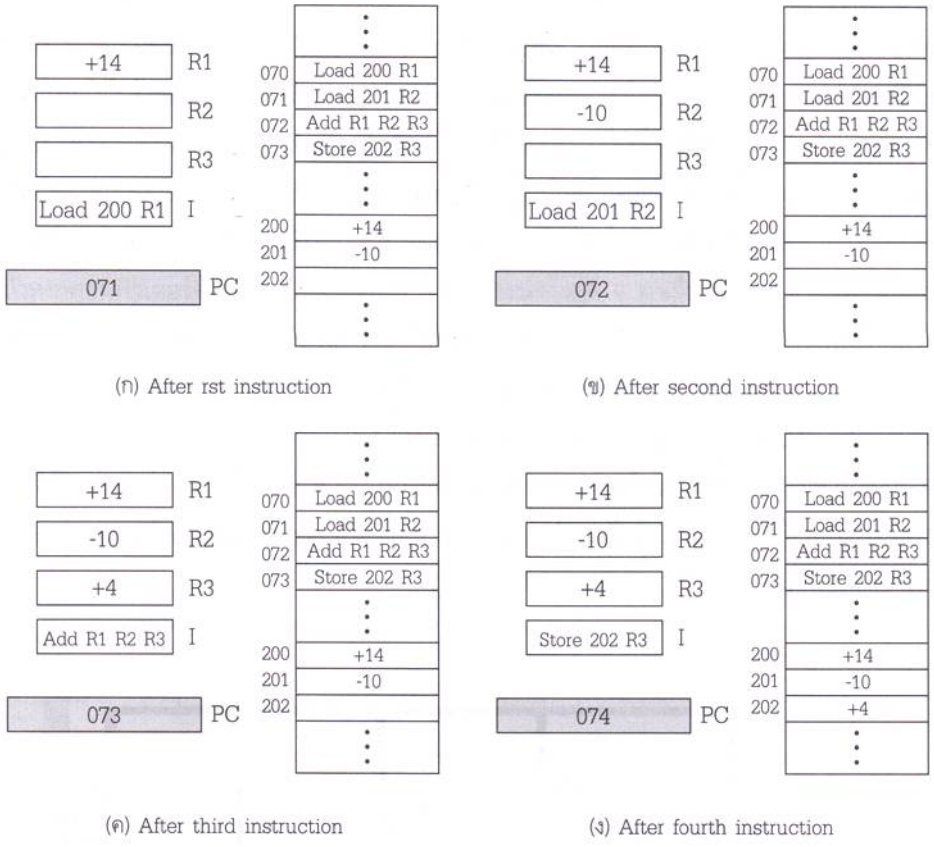
รีจิสเตอร์เป็นอุปกรณ์เก็บข้อมูลชั่วคราวที่อยู่ภายในซีพียู ขนาดของรีจิสเตอร์ของซีพียูแต่ละรุ่นก็จะแตกต่างกันไป ตัวอย่างเช่น ซีพียูที่ประมวลผลแบบ 8 บิต มักจะมีรีจิสเตอร์ขนาด 8 บิต ซีพียูที่ประมวลผลแบบ 16 บิต จะมีรีจิสเตอร์ขนาด 16 บิต ในการติดต่อกับซีพียูนั้นหากมีกลุ่มข้อมูลที่ติดต่อกับซีพียูก็มักจะเป็นการติดต่อกับรีจิสเตอร์เสมอ รีจิสเตอร์ภายในซีพียูนั้นสามารถแบ่งได้เป็นรีจิสเตอร์สำหรับใช้งานทั่วไปและรีจิสเตอร์ที่มีหน้าที่เฉพาะด้าน โดยทั่วไปแล้วซีพียูหลายๆ รุ่นมักจะมีรีจิสเตอร์ตัวหนึ่งที่สามารถติดต่อกับ ALU และพอร์ตได้เร็วมาก รีจิสเตอร์ตัวนี้มีชื่อว่า แอกคูมิวเลเตอร์ (Accumulator)

หากมีไมโครโปรเซสเซอร์ดังภาพที่ 3.11 โดยภายในมีรีจิสเตอร์สำหรับใช้งาน คือ R1,R2 และ R3 ส่วนรีจิสเตอร์ I ถูกออกแบบมาสำหรับเก็บคำสั่ง ละรีจิสเตอร์ PC เป็นรีจิสเตอร์สำหรับชี้ตำแหน่งหน่วยความจำที่ต้องการติดต่อ ระบบคอมพิวเตอร์ที่ออกแบบขึ้นนี้จะมีหน่วยความจำต่ออยู่โดยเชื่อมต่อกับไมโครโปรเซสเซอร์ผ่านทางบัสข้อมูลและบัสแอดเดรส ในหน่วยความจำถูกแบ่งเป็นส่วนที่ใช้เก็บโปรแกรมและส่วนที่ใช้เก็บข้อมูลโดยโปรแกรมเก็บอยู่ในหน่วยความจำตั้งแต่ตำแหน่ง 070 เป็นต้นไป ส่วนข้อมูลเก็บอยู่ในหน่วยความจำตั้งแต่ตำแหน่ง 200 เป็นต้นไป



ภาพที่ 3.11 แสดงระบบคอมพิวเตอร์โดยมีโปรแกรมอยู่ในหน่วยความจำ
ที่มา : ธีรวัฒน์ ประกอบผลและจันทนา ผ่องเพ็ญศรี (2551: 91)

รีจิสเตอร์ PC จะต่ออยู่กับแอดเดรสบัสเพื่ออ้างตำแหน่งหน่วยความจำ เมื่อไมโครโปรเซสเซอร์เริ่มทำโปรแกรม คำสั่งแรกจะทำงานคำสั่งที่รีจิสเตอร์ PC ชี้อยู่ คือ ตำแหน่งที่ 070 โดยทำการอ่านรหัสคำสั่งนี้เข้ามาเก็บในรีจิสเตอร์ I เมื่อหน่วยควบคุมตีความข้อมูลในรีจิสเตอร์ I เมื่อหน่วยควบคุมตีความข้อมูลในรีจิสเตอร์ I จะพบว่าคำสั่งนี้ให้อ่านค่าจากหน่วยความจำตำแหน่ง 200 มาเก็บในรีจิสเตอร์ R1 ดังนั้น หลังจากทำคำสั่งนี้จะทำให้ R1 มีค่าเป็น +14 และ PC จะชี้ไปยังตำแหน่งถัดไปคือตำแหน่ง 071 โดยขั้นตอนการทำงานต่างๆ จะเป็นดังภาพที่ 3.12

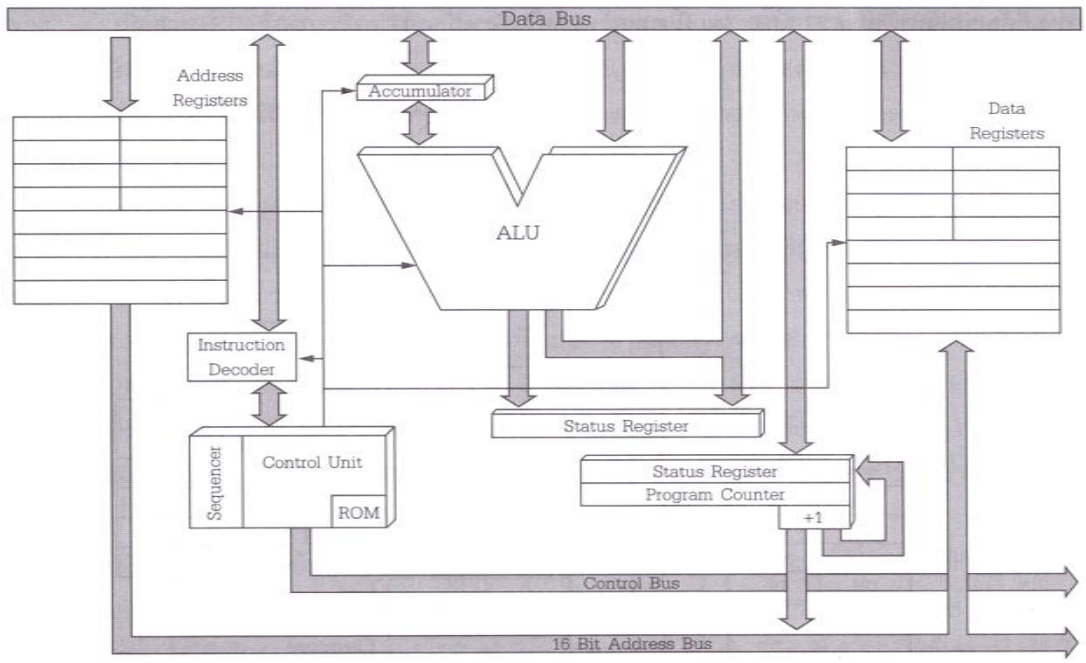


ภาพที่ 3.12 แสดงการทำคำสั่งต่างๆ ของระบบคอมพิวเตอร์

ที่มา : ธีรวัฒน์ ประกอบผลและจันทนา ผ่องเพ็ญศรี (2551: 92)

ต่อมาทำคำสั่งที่เก็บอยู่ในตำแหน่ง 071 โดยจะอ่านรหัสคำสั่งจากตำแหน่งนี้มาเก็บในรีจิสเตอร์ I เมื่อหน่วยควบคุมตีความจะพบว่าให้อ่านค่าจากตำแหน่งหน่วยความจำ 201 มาเก็บใน R2 หลังจากทำคำสั่งนี้จะทำให้รีจิสเตอร์ R2 มีค่าเป็น -10 และ PC จะเก็บค่าตำแหน่งถัดไปคือ 072 เมื่อทำคำสั่งที่สามโดยเริ่มอ่านรหัสคำสั่งจากตำแหน่ง 072 มาเก็บไว้ในรีจิสเตอร์ I หน่วยควบคุมจะตีความว่าให้บวกข้อมูลใน R1 และ R2 ผลลัพธ์ที่ได้เก็บใน R3 เมื่อระบบทำงาน หน่วยควบคุมจะนำค่าใน R1 ส่งไปเก็บในบัฟเฟอร์ของ ALU และนำ R2 ส่งไปเก็บในบัฟเฟอร์ของ ALU เช่นกัน จากนั้นหน่วยควบคุมจะสั่งงานให้ ALU ทำหน้าที่เป็นวงจรบวกแล้วนำผลลัพธ์ที่ได้ออกมาส่งไปเก็บใน R3 จากนั้นรีจิสเตอร์ PC จะชี้ไปยังแอดเดรสตำแหน่งถัดไป คือ ตำแหน่ง 073 เมื่อทำคำสั่งที่สี่ ระบบจะไปอ่านระบบคำสั่งจากแอดเดรส 073 มาเก็บไว้ในรีจิสเตอร์ I เมื่อตีความจะพบว่าคำสั่งนี้ให้นำข้อมูลจากรีจิสเตอร์ R3 ไปเก็บในหน่วยความจำ 202 หน่วยควบคุมก็จะส่งค่าแอดเดรสบัสออกไปเป็น 202 และนำข้อมูลที่เก็บ

อยู่ในรีจิสเตอร์ R3 ส่งออกไปทางบัสข้อมูลแล้วไปเก็บยังตำแหน่ง 202 จากนั้น PC จะชี้ไปยังตำแหน่งของคำสั่งต่อไป จากนั้นขั้นตอนการทำงานต่างๆ ในรูปที่ 4.9 จะพบว่าการทำงานของไมโครโปรเซสเซอร์ทั่วไปจะอยู่ในรูปของ อ่าน–ถอดรหัส–ทำตามคำสั่ง (Fetch–Decoder–Execute) โดยในการอ่านรหัสคำสั่งจะอ่านเข้ามาทีละคำสั่ง โดยรีจิสเตอร์ PC จะนำข้อมูลที่เก็บอยู่ส่งออกไปทางแอดเดรสบัสเพื่ออ้างตำแหน่งหน่วยความจำและหน่วยควบคุมจะเป็นส่วนที่ควบคุมการทำงานและการใช้บัสทั้งหมด การทำงานในลักษณะนี้ทำให้ไมโครโปรเซสเซอร์มีการออกแบบส่วนประกอบเพิ่มเติมขึ้นมาดังภาพที่ 3.13



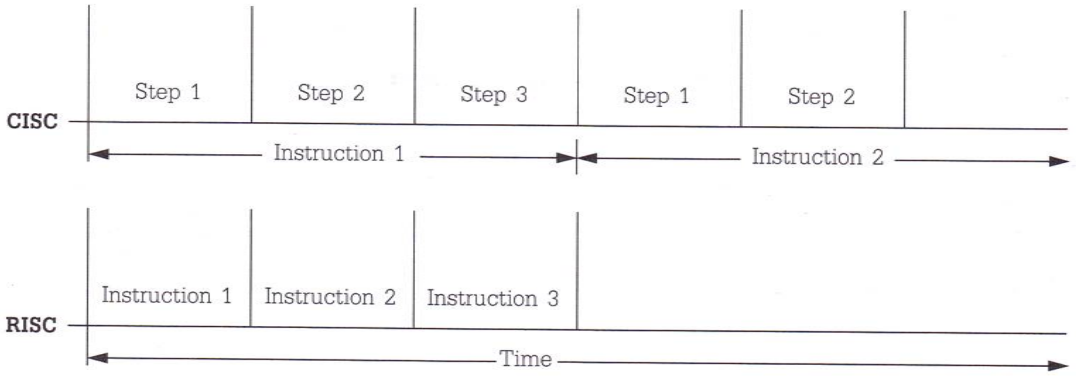
ภาพที่ 3.13 แสดงส่วนประกอบเพิ่มเติมภายในไมโครโปรเซสเซอร์
ที่มา : ธีรวัฒน์ ประกอบผลและจันทนา ผ่องเพ็ญศรี (2551: 93)

จากโครงสร้างของไมโครโปรเซสเซอร์ในภาพที่ 3.13 ถ้าหากต้องการอ้างตำแหน่งหน่วยความจำภายนอกจะใช้รีจิสเตอร์แอดเดรสในการอ้างโดยจะส่งสัญญาณออกมาทางแอดเดรสบัส หากแอดเดรสบัสมีขนาด 16 บิต ทำให้อ้างแอดเดรสได้ 65,536 ตำแหน่ง มีกลุ่มรีจิสเตอร์สำหรับเก็บข้อมูลทั่วไป มีแอกคูมิวเลเตอร์สำหรับติดต่อกับ ALU และเมื่อ ALU ประมวลผลออกมาแล้วลักษณะของผลลัพธ์ที่ได้จะถูกเก็บไว้ในรีจิสเตอร์สถานะ (Status Register) เมื่อซีพียูต้องการทำคำสั่งใดก็จะนำตำแหน่งคำสั่งซึ่งเก็บอยู่ในรีจิสเตอร์โปรแกรมเคาน์เตอร์ (PC) ส่งออกไปทางแอดเดรสบัสและอ่านรหัสคำสั่งเข้ามาเก็บในรีจิสเตอร์คำสั่ง

(Instruction Register) จากนั้นหน่วยควบคุมจะถอดรหัสคำสั่งและสั่งงานให้วงจรต่างๆ ทำงานตามคำสั่งนั้น โดยที่รีจิสเตอร์ PC จะมีค่าเพิ่มขึ้นหนึ่งค่าเพื่อชี้ไปยังตำแหน่งคำสั่งถัดไป

3.5 สถาปัตยกรรมแบบ CISC และ RISC

สถาปัตยกรรมตามชุดคำสั่งระบบปฏิบัติการแบ่งได้เป็น 2 กลุ่ม คือ สถาปัตยกรรมคำสั่งแบบ CISC (Complex Instruction Set Computer) และสถาปัตยกรรมคำสั่งแบบ RISC (Reduced Instruction Set Computer) ดังที่ ธีรวัฒน์ ประกอบผลและจันทนา ผ่องเพ็ญศรี (2551: 106-109) กล่าวว่า ไมโครโปรเซสเซอร์รุ่นเก่าๆ นั้นจะเป็นสถาปัตยกรรมแบบ CISC ทั้งสิ้น แต่ในปัจจุบันจะแยกความแตกต่างระหว่างสถาปัตยกรรมของไมโครโปรเซสเซอร์ทั้ง 2 ประเภทได้ยาก เนื่องจากชุดคำสั่งทั้งสองถูกออกแบบให้มีลักษณะใกล้เคียงกันตามที่ทราบมาแล้วว่าการทำงานของคำสั่งนั้นจะประกอบด้วยส่วนที่เป็นออปโค้ด (Opcode) และส่วนที่เป็นโอเปอร์แรนด์ (Operand) ในการทำคำสั่งแบบ CISC จะมีกระบวนการย่อยๆ ประกอบอยู่ภายใน แต่ละคำสั่งจะมีขั้นตอนการทำงานที่ไม่เท่ากันได้และมีขนาดและรหัสคำสั่งที่ไม่เท่ากันได้ ขึ้นอยู่กับว่าคำสั่งประเภทนั้นจะทำงานอย่างไร ส่วนคำสั่งแบบ RISC คำสั่งทุกคำสั่งของไมโครโปรเซสเซอร์จะมีความยาวของรหัสคำสั่งเท่ากันทั้งหมด



ภาพที่ 3.14 แสดงการทำคำสั่งของคำสั่งแบบ CISC และ RISC

ที่มา : ธีรวัฒน์ ประกอบผลและจันทนา ผ่องเพ็ญศรี (2551: 106)

3.5.1 สถาปัตยกรรมแบบ CISC

ไมโครโปรเซสเซอร์ของอินเทลในตระกูล Intel X86 เป็นตัวอย่างของไมโครโปรเซสเซอร์ที่ใช้สถาปัตยกรรมแบบ CISC สถาปัตยกรรมแบบนี้จะออกแบบให้ไมโครโปรเซสเซอร์มีชุดคำสั่งที่สามารถนำมาเขียนโปรแกรมได้ง่าย คำสั่งต่างๆ จะมีประสิทธิภาพในการทำงานสูงเพื่อให้นักโปรแกรมสามารถเขียนโปรแกรมได้ง่ายที่สุด ในไมโครโปรเซสเซอร์จะมีคำสั่งมากมาย คำสั่งบางคำสั่งสามารถทำงานที่ซับซ้อนได้และบางคำสามารถอ้างแอดเดรสได้หลายแบบ ทำให้ไมโครโปรเซสเซอร์ประเภทนี้มีคำสั่งต่างๆ ที่ทำงานได้ซับซ้อนแต่ก็ทำให้การออกแบบกลไกภายในซับซ้อนมากขึ้นด้วย

สำหรับการอ้างแอดเดรสได้หลายแบบนั้นจะทำให้คำสั่งได้ทำงานได้หลากหลาย แต่ก็จะทำให้คำสั่งของไมโครโปรเซสเซอร์มีความยาวที่แตกต่างกันได้ เนื่องจากโอเปอร์แรนด์ที่จะนำข้อมูลมาในการอ้างแอดเดรสแต่ละแบบนั้นมีขนาดต่างกัน ตัวอย่างเช่น คำสั่งของไมโครโปรเซสเซอร์ 80x86 โดยทั่วๆ ไปจะมีความยาวอยู่ระหว่าง 2-6 ไบต์ ส่วนคำสั่งที่ทำงานยากๆ จะใช้หน่วยความจำถึง 13 ไบต์ และเนื่องจากการอ้างแอดเดรสที่มีจำนวนมากนั้นทำให้การเขียนโปรแกรมเพื่ออ้างตำแหน่งหน่วยความจำต่างๆ สามารถทำได้ง่าย โครงสร้างภายในไมโครโปรเซสเซอร์จึงไม่ต้องมีรีจิสเตอร์จำนวนมาก ซึ่งการเพิ่มจำนวนรีจิสเตอร์นั้นจะทำให้โครงสร้างภายในมีความหนาแน่นของวงจรซับซ้อนยิ่งขึ้นไปอีก

คำสั่งที่มีความสามารถมากและอ้างแอดเดรสได้หลายแบบจะทำให้การทำงานภายในซับซ้อน การทำคำสั่งแต่ละคำสั่งจะมีขั้นตอนการทำงานหลายขั้นตอน ดังภาพที่ 3.12 เวลาที่ใช้ในแต่ละคำสั่งจะใช้เวลามาก การออกแบบไมโครโปรเซสเซอร์รุ่นต่อๆ มาจึงมีการแก้ไขปัญหาโดยนำการทำงานแบบไปป์ไลน์ (Pipelining) มาใช้แต่ก็จะเกิดปัญหาใหม่เมื่อมี การทำคำสั่งบางคำสั่งที่มีความยาวไม่เท่ากัน ตัวอย่างเช่น ถ้าหากไมโครโปรเซสเซอร์มีการทำงาน 2 Stage คือ เฟ็ตช์ (Fetch) และเอ็กซิคิวต์ (Execute) อาจเกิดปัญหาตรงที่การ Fetch หรือ Execute อาจไม่สามารถทำเสร็จได้ในเวลาเดียวกัน

3.5.2 สถาปัตยกรรมแบบ RISC

วิธีการปรับปรุงอัตราการประมวลผลเพื่อให้ไมโครโปรเซสเซอร์สามารถทำชุดคำสั่งได้มากขึ้นในหนึ่งหน่วยเวลาทำได้หลายวิธี เช่น การนำชุดคำสั่งมาหนึ่งครั้งสามารถทำได้หลายคำสั่ง เช่น การทำไปป์ไลน์ (Pipelining) หรือใช้หลักการซุปเปอร์สเกลลาร์ (Superscalar) การทำงานแบบโอเวอร์แลปนี้มีการเฟ็ตช์ชุดคำสั่งที่ตามหลังเข้ามาก่อนที่คำสั่งแรกจะทำงานเสร็จเรียกว่า พรีเฟ็ตช์ (Prefetch) ถ้าหากมีการสร้างไปป์ไลน์หลายระดับ ก็จะทำให้การประมวลผลโดยรวมเร็วขึ้น เนื่องจากโปรเซสเซอร์สามารถรับชุดคำสั่งเข้ามาประมวลผล

ได้หลายๆชุด และคำสั่งที่มีความยาวน้อยกว่าก็จะทำเสร็จก่อนคำสั่งอื่นๆ ที่มีความยาวมากกว่าด้วย จากเทคนิคการปรับปรุงอัตราการประมวลผลคำสั่งที่กล่าวมานี้เมื่อนำกลับมาใช้กับไมโครโปรเซสเซอร์แบบ CISC ที่มีชุดคำสั่งไม่แน่นอนและอ้างแอดเดรสได้หลายลักษณะนั้น จะทำให้การทำงานภายในซับซ้อนมากขึ้น ความซับซ้อนของชุดคำสั่งและการอ้างแอดเดรสจะทำให้ประสิทธิภาพในการทำงานของ CISC ลดลง จึงได้มีการเสนอแนวคิดในการสร้างไมโครโปรเซสเซอร์อีกรูปแบบหนึ่ง คือ แบบ RISC (Reduced Instruction Set Computer) ซึ่งขนาดคำสั่งแต่ละคำสั่งจะมีขนาดเท่ากันทำให้การทำงานแบบไปป์ไลน์หรือซุปเปอร์สเกลลาร์สามารถช่วยเพิ่มประสิทธิภาพในการทำงานของไมโครโปรเซสเซอร์ได้ดีขึ้น

RISC เป็นรูปแบบคำสั่งที่ทุกคำสั่งของไมโครโปรเซสเซอร์ตัวนั้นมีขนาดความยาวของคำสั่งคงที่หรือเท่ากันทุกคำสั่ง (Fixed Instruction Length) โดยปกติแล้วจะออกแบบชุดคำสั่งให้มีความยาวเท่ากับขนาดของบัสข้อมูล เช่น ระบบที่บัสแบบ 32 บิต ก็จะออกแบบคำสั่งให้มีความยาว 32 บิตด้วย คำสั่งที่ออกแบบขึ้นจะเป็นคำสั่งที่ทำงานง่ายๆ แต่ละคำสั่งจะทำงานเพียงขั้นตอนเดียวและต้องทำงานให้เสร็จภายในสัญญาณนาฬิกา 1 ลูก (One Instruction per Cycle) แต่จะยกเว้นสำหรับคำสั่งบางคำสั่ง เช่น คำสั่งหารเลข

เพื่อให้การทำงานของคำสั่งไม่ซับซ้อน ไมโครโปรเซสเซอร์ที่มีคำสั่งแบบ RISC นั้น จะมีโหมดการอ้างแอดเดรสไม่มากนัก และเป็นการอ้างแอดเดรสแบบง่ายๆ โดยใช้รีจิสเตอร์เป็นหลัก สำหรับการติดต่อกับหน่วยความจำภายนอกเพื่อดึงข้อมูลหรือเก็บข้อมูลจะใช้คำสั่งโหลด Load และ Store ในการโหลดและเก็บข้อมูลไว้ในรีจิสเตอร์โดยตรงเมื่อรีจิสเตอร์ทำการประมวลผลแล้วจึงจะนำค่านั้นไปเก็บในหน่วยความจำ

ถ้าหากเปรียบเทียบกันระหว่าง CISC กับ RISC ในเรื่องของขนาดคำสั่งจะพบว่าคำสั่งบางคำสั่งที่ทำงานไม่ซับซ้อนมากนัก คือ คำสั่งแบบ CISC อาจออกแบบให้รหัสคำสั่งมีจำนวนบิตไม่มากได้ โดยออกแบบให้เท่ากับโอเปอร์แรนด์(Operant) ที่ต้องการใช้งานจริง ส่วนคำสั่งแบบ RISC รหัสคำสั่งจะต้องมีความยาวตามที่กำหนดไว้เท่านั้น แม้ว่าคำสั่งนั้นจะทำงานง่ายๆ ก็ตาม เนื่องจากคำสั่งของไมโครโปรเซสเซอร์แบบ RISC มีขนาดคงที่ทำให้ไมโครโปรเซสเซอร์มีคำสั่งไม่กี่คำสั่ง มีคำสั่งได้จำนวนจำกัดและต้องมีรีจิสเตอร์สำหรับใช้งานเป็นจำนวนมากส่วนแบบ CISC สามารถเพิ่มจำนวนคำสั่งมากขึ้นได้ ในปัจจุบันเทคโนโลยีแบบ RISC ได้พัฒนาล้ำหน้าแบบ CISC ไปแล้วตั้งแต่ปี ค.ศ. 1995 แทบจะไม่มีไมโครโปรเซสเซอร์แบบ CISC รุ่นใหม่ๆ เข้าสู่ตลาดเลย

หากเปรียบเทียบสถาปัตยกรรมทั้ง 2 แบบ ในด้านการเขียนโปรแกรมหรือการสนับสนุนของคอมไพเลอร์จะพบว่าคำสั่งแบบ CISC มีชุดคำสั่งจำนวนมาก มีคำสั่งของ

การทำงานที่ซับซ้อนติดมากับไมโครโปรเซสเซอร์อยู่แล้ว ทำให้การเขียนโปรแกรมทำได้ง่าย เนื่องจากการทำงานที่ยุ่งยากสามารถทำงานได้ภายในคำสั่งเดียว ทำให้โปรแกรมที่เขียนขึ้นกับไมโครโปรเซสเซอร์แบบ CISC เป็นโปรแกรมที่มีขนาดเล็กใช้หน่วยความจำไม่มากนัก แต่เนื่องจากปัจจุบันหน่วยความจำของไมโครคอมพิวเตอร์มีราคาถูกลง ดังนั้น ข้อได้เปรียบนี้จึงไม่ใช่ประเด็นสำคัญ แต่การที่โปรแกรมที่มีขนาดเล็กนั้นจะทำให้ปริมาณคำสั่งที่จะต้องอ่านเข้ามาในซีพียูมีจำนวนน้อยลงทำให้ระบบเสียเวลาในการอ่านรหัสคำสั่งน้อยลงไปด้วย

3.6 บทสรุป

โครงสร้างคอมพิวเตอร์แบ่งได้เป็น 6 ชั้น คือ ชั้นที่ 0 จะเป็นวงจรพื้นฐานต่างๆ ของเครื่องที่เรียกว่า Digital Logic Level ชั้นที่ 1 เรียกว่า Micro Architecture Level ประกอบไปด้วยรีจิสเตอร์ต่างๆ รวมกันเป็นหน่วยความจำภายใน ชั้นที่ 2 เรียกว่า Instruction Set Architecture Level หรือ ISA Level สามารถสั่งงานให้ระบบทำงานต่างๆ จะสั่งงานผ่านคำสั่งของเครื่อง ชั้นที่ 3 เรียกว่า Operating System Machine Level เป็นชั้นของระบบปฏิบัติการซึ่งจะเชื่อมโยงระหว่างโปรแกรมต่างๆ กับอุปกรณ์ทางฮาร์ดแวร์ ชั้นที่ 4 เรียกว่า Assembly Language Level เป็นชั้นของภาษาเครื่อง ชั้นที่ 5 เรียกว่า Problem Oriented Language Level เป็นชั้นที่โปรแกรมเมอร์สามารถเขียนโปรแกรมควบคุมด้วยภาษาระดับสูงได้

โครงสร้างของระบบปฏิบัติการมีปัจจัยสำคัญในการดำเนินงาน ได้แก่ 1) สภาพแวดล้อมของระบบปฏิบัติการ 2) ระบบดำเนินการแบบทันที 3) สถาปัตยกรรมของระบบปฏิบัติการและ 4) บริการของระบบปฏิบัติการและโครงสร้างพื้นฐานภายในไมโครโปรเซสเซอร์จะประกอบด้วยส่วนประกอบหลัก คือ หน่วยประมวลผลทางคณิตศาสตร์และลอจิก หน่วยควบคุมและรีจิสเตอร์

สถาปัตยกรรมของระบบปฏิบัติการที่สำคัญ คือ สถาปัตยกรรม Monolithic Architecture สถาปัตยกรรม Layered Architecture สถาปัตยกรรม Microkernel Architecture

โครงสร้างพื้นฐานของไมโครโปรเซสเซอร์ว่าประกอบด้วยส่วนประกอบหลัก คือ หน่วยประมวลผลทางคณิตศาสตร์และลอจิก (ALU: Arithmetic and Logic Unit) หน่วยควบคุม (Control Unit) และรีจิสเตอร์ (Register) ซึ่งหน่วยประมวลผลทางคณิตศาสตร์และลอจิก (ALU) ถือว่าเป็นหัวใจหลักของไมโครโปรเซสเซอร์

สถาปัตยกรรมแบบ CISC และ RISC แบ่งได้เป็น 2 กลุ่ม คือ สถาปัตยกรรมคำสั่งแบบ CISC (Complex Instruction Set Computer) และสถาปัตยกรรมคำสั่งแบบ RISC (Reduced

Instruction Set Computer) ไมโครโปรเซสเซอร์ของอินเทลในตระกูล Intel X86 เป็นตัวอย่างของไมโครโปรเซสเซอร์ที่ใช้สถาปัตยกรรมแบบ CISC สถาปัตยกรรมแบบนี้จะออกแบบให้ไมโครโปรเซสเซอร์มีชุดคำสั่งที่สามารถนำมาเขียนโปรแกรมได้ง่าย คำสั่งต่างๆ จะมีประสิทธิภาพในการทำงานสูงเพื่อให้นักโปรแกรมสามารถเขียนโปรแกรมได้ง่ายที่สุด ในส่วนสถาปัตยกรรมคำสั่งแบบ RISC เป็นวิธีการปรับปรุงอัตราการประมวลผลเพื่อให้ไมโครโปรเซสเซอร์สามารถทำชุดคำสั่งได้มากขึ้นในหนึ่งหน่วยเวลาทำได้หลายวิธี เช่น การนำชุดคำสั่งมาหนึ่งครั้งสามารถทำได้หลายคำสั่ง เช่น การทำไปป์ไลน์ (Pipelining) เพื่อขยายประสิทธิภาพของคอมพิวเตอร์หรือใช้หลักการซุปเปอร์สเกลลาร์ (Super Scalar) เพื่อประมวลผลหลายคำสั่งพร้อมกันได้ในเวลาเดียวกัน ถ้าหากเปรียบเทียบกันระหว่าง CISC กับ RISC ในเรื่องของขนาดคำสั่ง ซึ่งปัจจุบันเทคโนโลยีแบบ RISC ได้พัฒนาล้ำหน้าแบบ CISC ไปแล้วตั้งแต่ปี ค.ศ. 1995 แทบจะไม่มีไมโครโปรเซสเซอร์แบบ CISC รุ่นใหม่ๆ เข้าสู่ตลาดเลย

สถาปัตยกรรมของระบบปฏิบัติการ ได้แก่ สถาปัตยกรรม Monolithic Architecture สถาปัตยกรรม Layered Architectureและสถาปัตยกรรม Microkernel Architecture ซึ่งสถาปัตยกรรม Microkernel ถูกนำไปใช้กับระบบปฏิบัติการต่างๆ ได้แก่ 1) Windows 2000 2) Windows XP 3) Mac OS X

3.7 แบบฝึกหัด

ตอนที่ 1 ให้ผู้เรียนตอบคำถามต่อไปนี้ด้วยตัวเอง

1. โครงสร้างคอมพิวเตอร์ ประกอบด้วยอะไรบ้าง พร้อมอธิบาย
2. โครงสร้างพื้นฐานภายในไมโครโปรเซสเซอร์หน่วยประมวลผลทางคณิตศาสตร์และลอจิกเรียกว่าอะไร และมีการทำงานอย่างไร
3. อธิบายเกี่ยวกับสภาพแวดล้อมของระบบปฏิบัติการ
4. อธิบายระบบดำเนินการแบบทันที
5. อธิบายสถาปัตยกรรมของระบบปฏิบัติการ
6. อธิบายบริการของระบบปฏิบัติการ
7. บอกความแตกต่างสถาปัตยกรรมแบบ CISC และสถาปัตยกรรมแบบ RISC
8. สถาปัตยกรรมพบในระบบปฏิบัติการชนิดใดบ้าง Microkernel Architecture

ตอนที่ 2 ให้ผู้เรียนแบ่งกลุ่มๆ ละ 3 คน เพื่อศึกษาหัวข้อดังต่อไปนี้แล้วจัดทำรายงานส่งในสัปดาห์ถัดไป

1. โครงสร้างคอมพิวเตอร์
2. สภาพแวดล้อมของระบบปฏิบัติการ
3. ระบบดำเนินการแบบทันที
4. บริการของระบบปฏิบัติการ

ตอนที่ 3 ให้ผู้เรียนสะท้อนคิดถึงสิ่งที่ได้เรียนรู้ในบทที่ 3 แล้วนำมาเขียนในลักษณะความเรียงจัดเก็บไว้ในแฟ้มสะสมผลงานเพื่อส่งผู้สอนตอนสิ้นสุดกิจกรรมการเรียนการสอนในภาคเรียนนี้