

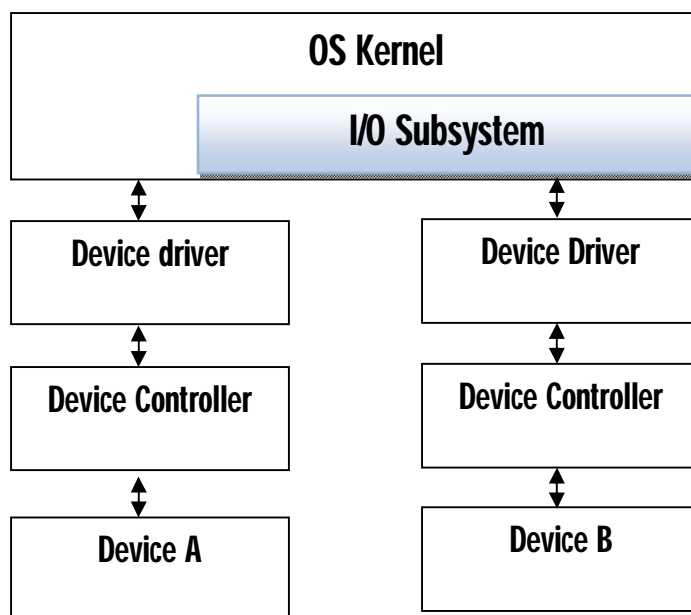
บทที่ 4

บทบาทหน้าที่ระบบปฏิบัติการในการจัดการระบบอินพุต/เอาต์พุต

ระบบปฏิบัติการเป็นตัวประสานการทำงานในระบบคอมพิวเตอร์ ซึ่งภาระงานหนึ่งที่มีความสำคัญต่อการทำงานของระบบคอมพิวเตอร์ คือ จัดการทรัพยากรต่างๆ ภายในระบบ ได้แก่ การจัดการระบบอินพุต/เอาต์พุต (Input / Output) การจัดการไฟล์ (File Management) การจัดการหน่วยความจำ (Memory Management) การจัดการโปรเซส (Process Management) และการจัดการระบบเครือข่าย (Network Management) จะช่วยให้ผู้ใช้สามารถใช้งานคอมพิวเตอร์ผ่านทางฮาร์ดแวร์ร่วมกับผู้ใช้คนอื่นๆ ได้ ในบทเรียนนี้จะกล่าวถึงบทบาทหน้าที่ของระบบปฏิบัติการในการจัดการระบบอินพุต/เอาต์พุตเพื่อการรับ-ส่ง ข้อมูลเข้าออกระหว่างคอมพิวเตอร์กับผู้ใช้ ได้แก่ 1) ความหมายของการจัดการระบบอินพุต/เอาต์พุต 2) ความสำคัญของการจัดการระบบอินพุต/เอาต์พุต 3) ประเภทของอุปกรณ์ในการจัดการระบบอินพุต/เอาต์พุต 4) การจัดการระบบอินพุต/เอาต์พุต และ 5) การจัดการซอฟต์แวร์ในระบบอินพุต/เอาต์พุต 6) ส่วนต่อประสานในระบบอินพุต/เอาต์พุต

4.1 ความหมายของการจัดการระบบอินพุต/เอาต์พุต

พิรพร หมุนสนิทและคณะ (2553: 200) ได้กล่าวถึงแนวทางหนึ่งที่น่าสนใจในการออกแบบระบบปฏิบัติการเพื่อควบคุมการทำงานของอุปกรณ์ คือ การออกแบบระบบปฏิบัติการให้เป็นอิสระต่อประเภทและรูปแบบของอุปกรณ์ (Device Independence) ซึ่งจะมองอุปกรณ์ต่างๆ เป็นเพียงช่องทางสำหรับการรับ-ส่งข้อมูลเท่านั้น โดยไม่ต้องคำนึงถึงคุณลักษณะพิเศษของอุปกรณ์นั้นๆ จึงทำให้ระบบปฏิบัติการสามารถติดต่อกับอุปกรณ์ทุกประเภทได้อย่างอิสระและมีประสิทธิภาพ ทั้งนี้ระบบปฏิบัติการจะต้องมีระบบงานสำหรับควบคุมและจัดการการใช้งานอุปกรณ์ต่างๆ อย่างเป็นระบบเรียกระบบงานย่อยของการจัดการภายในเคอร์เนล (Kernel) ที่เกี่ยวข้องกับอุปกรณ์อินพุต/เอาต์พุตว่า "ระบบอินพุต/เอาต์พุต" ซึ่งเป็นระบบงานที่ทำหน้าที่ส่งงาน ควบคุมและตรวจสอบสถานะของอุปกรณ์อินพุต/เอาต์พุต ทั้งหมดเพื่อให้ส่วนงานอื่นๆ ของระบบปฏิบัติการสามารถเรียกใช้ อุปกรณ์ต่างๆ ได้อย่างสะดวก ซึ่งแสดงโครงสร้างพื้นฐานของระบบอินพุต/เอาต์พุต ได้ดังภาพที่ 4.1



ภาพที่ 4.1 แสดงโครงสร้างพื้นฐานของระบบอินพุต/เอาต์พุต
ที่มา : พิศพร หมุนสนิทและคณะ(2553: 201)

จากภาพที่ 4.1 จะเห็นได้ว่าอุปกรณ์ต่างๆ สามารถติดต่อกับระบบปฏิบัติการได้อย่างอิสระผ่านทางส่วนควบคุมลักษณะพิเศษของอุปกรณ์ ซึ่งถูกออกแบบมาเพื่อติดต่อสื่อสารกับอุปกรณ์ต่างๆ แบ่งออกเป็น 2 ส่วน คือ 1) ตัวควบคุมอุปกรณ์ (Device Controller) หรือ Adapter ซึ่งทำหน้าที่รับ-ส่ง ข้อมูลระหว่างคอมพิวเตอร์กับตัวอุปกรณ์และ 2) ตัวขับอุปกรณ์ (Device Driver) ซึ่งเป็นซอฟต์แวร์เฉพาะที่ทำหน้าที่ควบคุมการติดต่อกับอุปกรณ์ โดยอุปกรณ์แต่ละตัวจะมีตัวควบคุมอุปกรณ์และตัวขับอุปกรณ์เป็นของตนเองซึ่งแตกต่างจากอุปกรณ์ตัวอื่น เนื่องจากการควบคุมการทำงานของอุปกรณ์แต่ละตัวไม่เหมือนกัน แม้ว่าจะเป็นอุปกรณ์ประเภทเดียวกัน แต่ถ้าผลิตจากผู้ผลิตต่างกันก็อาจมีการควบคุมการติดต่อที่ต่างกันได้ เช่น เครื่องพิมพ์ของ EPSON อาจมีการควบคุมต่างจากเครื่องพิมพ์ของ HP การออกแบบระบบปฏิบัติการให้มีการแยกส่วนควบคุมลักษณะพิเศษของอุปกรณ์ออกจากระบบปฏิบัติการมีข้อดีคือ 1) ระบบปฏิบัติการมีขนาดเล็กลง เนื่องจากไม่ต้องนำส่วนควบคุมของอุปกรณ์ต่างๆ ทั้งหมดมารวมไว้ในระบบปฏิบัติการ 2) การเพิ่มอุปกรณ์ใหม่เข้ามาในระบบคอมพิวเตอร์สามารถทำได้ง่ายเนื่องจากระบบปฏิบัติการไม่จำเป็นต้องทราบลักษณะการทำงานของอุปกรณ์ แต่ปล่อยให้ทำหน้าที่ของตัวขับอุปกรณ์นั้นๆ แทนได้

วิกิพีเดีย (2556: ออนไลน์) กล่าวถึง ระบบอินพุต/เอาต์พุต ย่อว่า ไอ/โอ (อังกฤษ: Input/Output: I/O) หรือภาษาไทยว่า รับเข้า/ส่งออก ในทางคอมพิวเตอร์ หมายถึง การสื่อสารระหว่างระบบประมวลผลสารสนเทศ (เช่นคอมพิวเตอร์) กับโลกภายนอก ซึ่งอาจเป็นมนุษย์หรือระบบประมวลผลสารสนเทศอีกระบบหนึ่ง อินพุตหรือสิ่งรับเข้าคือสัญญาณหรือข้อมูลที่ระบบรับเข้ามาและเอาต์พุตหรือสิ่งส่งออกคือสัญญาณหรือข้อมูลที่ระบบส่งออกไป ศัพท์นี้ใช้เรียกการกระทำเพียงส่วนหนึ่ง กล่าวคือ "การกระทำไอ/โอ" หมายถึง การปฏิบัติการรับเข้าหรือส่งออกสัญญาณหรือข้อมูลบุคคลหนึ่ง (หรือระบบอื่น) สามารถใช้อุปกรณ์ไอ/โอเพื่อสื่อสารกับคอมพิวเตอร์ ตัวอย่างเช่น คีย์บอร์ดหรือเมาส์จัดว่าเป็นอุปกรณ์รับเข้าสำหรับคอมพิวเตอร์ ในขณะที่จอภาพและเครื่องพิมพ์จัดว่าเป็นอุปกรณ์ส่งออกสำหรับคอมพิวเตอร์ ส่วนอุปกรณ์ที่สื่อสารระหว่างคอมพิวเตอร์ด้วยกัน เช่น โมเด็มหรือแผ่นวงจรเครือข่ายโดยปกติสามารถเป็นได้ทั้งอุปกรณ์รับเข้าและส่งออก

สรุปได้ว่า ระบบอินพุต / เอาต์พุต หมายถึง ระบบการทำงานเพื่อการรับ-ส่งข้อมูลภายในเครื่องคอมพิวเตอร์กับผู้ใช้ โดยมีระบบปฏิบัติการเข้ามาควบคุมการดำเนินงานของอุปกรณ์ต่างๆ โดยสามารถแยกเป็นอุปกรณ์รับข้อมูล ได้แก่ แป้นพิมพ์ (Keyboard) เมาส์ (Mouse) จอสัมผัส (Touch Screen) ไมโครโฟน (Microphone) อุปกรณ์บันทึกข้อมูล ได้แก่ ดิสก์ (Disk) เทป (Tape) ซีดีรอม (CD-Rom) และอุปกรณ์แสดงผลข้อมูล ได้แก่ จอภาพ (Monitor) ลำโพง (Speaker) เครื่องพิมพ์ (Printer) การติดตั้งระบบอินพุต / เอาต์พุตจะต้องผ่านการควบคุมการทำงานของระบบปฏิบัติการและมีการลงซอฟต์แวร์ผ่านระบบปฏิบัติการจึงจะสามารถใช้งานได้

4.2 ความสำคัญของการจัดการระบบอินพุต/เอาต์พุต

อุปกรณ์อินพุต/เอาต์พุตที่ต่อพ่วงในเครื่องคอมพิวเตอร์นั้น จะมีด้วยกันหลายอย่าง เมื่อผู้ใช้จัดหาซื้ออุปกรณ์คอมพิวเตอร์และนำมาต่อพ่วงกับเครื่องคอมพิวเตอร์ ในช่วงแรกนั้นจะยังไม่สามารถใช้งานอุปกรณ์นั้นได้ เช่น ถ้าผู้ใช้ต้องการพิมพ์งานออกจากเครื่องพิมพ์แล้วนำเครื่องพิมพ์มาต่อพ่วงในช่วงแรกก็ยังไม่สามารถพิมพ์งานออกมาได้เพราะคอมพิวเตอร์ไม่รู้จักรับกับเครื่องพิมพ์ที่ต่อพ่วง โดยจะต้องทำให้คอมพิวเตอร์รู้จักกับเครื่องพิมพ์ที่นำมาต่อพ่วงก่อน ซึ่งจะต้องผ่านกระบวนการที่เรียกว่าการจัดการระบบอินพุต/เอาต์พุต ในการทำงานของระบบคอมพิวเตอร์ตัวกลางที่จะทำหน้าที่ประสานและควบคุมการทำงานระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์อินพุต/เอาต์พุต คือ ระบบปฏิบัติการ ซึ่งระบบปฏิบัติการที่นิยมนำมาใช้งาน ได้แก่

ระบบปฏิบัติการ **Windows** ระบบปฏิบัติการ **Linux** ระบบปฏิบัติการ **Unix** และระบบปฏิบัติการ **Macintosh** ซึ่งระบบปฏิบัติการมีความสำคัญกับการจัดการระบบอินพุต/เอาต์พุตเพื่อให้ระบบคอมพิวเตอร์สามารถรับข้อมูลมาทำการประมวลผลและแสดงออกทางเอาต์พุตตามความต้องการของผู้ใช้ได้

4.3 ประเภทของอุปกรณ์ในการจัดการระบบอินพุต/เอาต์พุต

อุปกรณ์ต่อพ่วงที่ใช้ในงานในระบบคอมพิวเตอร์โดยทั่วไปสามารถแบ่งได้เป็น 4 ประเภท คือ อุปกรณ์ที่ใช้งานเฉพาะทาง (**Dedicated Device**) อุปกรณ์ที่ใช้งานร่วมกัน (**Shared Device**) อุปกรณ์เสมือน (**Virtual Device**) และอุปกรณ์ต่อพ่วงต่างๆ ที่ต่อผ่านพอร์ต **USB** โดยอุปกรณ์แต่ละประเภทมีคุณลักษณะเฉพาะและการทำงานที่ต่างกัน ซึ่งผู้สอนได้ศึกษาเกี่ยวกับอุปกรณ์ในการจัดการระบบอินพุต/เอาต์พุตจากนักวิชาการดังนี้ 1) พีระพันธ์ โสพิศ สติตย์ (2548: 88-92) 2) สัลยุทธ์ สว่างวรรณ (2548: 18-19) 3) พิรพร หมุนสนิทและคณะ (2553: 201-202) แล้วนำมาสรุปประเภทของอุปกรณ์ต่างๆ ได้ดังต่อไปนี้

4.3.1 อุปกรณ์ที่ใช้งานเฉพาะทาง (**Dedicated Device**)

เป็นอุปกรณ์ที่ถูกกำหนดให้ทำงานได้เพียง 1 งาน ในช่วงเวลาหนึ่งเท่านั้น ตัวอย่างเช่น เครื่องพิมพ์(**Printer**) และพล็อตเตอร์ (**Plotter**) เนื่องจากเป็นอุปกรณ์ที่มีความยุ่งยากหากจะกำหนดให้หลายโปรเซส (**Process**) ทำงานพร้อมกัน ดังนั้น ข้อเสียของอุปกรณ์ประเภทนี้ คือ ในช่วงของการดำเนินงาน (**Execute**) อุปกรณ์จะถูกจัดสรรให้กับโปรเซสผู้ใช้เพียง 1 โปรเซสเท่านั้น ซึ่งเป็นการใช้งานอุปกรณ์ได้ไม่เต็มประสิทธิภาพ

4.3.2 อุปกรณ์ใช้งานร่วมกัน (**Shared Device**)

เป็นอุปกรณ์ที่ถูกกำหนดให้ทำงานได้หลายโปรเซสพร้อมกันในช่วงเวลาหนึ่ง ตัวอย่างเช่น ดิสก์และอุปกรณ์จัดเก็บข้อมูลอื่นๆ แบบเข้าถึงโดยตรง(**DASD: Direct Access Storage Device**) ซึ่งสามารถใช้งานร่วมกันได้หลายโปรเซสในเวลาเดียวกันด้วยวิธีการแทรกหรือสลักร่องขอ โดยระบบปฏิบัติการจะต้องมีกลไกการควบคุมล่วงหน้าเพื่อตัดสินว่าการร้องขอของโปรเซสใดจะถูกดำเนินการก่อนเป็นลำดับแรก

4.3.3 อุปกรณ์เสมือน (Virtual Device)

เป็นการจัดประเภทอุปกรณ์ที่รวมคุณสมบัติระหว่างอุปกรณ์ใช้งานเฉพาะทางและอุปกรณ์ใช้งานร่วมกัน โดยพิจารณาจากการเปลี่ยนแปลงคุณสมบัติในการทำงานจากอุปกรณ์ประเภทเฉพาะทางไปเป็นอุปกรณ์ที่ใช้งานร่วมกัน ซึ่งเป็นการทำอุปกรณ์เสมือน เช่น เครื่องพิมพ์จัดเป็นอุปกรณ์ประเภทเฉพาะทางที่สามารถเปลี่ยนแปลงคุณสมบัติให้เป็นอุปกรณ์ประเภทใช้งานร่วมกันเพื่อให้สามารถใช้งานได้หลายโปรเซสในเวลาเดียวกัน โดยโปรเซสการร้องขอใช้งานเครื่องพิมพ์ทั้งหมดจะถูกส่งไปยังดิสก์และเก็บพักข้อมูลไว้ที่ระบบเก็บพักก่อนจะถูกส่งไปพิมพ์ยังเครื่องพิมพ์ จะเห็นได้ว่าการเปลี่ยนแปลงคุณสมบัติของเครื่องพิมพ์ลักษณะนี้คล้ายกับการสร้างเครื่องพิมพ์เสมือน (Virtual Printer) หลายๆ เครื่อง เพื่อให้สามารถรองรับการทำงานได้หลายโปรเซสในเวลาเดียวกัน ซึ่งเป็นการปรับปรุงทั้งด้านประสิทธิภาพและการใช้งานเพื่อให้เกิดความคล่องตัวในการทำงานของระบบ

4.3.4 อุปกรณ์ต่อพ่วงต่างๆ ที่ต่อผ่านพอร์ตยูเอสบี (USB: Universal serial Bus)

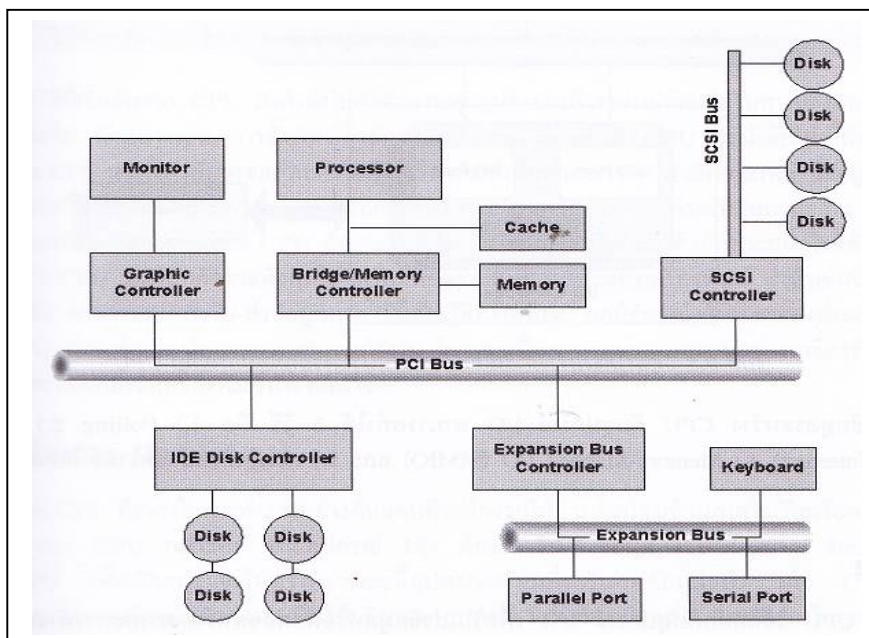
จะมีพอร์ตยูเอสบีควบคุม (USB Controller) เป็นตัวประสานการทำงานระหว่างระบบปฏิบัติการ ไดรฟ์เวอร์ของอุปกรณ์ (Device Driver) และแอปพลิเคชัน ซึ่งอนุญาตให้อุปกรณ์หลายๆ ตัวสามารถแลกเปลี่ยนข้อมูลระหว่างกันภายในระบบคอมพิวเตอร์ได้

4.4 การจัดการระบบอินพุต/เอาพุตในฮาร์ดแวร์

โครงสร้างเชิงฮาร์ดแวร์ในระบบอินพุต/เอาต์พุต ประกอบด้วยอุปกรณ์อินพุต/เอาต์พุตต่างๆ ที่นำมาเชื่อมต่อเพื่อทำงานร่วมกันเป็นจำนวนมาก โดยอุปกรณ์อินพุต/เอาต์พุตแต่ละประเภทมีคุณสมบัติและกลไกการทำงานที่แตกต่างกันออกไป สามารถแบ่งประเภทของอุปกรณ์อินพุต/เอาต์พุตในระบบคอมพิวเตอร์ตามลักษณะการใช้งานได้เป็น 3 กลุ่ม ได้แก่ อุปกรณ์ที่ใช้สำหรับบันทึกข้อมูล (Storage Device) เช่น ดิสก์และเทป อุปกรณ์ที่ช่วยในการส่งผ่านข้อมูล (Transmission Device) เช่น การ์ดเน็ตเวิร์ค (Network Card) และโมเด็ม (Modem) และอุปกรณ์ที่ช่วยในการติดต่อระหว่างผู้ใช้กับเครื่องคอมพิวเตอร์ (Human Interface Device) เช่น จอภาพ แป้นพิมพ์และเมาส์

อุปกรณ์แต่ละประเภทจะมีตัวควบคุมอุปกรณ์เฉพาะซึ่งเป็นแผงวงจรอิเล็กทรอนิกส์ที่ประกอบเข้าเป็นตัวชิป (Chip) เพื่อรวมกันบนแผ่นเมนบอร์ด การติดต่อระหว่างอุปกรณ์กับตัวควบคุมอุปกรณ์จะอยู่ในรูปแบบของภาษาเครื่องและถ่ายโอนข้อมูล ในระดับบิต นอกจากนี้การเชื่อมต่ออุปกรณ์อินพุต/เอาต์พุตต่างๆ เข้ากับระบบคอมพิวเตอร์จำเป็นต้องมี

ส่วนเชื่อมต่อที่เป็นมาตรฐานเพื่อใช้ในการติดต่อสื่อสารระหว่างคอมพิวเตอร์กับอุปกรณ์ ด้วยการส่งสัญญาณไปตามสายผ่านส่วนเชื่อมต่อต่างๆ โดยมีส่วนเชื่อมต่อมาตรฐานที่นิยมใช้งานในคอมพิวเตอร์โดยทั่วไป ได้แก่ 1) พอร์ต (Port) เป็นช่องสื่อสารที่สร้างมาพร้อมกับตัวเครื่องคอมพิวเตอร์ มักเชื่อมต่อกับอุปกรณ์เพียงตัวเดียวเท่านั้น เช่น Serial Port, Parallel Port และ Mouse Port 2) บัส (Bus) เป็นช่องสื่อสารที่ได้รับความนิยมอย่างมากในปัจจุบัน เนื่องจากมีคุณสมบัติที่เรียกว่า “Daisy Chain” ซึ่งเป็นความสามารถในการเชื่อมต่ออุปกรณ์ได้หลายประเภทโดยใช้สายหรือจุดเชื่อมต่อเดียวกัน สามารถแสดงโครงสร้างการเชื่อมต่ออุปกรณ์ในระบบคอมพิวเตอร์ทั่วไปได้ดังภาพที่ 4.2

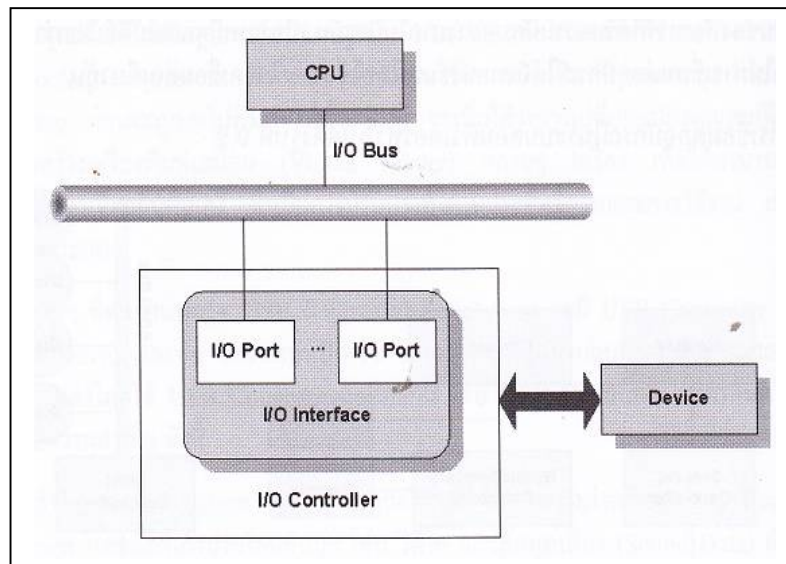


ภาพที่ 4.2 แสดงโครงสร้างการเชื่อมต่ออุปกรณ์ในระบบคอมพิวเตอร์ทั่วไป
ที่มา : พิศพร หมุนสนิทและคณะ (2553: 203)

จากภาพที่ 4.2 จะเห็นได้ว่าอุปกรณ์ต่างๆ ในระบบคอมพิวเตอร์จะเชื่อมต่อกันผ่านทางบัสที่เรียกว่า “PCI Bus” ซึ่งเชื่อมต่อกับอุปกรณ์ต่างๆ ได้แก่ 1) จอภาพ (Monitor) 2) หน่วยประมวลผล (Processor) 3) หน่วยความจำ (Memory) และ 4) อุปกรณ์อื่นๆ ที่มีความเร็วสูง เช่น ดิสก์ อุปกรณ์แต่ละประเภทจะมีหน่วยควบคุม (Controller) ทำหน้าที่ควบคุมการทำงานของอุปกรณ์นั้นๆ โดยหน่วยประมวลผลจะใช้คำสั่งอินพุต/เอาต์พุต (I/O Instruction) เป็นคำสั่งมาตรฐานเพื่อควบคุมให้อุปกรณ์อินพุต/เอาต์พุตทำงานตามที่ต้องการ นอกจากนี้ยังมีบัสส่วนขยาย (Expansion Bus) ที่เชื่อมต่อกับอุปกรณ์ที่มีความเร็วต่ำ เช่น

แป้นพิมพ์ รวมถึงระบบการเชื่อมต่อแบบ SCSI (SCSI ย่อมาจาก Small Computer System Interface) ซึ่งเป็นอินเตอร์เฟซ (Interface) มาตรฐานในการเชื่อมต่ออุปกรณ์ฮาร์ดแวร์เพื่อใช้ในการแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์ฮาร์ดแวร์กับบัสเชื่อมต่ออุปกรณ์จำพวกดิสก์แบบ SCSI

การติดต่อระหว่างซีพียู (CPU: Central Processing Unit) กับอุปกรณ์อินพุต/เอาต์พุตต่างๆ จะมีระบบปฏิบัติการเป็นตัวประสานการทำงานและกำหนดช่วงเวลาการติดต่อกับอุปกรณ์ให้สอดคล้องและเหมาะสมกับสถานะของอุปกรณ์ในขณะนั้น โดยซีพียูจะติดต่อกับอุปกรณ์เมื่อต้องการส่งคำสั่งและข้อมูลรวมถึงการตรวจสอบสถานะการทำงานของอุปกรณ์ผ่านทางหน่วยควบคุมอุปกรณ์นั้นๆ ซึ่งจะใช้คำสั่งอินพุต/เอาต์พุตในการรับ-ส่งข้อมูลกับหน่วยควบคุมอุปกรณ์ผ่านทางพอร์ตคำสั่งอินพุต/เอาต์พุต แสดงโครงสร้างการเชื่อมต่อของหน่วยควบคุมอุปกรณ์กับซีพียู แสดงได้ดังภาพที่ 4.3



ภาพที่ 4.3 แสดงโครงสร้างการเชื่อมต่อของหน่วยควบคุมอุปกรณ์กับซีพียู
ที่มา : พิศพร หมุนสนิทและคณะ (2553: 204)

วิธีการรับ-ส่งข้อมูลระหว่างซีพียูกับอุปกรณ์อินพุต/เอาต์พุต สามารถทำได้ 5 วิธี คือ 1) Polling 2) Programmed I/O 3) การขัดจังหวะ (Interrupt) 4) Memory Mapped I/O (MMIO) และ 5) การเข้าถึงหน่วยความจำโดยตรง (DMA: Direct Memory Access) ดังที่ พิศพร หมุนสนิทและคณะ (2553: 204-209) ได้แสดงไว้ดังต่อไปนี้

4.4.1 พูลลิ่ง (Polling)

เป็นวิธีการที่ซีพียูใช้ติดต่อกับอุปกรณ์ อินพุต/เอาต์พุตเพื่อรับ-ส่งข้อมูล หรือติดต่อขอทราบสถานะการทำงาน เช่น เมื่อซีพียูส่งสัญญาณออกไปถามสถานะการทำงาน ของอุปกรณ์ตัวหนึ่ง อาจได้รับการตอบกลับดังนี้

4.4.1.1 Command Ready หมายถึง อุปกรณ์พร้อมที่จะรับคำสั่งให้ทำงาน

4.4.1.2 Busy หมายถึง อุปกรณ์ไม่พร้อมที่จะรับคำสั่งใหม่ เนื่องจากกำลังทำงาน

4.4.1.3 Error หมายถึง อุปกรณ์ขัดข้องจึงไม่สามารถรับคำสั่งใดๆ ได้ การรับ-ส่ง ข้อมูลด้วยวิธีนี้อาจทำให้เกิดสภาวะการรอคอยที่เรียกว่า **“Busy-Wait Cycle”** ซึ่งเป็นช่วงเวลาที่เกิดขึ้นเมื่อซีพียูต้องรอผลการทำงานเป็นเวลานาน เช่น การอ่านหรือบันทึก ข้อมูลลงดิสก์ จึงไม่สามารถทำงานอื่นได้และต้องรอจนกว่าช่วงเวลานี้จะสิ้นสุดลง ทำให้ซีพียู ต้องเสียเวลาในการทำงานอย่างอื่น

4.4.2 โปรแกรมอินพุต / เอาต์พุต (Programmed I/O)

เป็นวิธีการรับ-ส่ง ข้อมูลระหว่างซีพียูกับอุปกรณ์ อินพุต / เอาต์พุต ด้วยการให้ซีพียูทำงานตามขั้นตอนในโปรแกรมที่สร้างไว้ โดยโปรแกรมนี้อาจประกอบด้วย ชุดคำสั่งสำหรับควบคุมการทำงาน 4 คำสั่ง ดังตารางที่ 4.1

ตารางที่ 4.1 แสดงชุดคำสั่งสำหรับควบคุมการทำงานของซีพียูและอุปกรณ์ อินพุต / เอาต์พุต

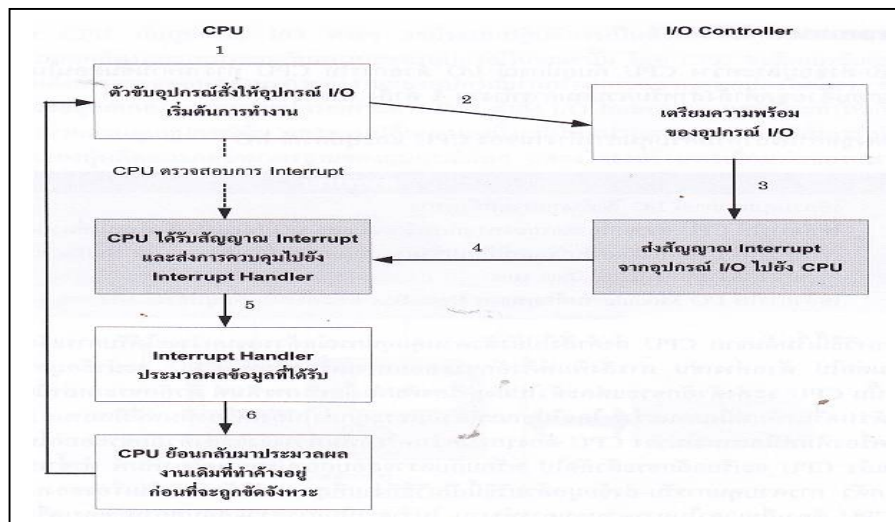
คำสั่ง	คำอธิบาย
Control	ใช้ควบคุมอุปกรณ์อินพุต/เอาต์พุต ให้ทำงานตามที่ต้องการ
Test	ใช้สั่งการให้ซีพียูตรวจสอบสถานะของอุปกรณ์อินพุต /เอาต์พุตและอินพุต /เอาต์พุตโมดูลซึ่งทำหน้าที่ควบคุมการย้ายข้อมูล
Read	ใช้สั่งการให้อินพุต / เอาต์พุตโมดูลรับข้อมูลที่ป้อนเข้ามาจากอุปกรณ์ อินพุต /เอาต์พุตมาวางไว้ในบัฟเฟอร์แล้วส่งต่อให้ซีพียูหรือหน่วยความจำผ่านทาง บัสข้อมูล
Write	ใช้สั่งการให้อินพุต / เอาต์พุตโมดูลรับข้อมูลจากบัสข้อมูลและส่งไปให้กับอุปกรณ์ อินพุต / เอาต์พุต

ที่มา : พิรพร หมุนสนิทและคณะ (2553: 205)

การทำงานของวิธีนี้เริ่มต้นจากซีพียูส่งคำสั่งไปยังตัวควบคุมอุปกรณ์แล้วรอจนกว่า จะได้รับการแจ้งสถานะการทำงาน จึงจะดำเนินการอย่างอื่นต่อไป เช่น การส่งพิมพ์ตัวอักขระ ออกทางเครื่องพิมพ์ซีพียูจะนำข้อมูลที่ส่งพิมพ์ไปจัดเก็บลงในบัฟเฟอร์ก่อน จากนั้นซีพียูจะส่ง ตัวอักขระแต่ละตัวไปยังเครื่องพิมพ์เพื่อทำการพิมพ์ ตัวอักขระเหล่านี้จะถูกส่งออกไปก็ต่อเมื่อ ซีพียูตรวจสอบได้ว่าเครื่องพิมพ์มีสถานะว่าง โดยในขณะที่ตัวอักขระถูกส่งไปยังเครื่องพิมพ์ที่มี สถานะว่าง ตัวอักขระนั้นก็จะถูกพิมพ์ทันที แต่ถ้าเครื่องพิมพ์มีสถานะไม่ว่างซีพียูต้องรอจนกว่า เครื่องพิมพ์ว่างจึงจะส่งตัวอักขระออกไปได้ หลังจากที่ตัวอักขระถูกพิมพ์เรียบร้อยแล้ว ซีพียูจะ เรียกอักขระตัวถัดไปพร้อมกับตรวจสอบสถานะของเครื่องพิมพ์ ทำซ้ำเช่นนี้ไปเรื่อยๆ จนกว่า จะพิมพ์อักขระครบทุกตัว การควบคุมการรับ-ส่ง ข้อมูลด้วยวิธีนี้เป็นวิธีที่ง่ายที่สุด แต่มี ข้อจำกัดในเรื่องของประสิทธิภาพในการใช้งานซีพียู เนื่องจากซีพียูต้องเสียเวลาในการควบคุม การทำงาน ไม่ว่าจะเป็นการตรวจสอบสถานะของเครื่องพิมพ์ การส่งข้อมูลไปยังเครื่องพิมพ์ และการรอจนกว่าเครื่องพิมพ์ทำงานเสร็จ

4.4.3 การขัดจังหวะ (Interrupt)

เป็นวิธีที่ช่วยให้ซีพียูมีเวลาในการทำงานอย่างอื่นแทนที่จะต้องรอให้ การถ่ายโอนข้อมูลเสร็จเรียบร้อยแล้ว โดยใช้หลักการขัดจังหวะการทำงานของซีพียู กล่าวคือ เมื่ออุปกรณ์อินพุต / เอาต์พุตต้องการรับ-ส่งข้อมูลก็จะส่งสัญญาณขัดจังหวะ (Interrupt) ไปขัดจังหวะการทำงานของซีพียู เพื่อเป็นการแจ้งให้ทราบว่าขณะนี้อุปกรณ์พร้อมที่จะรับ-ส่ง ข้อมูลแล้ว เมื่อซีพียูได้รับสัญญาณขัดจังหวะก็จะหยุดการทำงานปัจจุบันชั่วคราว เพื่อให้บริการ แก่อุปกรณ์อินพุต / เอาต์พุต ที่ส่งสัญญาณขัดจังหวะ โดยกระบวนการให้บริการจะเริ่มต้น จากโปรแกรมจัดการสัญญาณขัดจังหวะที่เรียกว่า "Interrupt Handler" ซึ่งเป็นโปรแกรมย่อย ที่ถูกสั่งให้ทำงานโดยอัตโนมัติ เมื่อมีการส่ง Interrupt Vector เพื่อจัดการกับสัญญาณขัดจังหวะ แต่ละประเภท โดยจะทำงานร่วมกับตัวชี้สัญญาณขัดจังหวะซึ่งเป็นอาร์เรย์ (Array) ของพอยน์เตอร์ (Pointer) ที่ชี้ไปที่โปรแกรม Interrupt Handler ต่างๆ ดังนั้นเมื่อซีพียูได้รับ สัญญาณขัดจังหวะก็จะพิจารณาว่าสัญญาณขัดจังหวะนั้นมีหมายเลขใด จากนั้นก็จะนำ หมายเลขไปตรวจสอบกับ Interrupt Vector เพื่อจะได้รับทราบตำแหน่งเริ่มต้นของโปรแกรม Interrupt Handler และสามารถข้ามไปทำงานได้อย่างถูกต้อง โดยสัญญาณขัดจังหวะแต่ละ ประเภทจะถูกเก็บในตำแหน่งที่แน่นอนในหน่วยความจำหลัก แสดงวงรอบการขัดจังหวะ ของอุปกรณ์อินพุต/เอาต์พุตได้ดังภาพที่ 4.4



ภาพที่ 4.4 แสดงผังการเชื่อมต่อของหน่วยควบคุมอุปกรณ์กับซีพียู
ที่มา : พิศพร หมุนสนิทและคณะ (2553: 204)

จากภาพที่ 4.4 สามารถสรุปขั้นตอนการขัดจังหวะของอุปกรณ์อินพุต/เอาต์พุตได้ดังนี้

4.4.3.1 ในขั้นแรกซีพียูจะตรวจสอบว่ามีการขัดจังหวะเกิดขึ้นหรือไม่ กรณีที่มีการขัดจังหวะโดยอุปกรณ์อินพุต / เอาต์พุต จึงจะทำงานในระบบอินพุต / เอาต์พุต โดยตัวควบคุมอุปกรณ์จะส่งให้อุปกรณ์อินพุต / เอาต์พุต เริ่มต้นการทำงาน

4.4.3.2 ตัวควบคุมอุปกรณ์จะเตรียมความพร้อมของอุปกรณ์เพื่อควบคุมการทำงานตามคำสั่งที่ได้รับผ่านสัญญาณ Interrupt

4.4.3.3 เตรียมความพร้อมของอุปกรณ์ที่เกี่ยวข้อง หากเป็นการรับข้อมูล (Input) ก็เตรียมอุปกรณ์สำหรับรับข้อมูล หากเป็นการส่งข้อมูล (Output) ก็เตรียมอุปกรณ์ส่งข้อมูลหรือกรณีที่เกิดข้อผิดพลาดในระหว่างการดำเนินงานก็จะส่งสัญญาณขัดจังหวะจากอุปกรณ์อินพุต / เอาต์พุต ไปยังซีพียู เพื่อแจ้งสถานการณ์การทำงานในปัจจุบันที่เกิดขึ้นให้ซีพียูทราบ

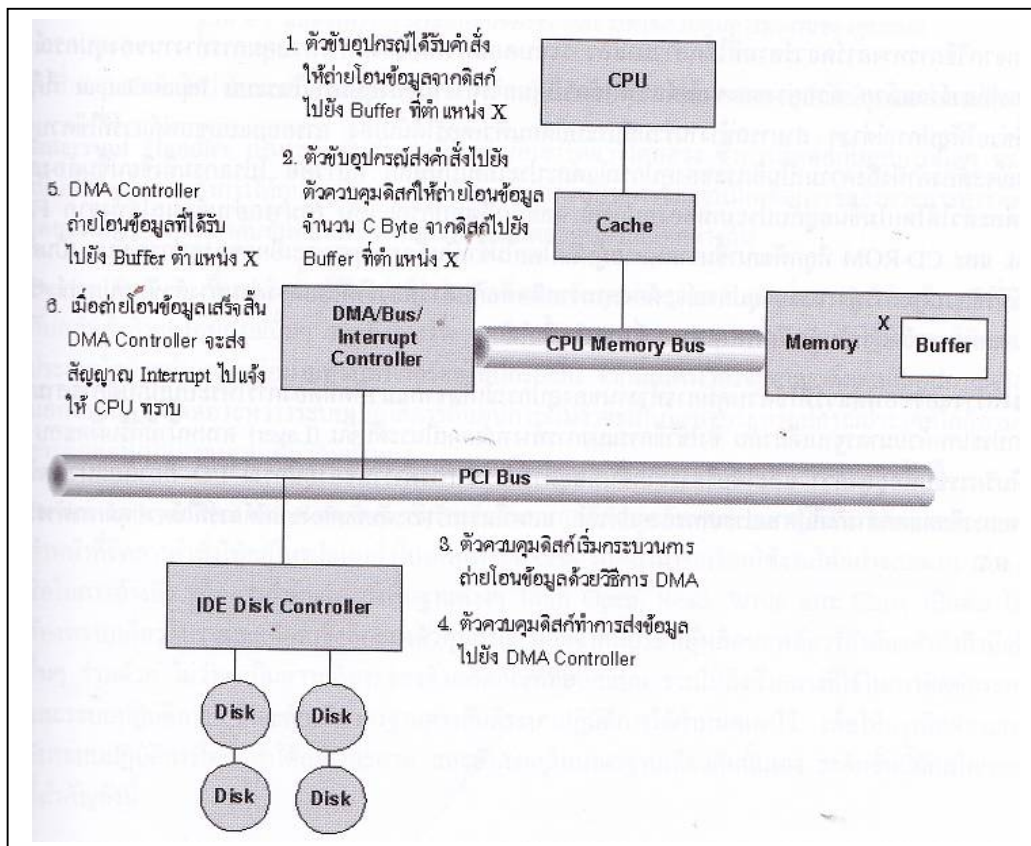
4.4.3.4 เมื่อซีพียูได้รับสัญญาณขัดจังหวะก็จะส่งมอบการควบคุมไปยังโปรแกรม Interrupt Handle เพื่อให้ดูแลการขัดจังหวะที่เกิดขึ้นและโอนการประมวลผลไปยังโปรแกรมนั้น

4.4.3.5 โปรแกรม Interrupt Handle จะทำการประมวลผลข้อมูลที่ได้รับจนเสร็จสิ้น เมื่อการทำงานในขั้นตอนที่ 5 สิ้นสุดลง การควบคุมการทำงานจะกลับคืนสู่ซีพียู โดยซีพียูจะย้อนกลับมาประมวลผลงานเดิมที่ทำค้างอยู่ก่อนที่จะถูกขัดจังหวะ

4.4.3.6 ซีพียูจะประมวลผลต่อไปเรื่อยๆ จนจบ หรือกรณีที่มีการขัดจังหวะโดยอุปกรณ์อินพุต / เอาต์พุต ก็จะทำย้อนกลับไปตรวจสอบการทำงานในขั้นตอนที่ 1 ใหม่ ทำเช่นนี้เรื่อยไปตามลำดับ

4.4.4 การเข้าถึงหน่วยความจำโดยตรง (DMA: Direct Memory Access)

เป็นวิธีการรับ-ส่ง ข้อมูลที่ได้รับการพัฒนาขึ้นมาเพื่อหลีกเลี่ยงปัญหาความล่าช้าที่เกิดขึ้นเมื่อมีการรับ-ส่งข้อมูลปริมาณมากในคราวเดียวกัน การรับ-ส่งข้อมูลด้วยวิธีนี้เป็น การดำเนินการระหว่างหน่วยความจำหลักและอุปกรณ์อินพุต/เอาต์พุต โดยตรงผ่านทางตัวควบคุมการเข้าถึงหน่วยความจำโดยตรง (DMA Controller) ซึ่งไม่จำเป็นต้องผ่านซีพียู จึงทำให้การรับ-ส่งข้อมูลเป็นไปด้วยความรวดเร็ว เนื่องจากซีพียูไม่ต้องเสียเวลารอการทำงานของอุปกรณ์อินพุต/เอาต์พุต ทำให้ซีพียูถูกใช้งานได้อย่างเต็มที่ ตัวควบคุมการเข้าถึงหน่วยความจำโดยตรงจะทำการถ่ายโอนข้อมูลขนาด X Byte จากอุปกรณ์ไปยังหน่วยความจำหลัก โดยการส่งสัญญาณขัดจังหวะไปขัดจังหวะการทำงานของซีพียู เมื่อซีพียูตอบกลับมาขัดจังหวะแล้วตัวควบคุมการเข้าถึงหน่วยความจำโดยตรงจะทำการยึดครองบัสและเริ่มกระบวนการถ่ายโอนข้อมูล โดยในระหว่างนี้ซีพียูสามารถประมวลผลงานอื่นๆ ต่อไปได้ แต่สามารถเรียกใช้ข้อมูลสำหรับการประมวลผลจากแคช (Cache) ได้เท่านั้น ไม่สามารถดึงข้อมูลจากหน่วยความจำสำรองได้ เนื่องจากตัวควบคุมการเข้าถึงหน่วยความจำโดยตรงทำการยึดครองบัสอยู่นั่นเอง เมื่อการถ่ายโอนข้อมูลเสร็จสิ้นตัวควบคุมการเข้าถึงหน่วยความจำโดยตรงจะส่งสัญญาณขัดจังหวะไปแจ้งให้กับซีพียูทราบ แสดงขั้นตอนการทำงานของตัวควบคุมการเข้าถึงหน่วยความจำโดยตรง ได้ดังภาพที่ 4.5



ภาพที่ 4.5 แสดงขั้นตอนการทำงานของตัวควบคุมการเข้าถึงหน่วยความจำโดยตรง
ที่มา : พिरพร หมุนสนิทและคณะ (2553: 209)

จากภาพที่ 4.5 สามารถสรุปขั้นตอนการทำงานของ DMA ได้ดังนี้

4.4.4.1 ตัวขับอุปกรณ์ได้รับคำสั่งจาก ซีพียู ให้ถ่ายโอนข้อมูลจากดิสก์ไปยัง Buffer ในหน่วยความจำหลักที่ตำแหน่ง X

4.4.4.2 ตัวขับอุปกรณ์ส่งคำสั่งไปยังตัวควบคุมดิสก์ให้ทำการถ่ายโอนข้อมูลจำนวน C Byte จากดิสก์ไปยัง Buffer ในหน่วยความจำหลักที่ตำแหน่ง X

4.4.4.3 ตัวควบคุมดิสก์ (Disk Controller) เริ่มกระบวนการถ่ายโอนข้อมูลด้วยวิธีการ DMA

4.4.4.4 ตัวควบคุมดิสก์ทำการส่งข้อมูลไปยัง DMA Controller

4.4.4.5 DMA Controller ถ่ายโอนข้อมูลที่ได้รับไปยัง Buffer

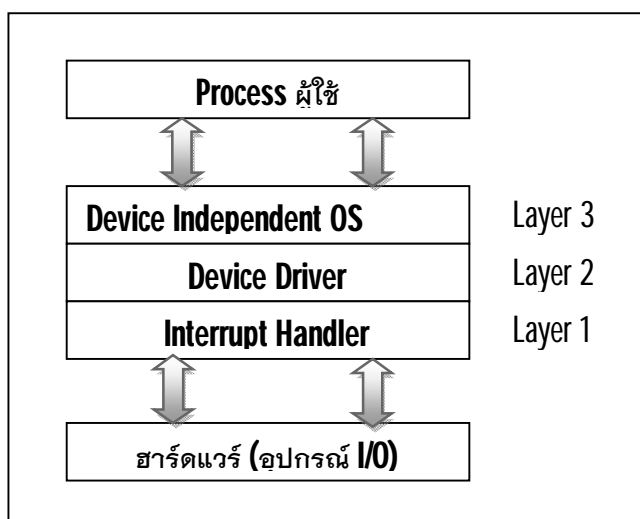
ในหน่วยความจำหลักที่ตำแหน่ง X

4.4.4.6 เมื่อการถ่ายโอนข้อมูลเสร็จสิ้น DMA Controller จะส่งสัญญาณ Interrupt ไปแจ้งให้กับซีพียูทราบ

4.5 การจัดการระบบอินพุต/เอาต์พุตในซอฟต์แวร์

ระบบคอมพิวเตอร์มีการควบคุมการทำงานของอุปกรณ์อินพุต/เอาต์พุตด้วยวิธีการทางซอฟต์แวร์ ตัวอย่างของซอฟต์แวร์ที่ใช้ควบคุมการทำงานของอุปกรณ์ในระบบอินพุต/เอาต์พุตก็คือโปรแกรมไดรฟ์เวอร์ที่ช่วยให้อุปกรณ์ต่างๆ สามารถทำงานร่วมกับระบบคอมพิวเตอร์ได้ การเขียนโปรแกรมเพื่อควบคุมการเข้าถึงอุปกรณ์แต่ละตัวจะต้องไม่ขึ้นอยู่กับประเภทของอุปกรณ์และผู้ผลิตอุปกรณ์ เช่น สามารถอ่านข้อมูลได้ทั้งจาก Floppy Disk, Hard Disk และ CD-ROM ที่ถูกพัฒนาขึ้นจากหลายผู้ผลิตโดยไม่จำเป็นต้องระบุรายละเอียดของอุปกรณ์นั้นๆ โครงสร้างของซอฟต์แวร์ที่ใช้ควบคุมการทำงานของอุปกรณ์เกิดขึ้นจากแนวคิดที่ต้องการให้ระบบปฏิบัติการสามารถติดต่อกับอุปกรณ์ทุกประเภทด้วยมาตรฐานเดียวกัน ซึ่งใช้วิธีการแบ่งการทำงานออกเป็นระดับชั้น (Layer) ตามหน้าที่ที่รับผิดชอบ โดยแต่ละชั้นมีหน้าที่ให้บริการชั้นที่อยู่สูงกว่าและใช้บริการจากชั้นที่อยู่ต่ำกว่า ซึ่งการทำงานของอุปกรณ์อินพุต/เอาต์พุตก็อาศัยแนวคิดนี้เช่นเดียวกัน แต่อาจมีรายละเอียดแตกต่างกันไปตามประเภทของอุปกรณ์ แสดงโครงสร้างระดับชั้นของซอฟต์แวร์ที่ใช้ควบคุมการทำงานของอุปกรณ์ ซึ่งผู้สอนได้ศึกษาเกี่ยวกับการจัดการระบบอินพุต/เอาต์พุตในซอฟต์แวร์จากนักวิชาการดังนี้

- 1) ไพศาล โมลิสกุลมงคล (2545: 226-232)
- 2) พิระพันธ์ โสพัศสถิตย์ (2548: 91-95)
- 3) พิรพร หมุนสนิทและคณะ (2553: 211) แล้วนำมาสรุปประเภทของอุปกรณ์ต่างๆ ได้ดังต่อไปนี้



ภาพที่ 4.6 แสดงโครงสร้างระดับชั้นของซอฟต์แวร์ที่ใช้ควบคุมการทำงานของอุปกรณ์
ที่มา : พิรพร หมุนสนิทและคณะ (2553: 211)

จากภาพที่ 4.6 จะเห็นได้ว่าโครงสร้างระดับชั้นของซอฟต์แวร์ที่ใช้ควบคุมการทำงานของอุปกรณ์ถูกแบ่งออกเป็น 3 ชั้น แสดงนี้

4.5.1 ชั้นการขัดจังหวะ (Interrupt Handler)

เป็นระดับชั้นล่างสุดที่ติดต่อกับฮาร์ดแวร์โดยตรง ซึ่งการติดต่อกับอุปกรณ์ใดๆ จะใช้โปรแกรมเป็นตัวติดต่อและสั่งการให้อุปกรณ์ทำงาน ระดับชั้นนี้ทำหน้าที่ในการจัดการเมื่อเกิดการขัดจังหวะการทำงานของอุปกรณ์โดยจะมีการกำหนดลำดับขั้นตอนการดำเนินงานของแต่ละอุปกรณ์แตกต่างกัน

4.5.2 ชั้นอุปกรณ์ไดรฟ์เวอร์ (Device Driver)

เป็นระดับชั้นที่ทำหน้าที่แปลงคำสั่งและข้อมูลต่างๆ ของระดับชั้นบนสุดให้อยู่ในรูปแบบที่เหมาะสมกับกลุ่มคำสั่งของอุปกรณ์นั้นๆ รวมถึงการจัดการเกี่ยวกับ Buffer ของอุปกรณ์ การติดต่อระหว่างระบบปฏิบัติการกับอุปกรณ์ไม่ว่าจะเป็นไดรฟ์เวอร์ของอุปกรณ์ประเภทใดก็ตามจะมีรูปแบบการติดต่อที่เป็นมาตรฐานเดียวกันทั้งหมด

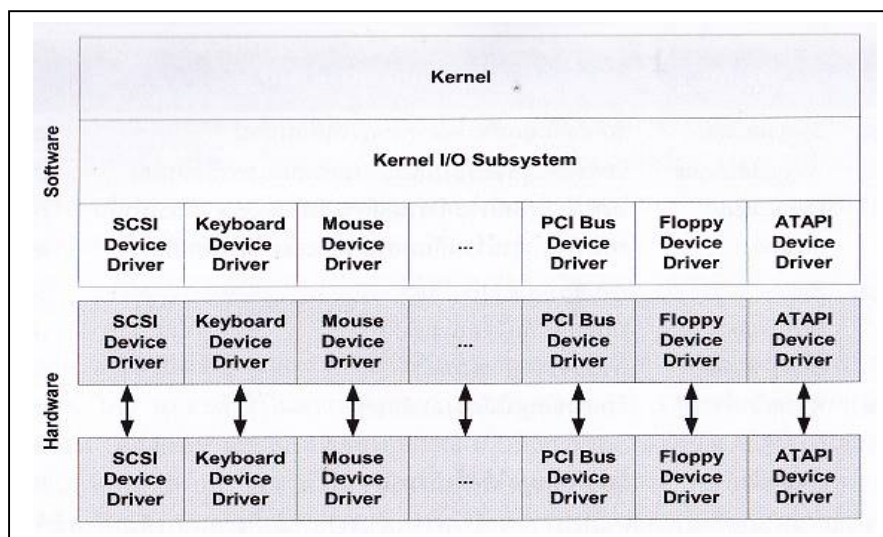
4.5.3 ชั้นอุปกรณ์ซอฟต์แวร์อิสระ (Device Independent OS Software)

เป็นระบบชั้นบนสุดที่ติดต่อกับโปรเซสผู้ใช้ในระดับตรรกะ (Logical) โดยทำหน้าที่จัดการคำสั่งให้อยู่ในรูปแบบทั่วไปเพื่อให้โปรเซสผู้ใช้สามารถเรียกใช้งานได้อย่างสะดวก เช่น การกำหนดชื่อในการอ้างอิงหรือการกำหนดคำสั่งพื้นฐานต่างๆ ได้แก่ Open, Read, Write และ Close โดยไม่จำเป็นต้องทราบเกี่ยวกับรายละเอียดเชิงลึกของตัวอุปกรณ์

4.6 ส่วนต่อประสานในระบบอินพุต / เอาต์พุต

ปัจจุบันอุปกรณ์อินพุต/เอาต์พุตมีความแตกต่างกันในหลายๆ ด้านและมาจากหลายผู้ผลิต ส่งผลให้มาตรฐานการใช้งานอุปกรณ์แต่ละประเภทมีความแตกต่างกันด้วย เรียกมาตรฐานที่ใช้ในการติดต่อระหว่างระบบปฏิบัติการกับอุปกรณ์ว่า "ส่วนต่อประสาน (Interface)" ระบบปฏิบัติการจะมีส่วนต่อประสานมาตรฐานในการใช้งานแต่ละอุปกรณ์ที่ต่างกัน โดยมีระบบย่อยอินพุต/เอาต์พุต (I/O Subsystem) ซึ่งเป็นส่วนหนึ่งของเคอร์เนล (Kernel) ในระบบปฏิบัติการทำหน้าที่ดูแลการติดต่อระหว่างตัวควบคุมอุปกรณ์แต่ละประเภท จะมีไดรฟ์เวอร์เป็นของตนเอง ซึ่งในระดับชั้น Device Driver ของโปรแกรมไดรฟ์เวอร์จะช่วยเหลือรายละเอียดความแตกต่างของตัวควบคุมอุปกรณ์เอาไว้จากโปรแกรมเคอร์เนลของระบบปฏิบัติการ เพื่อให้การติดต่อระหว่างอุปกรณ์ประเภทต่างๆ กับระบบปฏิบัติการใช้มาตรฐานเดียวกัน ซึ่งผู้สอนได้ศึกษาเกี่ยวกับส่วนต่อประสานในระบบอินพุต / เอาต์พุต

จากนักวิชาการดังนี้ 1) ไพศาล โมลิสกุลมงคล (2545: 235-239) 2) สลยุทธ สว่างวรรณ (2548: 157-164) 3) พิรพร หมุนสนิทและคณะ (2553: 212-215) แล้วนำมาสรุปประเภทของ อุปกรณ์ต่างๆ ได้ดังต่อไปนี้



ภาพที่ 4.7 แสดงโครงสร้างอินเตอร์เฟสของเคอร์เนลในระบบ อินพุต/เอาต์พุต
ที่มา : พิรพร หมุนสนิทและคณะ (2553: 213)

จากภาพที่ 4.7 จะเห็นได้ว่าโครงสร้างอินเตอร์เฟสของแต่ละอุปกรณ์มีความแตกต่างกันขึ้นอยู่กับประเภทของอุปกรณ์นั้นๆ การติดต่อระหว่างแต่ละอุปกรณ์กับระบบปฏิบัติการจะต้องมีฮาร์ดแวร์ควบคุมการทำงานของอุปกรณ์นั้นๆ คือ ตัวควบคุมอุปกรณ์ และทำการสื่อสารกับระบบย่อยอินพุต/เอาต์พุต (I/O Subsystem) ผ่านทางซอฟต์แวร์ คือ ไดรฟ์เวอร์ที่ทำหน้าที่จัดการการทำงานของอุปกรณ์นั้นๆ จากความหลากหลายของอุปกรณ์ทำให้ระบบย่อยอินพุต / เอาต์พุต ต้องทราบคุณลักษณะของอุปกรณ์ทุกตัวเป็นอย่างดี จึงจะทำงานร่วมกันได้อย่างราบรื่น ดังนั้นไดรฟ์เวอร์จึงเปรียบเสมือนล่ามที่ช่วยแปลภาษาของแต่ละอุปกรณ์ให้เป็นภาษามาตรฐานที่ระบบย่อยอินพุต/เอาต์พุตสามารถเข้าใจได้ การใช้ไดรฟ์เวอร์ร่วมกับตัวควบคุมอุปกรณ์ที่พัฒนาขึ้นมาใช้งานเฉพาะอุปกรณ์นั้นๆ สามารถควบคุมให้อุปกรณ์ทำงานได้อย่างถูกต้อง โครงสร้างอินเตอร์เฟสลักษณะนี้ช่วยให้การเชื่อมต่อกับอุปกรณ์อินพุต / เอาต์พุต เข้ากับระบบคอมพิวเตอร์ทำได้สะดวก เพียงแค่ติดตั้งโปรแกรมไดรฟ์เวอร์ที่ถูกต้องตามข้อกำหนดของอุปกรณ์ก็สามารถใช้งานได้ทันที โดยไม่ต้องแก้ไขหรือปรับปรุงการทำงานของระบบปฏิบัติการแต่อย่างใด

นอกจากนี้อุปกรณ์อินพุต / เอาต์พุตต่างๆ ยังสามารถแบ่งประเภทตามฟังก์ชันการทำงานของโปรแกรมประยุกต์ โดยแบ่งออกเป็นลักษณะต่างๆ ดังตารางที่ 4.2

ตารางที่ 4.2 แสดงการจัดประเภทอุปกรณ์ตามฟังก์ชันการทำงานของโปรแกรมประยุกต์

ลักษณะงานประยุกต์	ประเภท	ลักษณะการทำงาน	ตัวอย่างอุปกรณ์
รูปแบบการส่งข้อมูล	Character	รับ-ส่งข้อมูลที่ละตัวอักษร	แป้นพิมพ์
	Block	รับ-ส่งข้อมูลเป็นกลุ่ม เช่น 16 Byte หรือ 512 Byte	ดิสก์
การเข้าถึงข้อมูล	Sequential	เข้าถึงข้อมูลแบบเรียงตามลำดับ	โมเด็ม
	Random	สามารถค้นหาหรือย้ายตำแหน่งในการเข้าถึงข้อมูลได้	ซีดีรอม
จังหวะการรับ - ส่งข้อมูล	Synchronous	รับ-ส่งข้อมูลภายใต้กำหนดเวลาที่แน่นอน	เทป
	Asynchronous	รับ-ส่งข้อมูลโดยไม่เป็นไปตามกำหนดเวลาที่แน่นอน	แป้นพิมพ์
การใช้ข้อมูลร่วมกัน	Dedicated	ในช่วงเวลาหนึ่งจะทำงานเฉพาะกับโปรเซสเดียวเท่านั้น	เทป
การใช้ข้อมูลร่วมกัน	Shared	สามารถทำงานร่วมกับหลายโปรเซสได้พร้อมกัน	แป้นพิมพ์
ทิศทางการรับ-ส่งข้อมูล	Read-Only	อ่านข้อมูลได้อย่างเดียว	ซีดีรอม
	Write-Only	บันทึกข้อมูลได้อย่างเดียว	เครื่องพิมพ์
	Read-Write	อ่านและบันทึกข้อมูลได้	ดิสก์
ความเร็วของอุปกรณ์	ความเร็วต่ำ	รับ-ส่งข้อมูลได้ครั้งละน้อย ๆ	แป้นพิมพ์ (มีความเร็ว 10 Byte ต่อวินาที)
	ความเร็วสูง	รับ-ส่งข้อมูลได้ครั้งละมาก ๆ	สแกนเนอร์ (มีความเร็ว 400 KB ต่อวินาที)

ที่มา : พирพร หมุนสนิทและคณะ (2553: 213-214)

จากลักษณะการทำงานต่างๆ ของโปรแกรมประยุกต์ข้างต้น ระบบอินพุต/เอาต์พุต จะทำหน้าที่ควบคุมการรับ-ส่งข้อมูลเพื่อประสานการทำงานอย่างเป็นระบบตามประเภทของ อุปกรณ์ที่ทำการติดต่อ ซึ่งแบ่งอุปกรณ์ออกเป็น 4 ประเภทหลัก คือ

4.6.1 อุปกรณ์ที่รับ-ส่งข้อมูลเป็นกลุ่มและทีละตัวอักษร (Block and Character Device)

อุปกรณ์ที่รับ-ส่งข้อมูลแบ่งออกเป็น ตัวอย่างอุปกรณ์ประเภท **Character** คือ แป้นพิมพ์ การติดต่อกับอุปกรณ์ประเภทนี้เพื่อให้ส่งข้อมูลที่ละตัวอักษรจะใช้คำสั่งของระบบ เช่น **Get()** หรือ **Put()** นอกจากนี้หากต้องการให้ส่งข้อมูลที่ละบรรทัดจะต้องใช้ **Buffer** ร่วมกับการเขียนชุดคำสั่งเพิ่มเติม ส่วนตัวอย่างของอุปกรณ์ประเภท **Block** คือ ดิสก์ การติดต่อกับอุปกรณ์ประเภทนี้จะใช้คำสั่งของระบบ เช่น **Read()** และ **Write()** สำหรับการอ่าน และบันทึกข้อมูลด้วย หากเป็นอุปกรณ์ที่สามารถเข้าถึงข้อมูลได้แบบสุ่ม (**Random**) จะมีการใช้คำสั่ง **Seek()** เพื่อค้นหาข้อมูลร่วมด้วย

4.6.2 อุปกรณ์ที่รับ-ส่ง ข้อมูลภายใต้กำหนดเวลาที่แน่นอน (Asynchronous) Synchronous and Asynchronous Device

เมื่อโปรแกรมประยุกต์เรียกใช้อุปกรณ์ประเภท **Synchronous** เช่น เทป ระบบคอมพิวเตอร์ที่กำลังประมวลผลโปรแกรม ในขณะนั้นจะต้องหยุดการทำงานชั่วคราว เพื่อให้คำสั่งในโปรแกรมประยุกต์ทำงานจนเสร็จก่อน จึงจะสามารถกลับมาประมวลผลโปรแกรมต่อไป สำหรับโปรแกรมประยุกต์ที่เรียกใช้อุปกรณ์ประเภท **Asynchronous** เช่น แป้นพิมพ์ระบบคอมพิวเตอร์ยังสามารถประมวลผลโปรแกรม ในขณะนั้นต่อได้จนกระทั่งเสร็จสิ้นกระบวนการจึงค่อยนำคำสั่งของโปรแกรมมาประมวลผล เทคนิคการเรียกใช้งานอุปกรณ์ประเภท **Asynchronous** คือ การใช้ **Multithreading** กล่าวคือ โปรแกรมหนึ่งจะมีมากกว่า 1 เธรด (**Thread**) โดยกำหนดให้เธรดหนึ่งทำงานหลักคือประมวลผลและแสดงผล ในขณะที่อีก **Thread** หนึ่งทำหน้าที่รับข้อมูล ซึ่งจะช่วยให้การทำงานมีประสิทธิภาพมากยิ่งขึ้น

4.6.3 อุปกรณ์เครือข่าย (Network Device)

เป็นอุปกรณ์สำหรับการติดต่อสื่อสารบนเครือข่าย เช่น การ์ดเน็ตเวิร์ค (**NIC**) การติดต่อกับอุปกรณ์ประเภทนี้จะใช้โปรแกรมที่เรียกว่า **"Socket"** ซึ่งแตกต่างจากการติดต่อกับอุปกรณ์โดยทั่วไปตรงที่ภายในโปรแกรม **Socket** จะมีฟังก์ชัน **Select** ที่ใช้ติดต่อกับ **Remote Address** โดยฟังก์ชันนี้จะเป็นตัวจัดการในการติดต่อประสานงานเพื่อรับ-ส่งข้อมูลต่างๆ เมื่อระบบคอมพิวเตอร์มีการเรียกใช้ฟังก์ชัน **Select** ฟังก์ชันนี้จะส่งค่าสถานะในขณะนั้นกลับไปยังส่วนที่เรียกใช้เพื่อทำการรับ-ส่งข้อมูลตามที่กำหนด

4.6.4 ระบบย่อยอินพุต / เอาต์พุต (I/O Subsystem)

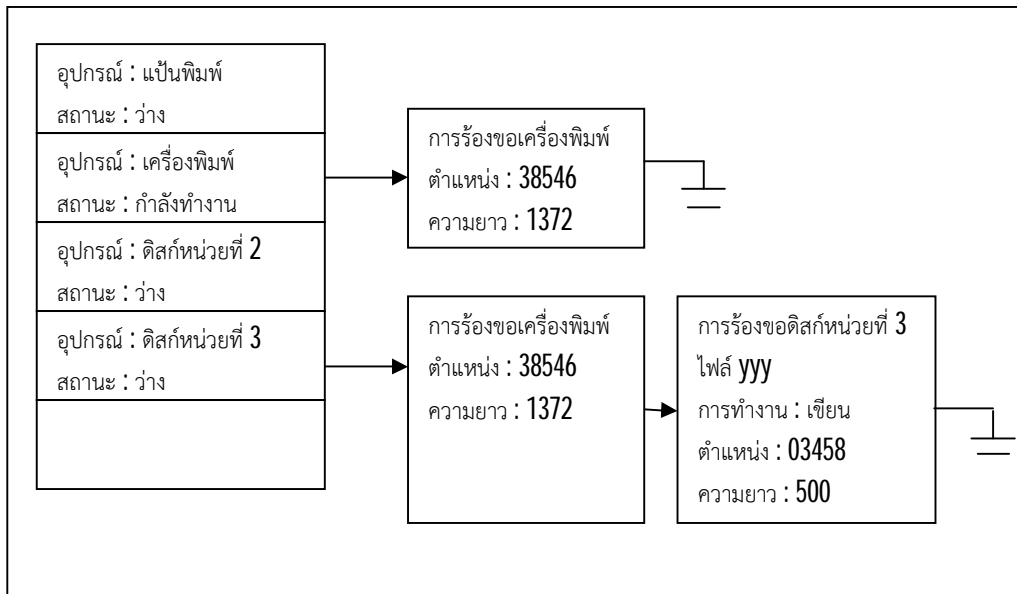
เป็นระบบย่อยของ Kernel ในระบบปฏิบัติการที่ทำหน้าที่ควบคุมการดำเนินงานของอุปกรณ์ต่างๆ ในระบบอินพุต/เอาต์พุตรวมทั้งจัดเตรียมบริการต่างๆ ที่สนับสนุนการทำงานของอุปกรณ์ ได้แก่ การจัดตารางอุปกรณ์ (I/O Scheduling) การใช้งาน Buffer Cache และ Spooling รวมไปถึงการจัดการกับข้อผิดพลาดต่างๆ ที่อาจเกิดขึ้น มีรายละเอียดที่สำคัญของแต่ละบริการดังนี้

4.6.4.1 การจัดตารางอุปกรณ์ (I/O Scheduling)

เป็นบริการที่สำคัญอย่างหนึ่งของ I/O Subsystem ที่ช่วยควบคุมการใช้งานอุปกรณ์ต่างๆ อย่างเป็นระบบ โดยการบันทึกข้อมูลการทำงานของอุปกรณ์อินพุต/เอาต์พุตแต่ละตัวไว้ในตารางที่เรียกว่า "Device-Status Table" ซึ่งประกอบด้วยข้อมูลต่างๆ ได้แก่ ประเภทของอุปกรณ์ ตำแหน่ง (Address) และสถานะของอุปกรณ์ เมื่อมีสัญญาณขัดจังหวะเกิดขึ้น ระบบปฏิบัติการจะตรวจสอบว่าสัญญาณขัดจังหวะนั้นมาจากอุปกรณ์อินพุต/เอาต์พุตตัวใดและชี้ไปยังตารางของอุปกรณ์นั้น เพื่อตรวจสอบสถานะของอุปกรณ์ว่าพร้อมดำเนินการหรือไม่ ถ้าพร้อมดำเนินการก็จะทำการปรับปรุงค่าในตารางให้ถูกต้องตามสัญญาณขัดจังหวะนั้น กรณีที่มีคิวของอุปกรณ์รออยู่ ระบบปฏิบัติการก็จะทำงานตามการร้องขอถัดไป เมื่อดำเนินการเสร็จสิ้นก็จะคืนการควบคุมเพื่อกลับไปประมวลผลงานเดิมก่อนที่จะถูกขัดจังหวะ แสดงตัวอย่างตารางแสดงสถานะการทำงานของอุปกรณ์ได้ดังภาพที่ 4.10

4.6.4.2 การใช้งาน Buffer Cache และ Spooling

ระบบย่อยอินพุต/เอาต์พุต (I/O Subsystem) มีการนำเทคนิคการแก้ปัญหาในเรื่องของความแตกต่างด้านความเร็วในการรับ-ส่งข้อมูลระหว่างซีพียูกับอุปกรณ์อินพุต / เอาต์พุต โดยการใช้งาน Buffer Cache และ Spooling เพื่อช่วยจัดเก็บข้อมูลในขณะที่มีการถ่ายโอนข้อมูลมีรายละเอียดที่สำคัญดังภาพที่ 4.8



ภาพที่ 4.8 แสดงตัวอย่างตารางสถานะของอุปกรณ์
ที่มา : พิศพร หมุนสนิทและคณะ (2553: 215)

1) การใช้งาน **Buffer Buffer** เป็นที่พักข้อมูลชั่วคราวเพื่อส่งข้อมูลจากคอมพิวเตอร์ไปยังอุปกรณ์ หรือส่งข้อมูลจากอุปกรณ์มายังคอมพิวเตอร์ เนื่องจากอัตราความเร็วในการทำงานของแต่ละอุปกรณ์กับซีพียูไม่เท่ากัน คอมพิวเตอร์จึงจำเป็นต้องมีหน่วยความจำสำรองเก็บพักข้อมูลไว้ก่อนชั่วคราวเพื่อให้กระบวนการถ่ายโอนข้อมูลมีความต่อเนื่องและช่วยเพิ่มประสิทธิภาพการทำงานให้กับระบบ ข้อดีของการใช้งาน **Buffer** คือ

- 1) ช่วยให้อุปกรณ์ที่มีความเร็วในการรับ-ส่งข้อมูลสูงกว่าไม่ต้องเสียเวลารอการทำงานของอุปกรณ์ที่มีความเร็วช้ากว่า ตัวอย่างเช่น ไฟล์ข้อมูลที่อ่านจากโมเด็มซึ่งมีความเร็วช้ากว่าดิสก์ ถ้าสามารถอ่านข้อมูลจากโมเด็มมาเก็บไว้ในหน่วยความจำหลักซึ่งใช้เป็น **Buffer** ก่อน แล้วค่อยเขียนลงดิสก์ครั้งเดียวจะช่วยให้การทำงานเร็วยิ่งขึ้น
- 2) ช่วยให้อุปกรณ์ที่มีความเร็วในการรับ-ส่ง ข้อมูลต่างกันสามารถทำงานควบคู่กันได้อย่างมีประสิทธิภาพ และ
- 3) ช่วยในการสำรองข้อมูล (**Back Up**) เพื่อป้องกันการสูญหายของข้อมูล

2) การใช้งาน **Cache** เป็นหน่วยความจำความเร็วสูงที่อยู่ระหว่าง ซีพียู และหน่วยความจำหลัก ซึ่งช่วยเพิ่มความเร็วในการดึงข้อมูลจากหน่วยความจำ โดยการนำเอาข้อมูลบางส่วนมาไว้ใน **Cache** เนื่องจากการอ่านข้อมูลใน **Cache** เร็วกว่า แต่ถ้าไม่มีข้อมูลใน **Cache** ก็จะไปอ่านในหน่วยความจำ การใช้งาน **Cache** มีวัตถุประสงค์เดียวกับการใช้งาน **Buffer** คือ เพื่อเก็บสำรองข้อมูลบางส่วน แต่ **Cache** มีราคาสูงกว่าและทำงานได้เร็วกว่า **Buffer** ดังนั้น งานบางประเภทที่ต้องการความรวดเร็วจึงเหมาะที่จะใช้ **Cache** มากกว่า

3) การใช้งาน Spooling เป็นที่พักข้อมูลชั่วคราวสำหรับจัดเก็บ

ข้อมูลที่จะถูกส่งออกไปยังอุปกรณ์ เพื่อแก้ปัญหาของอุปกรณ์ที่ไม่สามารถสลับการทำงานหรือทำงานพร้อมกันระหว่างโปรเซสได้ เช่น ในระบบคอมพิวเตอร์ที่มีการใช้งานเครื่องพิมพ์ร่วมกัน ในขณะที่สั่งพิมพ์งาน ระบบปฏิบัติการจะใช้ Spooling เข้ามาจัดการกับข้อมูลเพื่อจัดเก็บข้อมูลตามลำดับที่ส่งเข้ามา โดยทำหน้าที่จัดลำดับคิวงานให้กับเครื่องพิมพ์ ขณะทำงานต่างๆ เก็บอยู่ใน Spooling ผู้ใช้สามารถควบคุมการทำงานผ่านทางระบบปฏิบัติการได้ เช่น การสั่งยกเลิกการพิมพ์

4.6.4.3 การจัดการกับข้อผิดพลาด (Error Handling)

เนื่องจากระบบปฏิบัติการทำหน้าที่เป็นตัวกลางในการประสานการทำงานระหว่างอุปกรณ์และโปรแกรมประยุกต์ของผู้ใช้ทำให้ระบบปฏิบัติการมีความสามารถในการจัดการกับข้อผิดพลาดที่อาจเกิดขึ้นกับอุปกรณ์ เพื่อให้ระบบยังคงทำงานต่อไปได้ ตัวอย่างเช่น ข้อผิดพลาดที่เกิดขึ้นกับเครื่องพิมพ์กรณีที่กระดาษหมดหรือการรับ-ส่งข้อมูลในเครือข่ายติดขัด ซึ่งข้อผิดพลาดเพียงเล็กน้อยเช่นนี้ไม่ควรทำให้การทำงานของระบบต้องหยุดชะงักลง ระบบปฏิบัติการโดยทั่วไปจะจัดการกับข้อผิดพลาดเหล่านี้ให้ในระดับหนึ่ง เช่น การแจ้งข้อผิดพลาดให้โปรเซสของผู้ใช้ทราบพร้อมทั้งบอกสาเหตุและแนวทางในการแก้ไข

4.7 บทสรุป

ระบบอินพุต/เอาต์พุต (I/O System) หมายถึง ระบบการทำงานเพื่อการรับ-ส่งข้อมูลภายในเครื่องคอมพิวเตอร์กับผู้ใช้ โดยมีระบบปฏิบัติการเข้ามาควบคุมการดำเนินงานของอุปกรณ์ต่างๆ ซึ่งอุปกรณ์รับข้อมูล ได้แก่ แป้นพิมพ์ (Keyboard) เมาส์ (Mouse) จอสัมผัส (Touch Screen) ไมโครโฟน (Microphone) อุปกรณ์บันทึกข้อมูล ได้แก่ ดิสก์ (Disk) เทป (Tape) ซีดีรอม (CD-Rom) และอุปกรณ์แสดงผลข้อมูล ได้แก่ จอภาพ (Monitor) ลำโพง (Speaker) เครื่องพิมพ์ (Printer) การติดตั้งระบบอินพุต / เอาต์พุตจะต้องผ่านการควบคุมการทำงานของระบบปฏิบัติการและมีการลงซอฟต์แวร์ผ่านระบบปฏิบัติการจึงจะสามารถใช้งานได้

ระบบอินพุต/เอาต์พุต (I/O System) คือ ระบบการทำงานที่สำคัญในคอมพิวเตอร์ที่ครอบคลุมการทำงานตั้งแต่การรับข้อมูลคือคำสั่งจากภายนอกเข้าสู่เครื่องคอมพิวเตอร์ผ่านทางอุปกรณ์รับข้อมูล (Input Device) เพื่อนำข้อมูลมาประมวลผลและแสดงผลลัพธ์ที่ได้ผ่านทางอุปกรณ์แสดงผล (Output Device) โดยกระบวนการเหล่านี้จำเป็นต้องมีระบบปฏิบัติการเข้ามาควบคุมเพื่อให้การดำเนินงานเป็นไปอย่างราบรื่น ระบบการทำงานย่อยของ

ระบบปฏิบัติการที่ทำหน้าที่ควบคุมการทำงานของระบบอินพุต/เอาต์พุตเรียกว่า “ระบบย่อยอินพุต/เอาต์พุต” (I/O Subsystem) ซึ่งเป็นระบบงานที่ทำหน้าที่สั่งงาน ควบคุมและตรวจสอบสถานะของอุปกรณ์อินพุต/เอาต์พุตทั้งหมด เพื่อให้ส่วนงานอื่นๆ ของระบบปฏิบัติการสามารถเรียกใช้อุปกรณ์ต่างๆ ได้อย่างสะดวก ระบบปฏิบัติการที่นิยมใช้งาน ได้แก่ ระบบปฏิบัติการ Windows ระบบปฏิบัติการ Linux ระบบปฏิบัติการ Unix และระบบปฏิบัติการ Macintosh

ประเภทของอุปกรณ์อินพุต/เอาต์พุต แบ่งได้ 4 ประเภท ได้แก่ อุปกรณ์ที่ใช้งานเฉพาะทาง (Dedicated Device) อุปกรณ์เสมือน (Virtual Device) และอุปกรณ์ต่อพ่วงต่างๆ ที่ต่อผ่านพอร์ตยูเอสบี (USB: Universal serial Bus) ซึ่งวิธีการจัดการระบบอินพุต/เอาต์พุตในการรับ-ส่งข้อมูลระหว่างซีพียูกับอุปกรณ์ อินพุต/เอาต์พุต สามารถทำได้ 5 วิธี คือ 1) Polling 2) Programmed I/O 3) Interrupt 4) Memory-Mapped I/O (MMIO) และ 5) DMA โดยแต่ละวิธีจะมีกลไกการทำงานที่แตกต่างกันออกไปขึ้นอยู่กับการทำงานของแต่ละระบบปฏิบัติการ

การจัดการระบบอินพุต / เอาท์พุต ในซอฟต์แวร์จะแบ่งตามลำดับชั้นต่างๆ ชั้นการขัดจังหวะ ชั้นอุปกรณ์ไดรฟ์เวอร์และชั้นอุปกรณ์ซอฟต์แวร์อิสระ

นอกจากนี้อุปกรณ์อินพุต / เอาต์พุต ต่างๆ ยังสามารถแบ่งประเภทตามฟังก์ชันการทำงานของโปรแกรมประยุกต์เพื่อให้การทำงานของอุปกรณ์มีความสอดคล้องกับลักษณะการทำงานของโปรแกรมประยุกต์มากที่สุด โดยแบ่งออกเป็น 4 ลักษณะ คือ 1) Block and Character Device 2) Synchronous and Asynchronous Device 3) Network Device และ 4) I/O System ระบบปฏิบัติการที่ดีควรมีบริการต่างๆ ที่สนับสนุนการทำงานของอุปกรณ์ ได้แก่ การจัดตารางอุปกรณ์ (I/O Scheduling) การใช้งาน Buffer Cache และ Spooling รวมไปถึงการจัดการกับข้อผิดพลาดต่างๆ

4.8 แบบฝึกหัด

ตอนที่ 1 ให้ผู้เรียนตอบคำถามต่อไปนี้ด้วยตัวเอง

1. ระบบปฏิบัติการมีวิธีการจัดการระบบอินพุต / เอาต์พุตอย่างไร
2. โครงสร้างพื้นฐานของระบบอินพุต / เอาต์พุตแตกต่างกันอย่างไร
3. ระบบคอมพิวเตอร์แบ่งอุปกรณ์อินพุต / เอาต์พุตตามลักษณะการใช้งาน

เป็นที่ประเภท อะไรบ้าง

4. ในระบบคอมพิวเตอร์สามารถเชื่อมต่ออุปกรณ์อินพุต / เอาต์พุตเข้าด้วยกันได้

อย่างไร

5. การรับ-ส่งข้อมูลระหว่างซีพียูกับอุปกรณ์อินพุต / เอาต์พุตแบ่งออกเป็นกี่วิธี

อะไรบ้าง

6. ชุดคำสั่งสำหรับควบคุมการทำงานใน Programmed I/O ประกอบด้วยคำสั่ง

ใดบ้าง

7. ขั้นตอนการขัดจังหวะอุปกรณ์อินพุต / เอาต์พุตมีอะไรบ้าง อธิบายพอสังเขป

8. ขั้นตอนการทำงานของ DMA (Direct Memory Access) มีอะไรบ้าง อธิบายพอ

สังเขป

9. โครงสร้างระดับชั้นของซอฟต์แวร์ที่ใช้ควบคุมการทำงานของอุปกรณ์แบ่งออกเป็นกี่ระดับชั้น แต่ละระดับชั้นมีลักษณะการทำงานอย่างไร อธิบายพอสังเขป

10. การแก้ปัญหาในการรับ-ส่งข้อมูลระหว่างอุปกรณ์อินพุต / เอาต์พุตและซีพียูสามารถใช้เทคนิคใดบ้าง อธิบายพอสังเขป

ตอนที่ 2 ให้ผู้เรียนแบ่งกลุ่มๆ ละ 4 คน เพื่อศึกษาข้อมูลเกี่ยวกับการจัดตารางอุปกรณ์ (I/O Scheduling) ในระบบปฏิบัติการดังต่อไปนี้ แล้วจัดทำรายงานส่งในสัปดาห์ถัดไป

1. ระบบปฏิบัติการ Linux
2. ระบบปฏิบัติการ Window 7

ตอนที่ 3 ให้ผู้เรียนสะท้อนคิดถึงสิ่งที่ได้เรียนรู้ในบทที่ 4 แล้วนำมาเขียนในลักษณะความเรียงจัดเก็บไว้ในแฟ้มสะสมผลงานเพื่อส่งผู้สอนตอนสิ้นสุดกิจกรรมการเรียนการสอนในภาคเรียนนี้