เอกสารประกอบการสอน

วิชา ระบบปฏิบัติการ 2 (Operating Systems 2) รหัส 4121402
บทที่ 5 ระบบรับและแสดงผลข้อมูล (Input/output System)
หลักสูตรระดับปริญญาตรี
พุทธศักราช 2551 (ปรับปรุง 2554)

โดย

จุฑาวุฒิ จันทรมาลี

สาขาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏสวนดุสิต

บทที่ 5 ระบบรับและแสดงผลข้อมูล (Input/output System)

การติดต่อระหว่างอุปกรณ์คอมพิวเตอร์ ในการรับข้อมูลหรือส่งผลลัพธ์ที่ได้จากการประมวลผลออก จำเป็นจะต้องมีระบบปฏิบัติการเพื่อใช้ควบคุมการดำเนินการต่างๆ เป็นไปอย่างอิสระและมีประสิทธิภาพ เรียกระบบจัดการส่วนนี้ว่า “ระบบรับและแสดงผลข้อมูล (I/O System)” ซึ่งเป็นระบบที่ใช้ควบคุมการทำงานของอุปกรณ์ต่างๆ ที่ใช้รับและแสดงผลข้อมูล การเชื่อมต่อตลอดจนการลดช่องว่างระหว่างอุปกรณ์ในการเชื่อมต่อ (Hardware Interface) กับคอมพิวเตอร์ ซึ่งมีลักษณะการทำงานและวิธีการดำเนินการเชื่อม (Application Interface) ที่แตกต่างกันให้สามารถทำงานร่วมกันได้

5.1 หลักการทำงานและประเภทของ I/O System

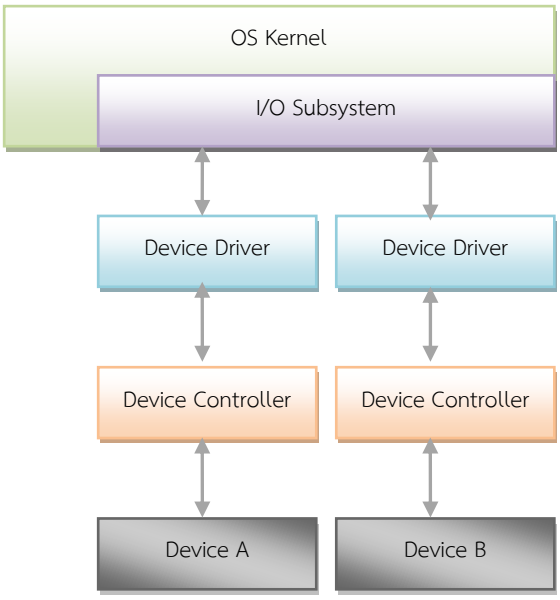
ระบบรับและแสดงผลข้อมูล (I/O System) เป็นระบบที่ใช้ควบคุมการทำงานของอุปกรณ์ต่างๆ ที่ใช้รับและแสดงผลข้อมูล ซึ่งเรียกอุปกรณ์ประเภทนี้ว่า “อุปกรณ์ต่อพ่วง (Peripheral Device)” ซึ่งในปัจจุบันมีการพัฒนาอุปกรณ์ที่ทำหน้าที่ในการรับหรือส่งข้อมูลขึ้นมาหลากหลายประเภท ดังนี้

1. อุปกรณ์ประเภทรับข้อมูลเข้า เช่น เมาส์ (Mouse) แป้นพิมพ์ (Keyboard) จอสัมผัส (Touch Screen) และไมโครโฟน (Microphone) เป็นต้น
2. อุปกรณ์ประเภทแสดงผลข้อมูลออก เช่น จอภาพ (Monitor) ลำโพง (Speaker) และเครื่องพิมพ์ (Printer) เป็นต้น
3. อุปกรณ์ประเภทบันทึก เช่น ดิสก์ (Disk) เทป (Tape) และซีดีรอม (CD-ROM) เป็นต้น

จะเห็นว่าอุปกรณ์แต่ประเภทมีลักษณะการทำงานและความเร็วในการรับส่งข้อมูลที่แตกต่างกัน ดังนั้น ระบบปฏิบัติการจะต้องให้การสนับสนุนตลอดจนกลไกในการควบคุมอุปกรณ์แต่ละประเภทและจะต้องทราบขั้นตอนและรูปแบบการทำงานของอุปกรณ์แต่ละประเภทให้สามารถทำงานได้อย่างถูกต้องและเกิดประสิทธิภาพสูงสุด

ดังนั้นจึงจำเป็นต้องออกแบบและพัฒนาระบบปฏิบัติการให้รองรับการทำงานที่เป็นอิสระต่อประเภทและอุปกรณ์แต่ละชนิด (Device Independence) ซึ่งจะมองอุปกรณ์ต่างๆ เป็นเพียงช่องทางในการรับหรือส่งข้อมูลเท่านั้น โดยไม่ต้องคำนึงถึงคุณลักษณะพิเศษของอุปกรณ์นั้นๆ จึงทำให้ระบบปฏิบัติการสามารถติดต่อกับอุปกรณ์ทุกประเภทได้อย่างเป็นอิสระและมีประสิทธิภาพ โดยเรียกระบบย่อยที่ใช้จัดการภายใน (Kernel) ที่เกี่ยวข้องกับอุปกรณ์ I/O นี้ว่า “ระบบรับและแสดงผลข้อมูล I/O System” คอยทำหน้าที่ในการสั่งงาน ควบคุม และตรวจสอบสถานะของอุปกรณ์ I/O ทั้งหมด สามารถที่จะเรียกใช้อุปกรณ์ I/O ได้อย่างสะดวก แสดงโครงสร้างการทำงานของระบบรับและแสดงผลข้อมูล ซึ่งแต่ละอุปกรณ์ที่เชื่อมต่อกับระบบปฏิบัติการจะ

มีฮาร์ดแวร์ที่ใช้ควบคุมการทำงานของอุปกรณ์ต่างๆ โดยเฉพาะ ประกอบด้วย ตัวควบคุมอุปกรณ์ (Device Controller) ทำหน้าที่รับและส่งข้อมูลและตัวขับ (Device Driver) เป็นซอฟต์แวร์ทำหน้าที่ติดต่อกับอุปกรณ์ I/O แต่ละประเภท แสดงดังรูปที่ 5.1



รูปที่ 5.1 โครงสร้างพื้นฐานของระบบรับและแสดงผลข้อมูล (Infrastructure of I/O System)

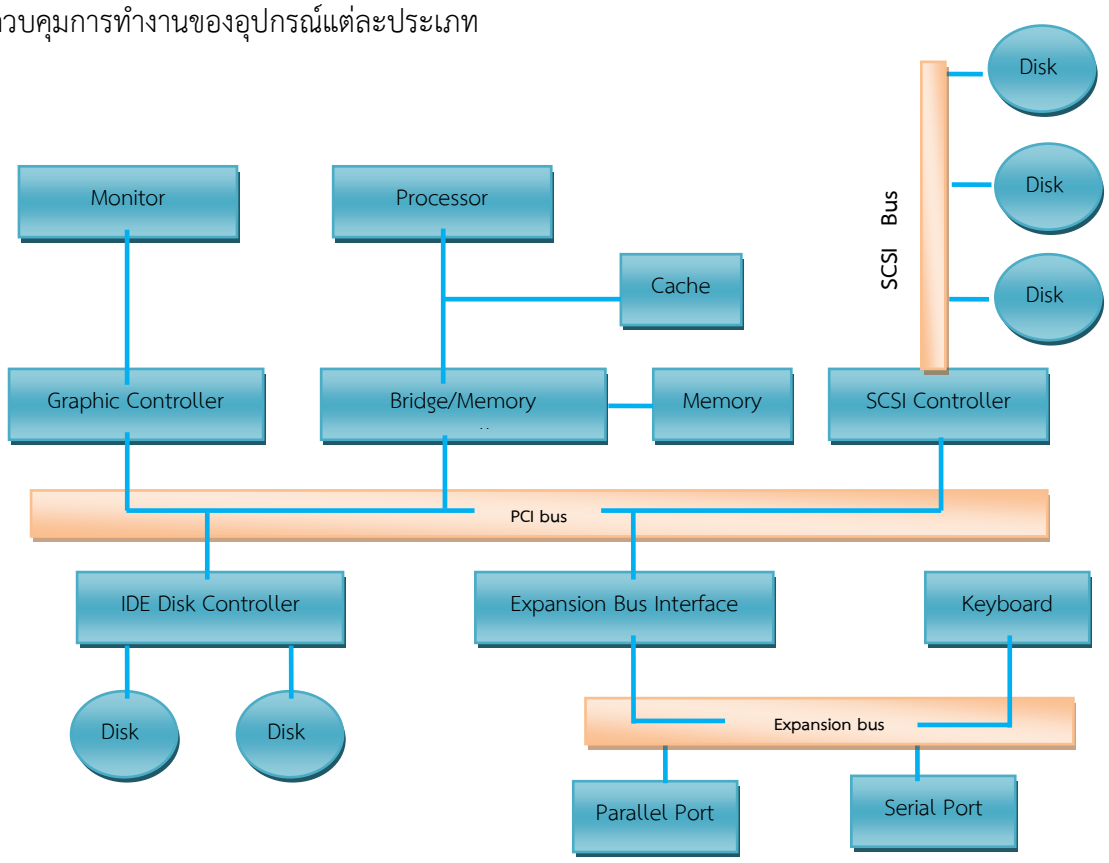
5.2 ฮาร์ดแวร์ของรับและแสดงผลข้อมูล (I/O Hardware)

ฮาร์ดแวร์ของหน่วยรับและแสดงผลข้อมูล (I/O Hardware) ประกอบด้วยอุปกรณ์ของหน่วยรับและแสดงผลประเภทต่างๆ ที่มีความสัมพันธ์กัน ดังนั้นระบบคอมพิวเตอร์จะต้องสามารถที่จะใช้ซอฟต์แวร์เชื่อมต่อเพื่อควบคุมการทำงานรวมกันระหว่างอุปกรณ์ฮาร์ดแวร์จำนวนมาก ซึ่งอุปกรณ์ฮาร์ดแวร์แต่ละประเภทก็จะมีวิธีการเชื่อมต่อที่แตกต่างกันออกไป ดังนั้นการเชื่อมต่ออุปกรณ์ I/O ต่างๆ เขากับระบบคอมพิวเตอร์จำเป็นจะต้องมีส่วนเชื่อมต่อที่เป็นมาตรฐานเพื่อใช้ในการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์โดยการส่งสัญญาณไปตามตามสายส่งผ่านส่วนเชื่อมต่อ เช่น

1. การติดต่อผ่านจุดเชื่อมต่อที่เรียกว่า “พอร์ต (Port”) เป็นส่วนเชื่อมต่อที่ติดตั้งมาพร้อมกับตัวเครื่องคอมพิวเตอร์ มักใช้เชื่อมต่อกับอุปกรณ์ตัวเดียว เช่น การเชื่อมต่อแบบอนุกรม (Serial Port) การเชื่อมต่อแบบขนาน (Parallel Port) การเชื่อมต่อของเมาส์ (Mouse) เป็นต้น

2. การติดต่อผ่านจุดเชื่อมต่อที่เรียกว่า “บัส (Bus)” เป็นส่วนเชื่อมต่อที่ใช้เชื่อมต่อกับอุปกรณ์หลายตัว โดยการออกแบบโปรโตคอลเพื่อใช้กลุ่มคำสั่งในการควบคุมการส่งสัญญาณผ่ายสายส่งเรียกวิธีการนี้ว่า “Daisy Chain” ซึ่งเป็นวิธีการที่ใช้สายสัญญาณหรือจุดเชื่อมต่ออุปกรณ์หลายประเภทเข้าด้วยกัน แสดงดังรูปที่ 5.2 ซึ่งแสดงการเชื่อมต่ออุปกรณ์ในระบบคอมพิวเตอร์ทั่วไป ผ่านจุดเชื่อมต่อที่เรียกว่า “PCI Bus” ที่เชื่อมต่อกับ จอภาพ (Monitor) หน่วยประมวลผล (Processor) หน่วยความจำ(Memory) และอุปกรณ์ประเภทอื่นที่มีความเร็วสูง และระบบการเชื่อมต่อโดยใช้อินเตอร์เฟซที่เรียกว่า SCSI (Small Computer System Interface) หรือ IDE (Integrated Development Environment) ซึ่งเป็นมาตรฐานอิเล็กทรอนิกส์อินเตอร์เฟซวิธีหนึ่งเพื่อใช้ในการเชื่อมต่ออุปกรณ์เพื่อแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์ฮาร์ดแวร์กับบัส เช่น อุปกรณ์จำพวกดิสก์ (Disk) นอกจากนี้ยังบัสส่วนขยาย (Expansion Bus) เพื่อใช้เชื่อมต่ออุปกรณ์ที่มีความเร็ว

ต่ำ เช่น แป้นพิมพ์ (Keyboard) เป็นต้น โดยที่อุปกรณ์แต่ละประเภทจะมีหน่วยควบคุม (Controller) ซึ่งเป็นแผงวงจรอิเล็กทรอนิกส์ที่ประกอบเข้าเป็นตัวชิบ (Chip) บรรจุลงบนแผ่นวงจร (Circuit Broad) ทำหน้าที่ในการควบคุมการทำงานของอุปกรณ์แต่ละประเภท



รูปที่ 5.2 โครงสร้างการเชื่อมต่ออุปกรณ์แต่ละชนิดของระบบคอมพิวเตอร์
(A Typical PC Bus Structure)

ในการควบคุมการรับส่งคำสั่งและข้อมูลระหว่างอุปกรณ์ผ่านทางจุดเชื่อมต่อ (Port) จะอยู่ภายใต้การควบคุมของโดยหน่วยประมวลผลกลาง (CPU) ซึ่งจะมีการใช้คำสั่ง (I/O Instruction) ในการสลับช่องทาง (Triggers Bus Lines) เพื่อติดต่อกับอุปกรณ์และส่งข้อมูลเข้าออกในระดับบิตซึ่งจะทำให้เกิดการเปลี่ยนแปลงค่ารีจิสเตอร์ (Register) บางตัวในหน่วยควบคุมอุปกรณ์นั้นและเรียกวิธีการนี้ว่า “Memory-Mapped I/O” โดยตำแหน่งของจุดเชื่อมต่อของอุปกรณ์รับและแสดงผลแต่ละประเภท แสดงได้ดังรูปที่ 5.3 ซึ่งจะเกี่ยวข้องกับรีจีสเตอร์ที่สำคัญ 4 ชนิด ดังนี้

1.รีจีสเตอร์บอกสถานะ (The Status Register) ประกอบด้วยบิตเพื่อบอกสถานะว่าสามารถอ่านข้อมูลหรือแสดงความผิดพลาดของอุปกรณ์ (Device Error)

2.รีจีสเตอร์ควบคุม (The Control Register) ประกอบด้วยคำสั่งในการควบคุมอัตราความเร็วการรับส่งข้อมูลรวมถึงการใช้บิตตรวจสอบ (Parity Checking Bit)

3.รีจีสเตอร์แสดงข้อมูลเข้า (The Data-In Register) ประกอบด้วยคำสั่งที่แสดงว่าพร้อมที่จะอ่านข้อมูลเข้ามาเก็บในรีจีสเตอร์

4.รีจีสเตอร์แสดงข้อมูลเข้า (The Data-Out Register) ประกอบด้วยคำสั่งที่แสดงว่าพร้อมที่จะเขียนข้อมูลและส่งข้อมูลออกไปยังอุปกรณ์ I/O

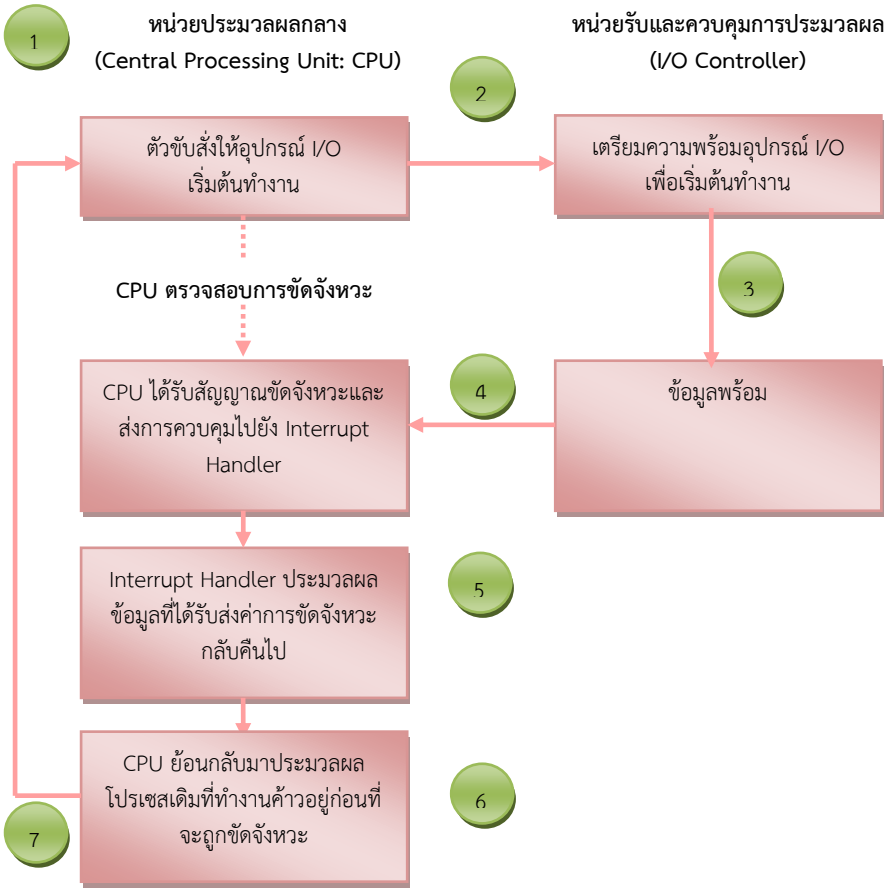
ตำแหน่งของอุปกรณ์รับและแสดงผล (เลขฐานสิบหก)	ประเภทอุปกรณ์ (Device)
000-00F	ตัวควบคุมการเข้าถึงหน่วยความจำโดยตรง (DMA Controller)
020-021	ตัวควบคุมการขัดจังหวะ (Interrupt Controller)
040-043	ตัวควบคุมเวลา (Timer)
200-20F	ตัวควบคุมเกมส์ Game Controller
2F8-2FF	พอร์ตแบบอนุกรม (พอร์ต 2) Serial Port (Secondary)
320-32F	ตัวควบคุมฮาร์ดดิสก์ Hard-Disk Controller
378-37F	พอร์ตแบบขนาน Parallel Port
3D0-3DF	ตัวควบคุมกราฟิก Graphics Controller
3F0-3F7	ตัวควบคุมดิสก์ไดร์ฟ Diskettes-Drive Controller
3F8-3FF	พอร์ตแบบอนุกรม (พอร์ต 1) Serial Port (Secondary)

รูปที่ 5.3 ตำแหน่งของจุดเชื่อมต่อของอุปกรณ์รับและแสดงผลแต่ละประเภท
(Device I/O Port Location on PCs :partial)

วิธีรับหรือส่งข้อมูลระหว่างหน่วยประมวลผลกลางกับอุปกรณ์นำเข้า/แสดงผล (CPU with I/O Device)

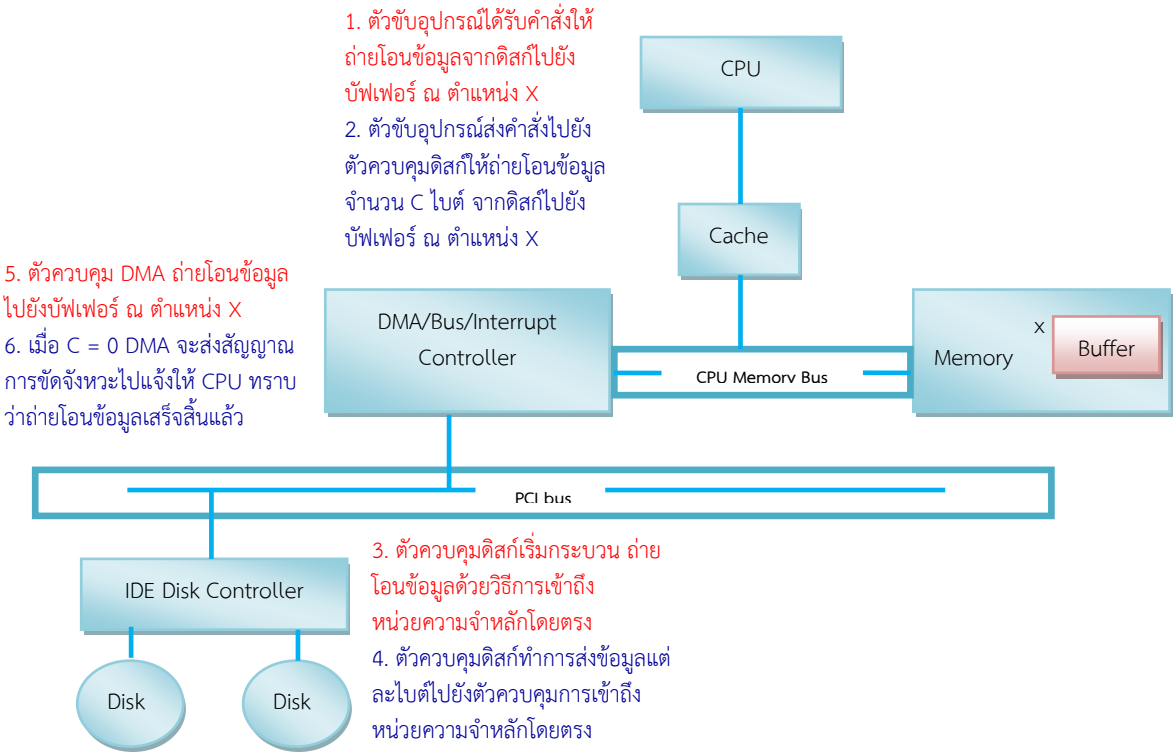
1. Polling เป็นวิธีที่การส่งสัญญาณโต้ตอบกันระหว่างหน่วยประมวลผลกลาง (CPU) กับอุปกรณ์ I/O เพื่อของทราบสถานะทำงานของอุปกรณ์ I/O นั้นๆ ซึ่งเกี่ยวข้องกับคำสั่งแสดงสถานะการทำงาน ดังนี้
1.1 คำสั่งพร้อม (Command Ready) เป็นคำสั่งที่แสดงว่าอุปกรณ์พร้อมที่จะรับคำสั่งให้ทำงานตามคำขอ
1.2 ไม่ว่าง-รอก่อน (Busy-Waiting) เป็นคำสั่งที่แสดงว่าอุปกรณ์ไม่พร้อมที่จะทำงาน เนื่องจากมีการใช้งานอุปกรณ์ I/O นั้นอยู่ ต้องรอจนกว่าอุปกรณ์ I/O นั้นจะถูกใช้งานจนเสร็จ
1.3 ผิดพลาด (Error) เป็นคำสั่งที่แสดงว่าอุปกรณ์ขัดข้อง หรือไม่สามารถรับคำสั่งใดๆ ได้ วิธีการแบบ Polling นี้มีข้อเสียตรงที่ว่าหน่วยประมวลกลาง (CPU) ต้องเสียเวลาในการรอคอยการใช้งานอุปกรณ์ I/O บางประเภทเป็นเวลานานๆ ทำให้ไม่สามารถทำงานประเภทอื่นต่อไปได้เพราะต้องรอจนกว่าอุปกรณ์ I/O นั้นจะถูกใช้งานจนเสร็จ เช่น การอ่านหรือบันทึกข้อมูลลงในดิสก์ เป็นต้น ซึ่งมักเรียกสภาวะการที่เกิดขึ้นนี้ว่า “Busy-Wait Cycle”
2. การขัดจังหวะ (Interrupts) เป็นวิธีที่ช่วยให้หน่วยประมวลกลาง (CPU) สามารถทำงานอย่างอื่นได้โดยไม่ต้องรอการใช้งานอุปกรณ์ I/O จนเสร็จ โดยการส่งสัญญาณการไปขัดจังหวะ (Interrupt) การทำงานของหน่วย

ประมวลกลาง (CPU) เพื่อบอกว่าอุปกรณ์ I/O พร้อมที่จะรับส่งข้อมูลแล้วผ่านทาง Interrupt-Request Line เมื่อ CPU ได้รับสัญญาณขัดจังหวะผ่านช่องทางดังกล่าวก็จะหยุดการทำงานชั่วคราวเพื่อให้บริการกับอุปกรณ์ที่ร้องขอการขัดจังหวะเข้ามา ซึ่งแสดงขั้นตอนการวนรอบการขัดจังหวะของอุปกรณ์รับและแสดงผลดังรูปที่ 5.4



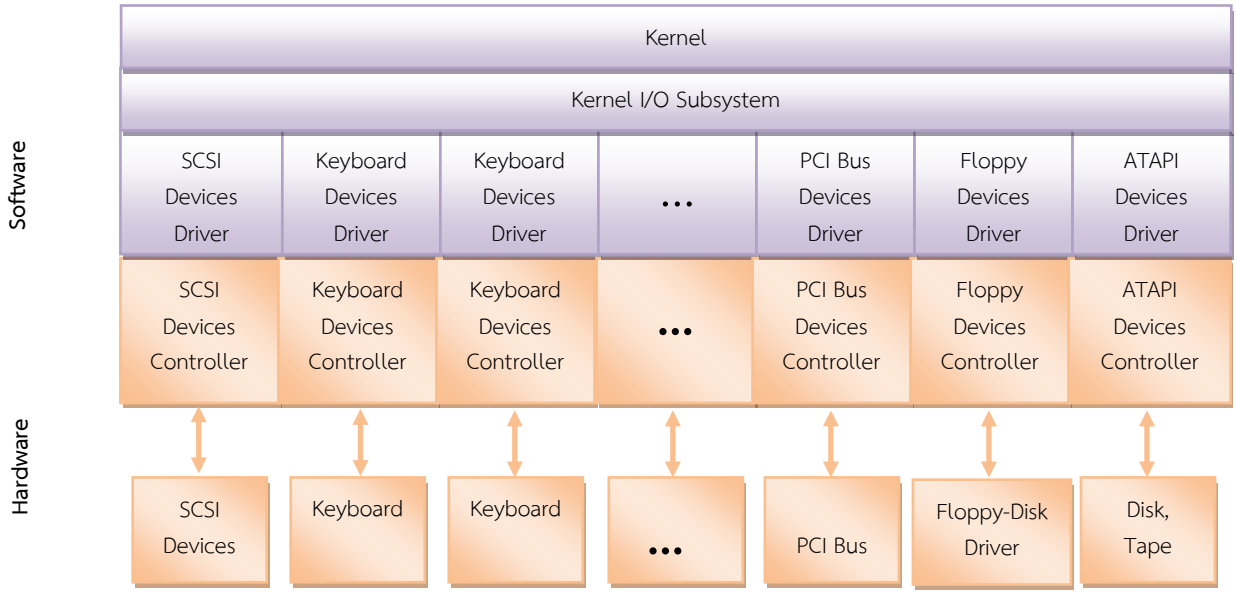
รูปที่ 5.4 การวนรอบการขัดจังหวะของอุปกรณ์รับและแสดงผล (Interrupt-Device I/O Cycle)

3. การเข้าถึงหน่วยความจำหลักโดยตรง (Direct Access Memory) เป็นวิธีที่พัฒนาขึ้นมาเพื่อการรับหรือส่งข้อมูลปริมาณมากๆ ในเวลาเดียวกัน โดยใช้อุปกรณ์ตัวควบคุมการดำเนินการในการรับส่งข้อมูลระหว่างหน่วยความจำหลัก (Main Memory) กับอุปกรณ์ I/O ที่เรียกว่า ตัวควบคุมการเข้าถึงหน่วยความจำหลักโดยตรง (Direct Access Memory: DMA) โดยไม่จำเป็นต้องผ่านหน่วยประมวลผลกลาง (CPU) จึงทำให้การรับหรือส่งข้อมูลทำได้เร็วขึ้นเพราะ CPU ไม่ต้องเสียเวลาในรอการทำงานของอุปกรณ์ I/O ทำให้สามารถใช้งาน CPU ได้อย่างเต็มที่ ตัวควบคุมการเข้าถึงหน่วยความจำหลักโดยตรง (DMA Controller) จะทำการส่งผ่านข้อมูลจำนวน X ไบต์จากอุปกรณ์หน่วยความจำหลัก โดยส่งสัญญาณการขัดจังหวะ (Interrupt) การทำงานไปยัง CPU เมื่อ CPU ตอบกลับ (Acknowledge) ตัวควบคุมการเข้าถึง (DMA Controller) จะทำการยึดคลองบัส (Bus) และเริ่มขบวนการส่งผ่านข้อมูลซึ่งในขณะนั้น CPU ก็สามารถที่จะทำงานด้านอื่นๆ ต่อไปได้ แสดงได้ดังรูปที่ 5.5



รูปที่ 5.5 แสดงขั้นตอนการทำงานของการเข้าถึงหน่วยความจำหลัก (Step in a DMA Transfer)

5.3 ส่วนต่อประสานงานในระบบรับและแสดงผล (Application I/O Interface) เป็นมาตรฐานการใช้งานอุปกรณ์แต่ละประเภทที่มีความแตกต่างกันสามารถติดต่อกันระหว่างระบบปฏิบัติการกับอุปกรณ์ I/O ที่มีความแตกต่างกันในหลายๆ ด้าน เช่น บริษัทผู้ผลิต ชนิดหรือประเภทของอุปกรณ์ เป็นต้น เรียกมาตรฐานนี้ว่า “ส่วนประสาน (Interface)” ซึ่งความแตกต่างของตัวอุปกรณ์ I/O จะถูกซ่อนรายละเอียดไว้ภายในคอร์เนลโมดูล (Kernel Modules) โดยใช้ไดรเวอร์ (Driver) เป็นตัวควบคุมการทำงานของอุปกรณ์ I/O แต่ละประเภท เพื่อให้การติดต่อระหว่างอุปกรณ์ต่างๆ กับระบบปฏิบัติการใช้มาตรฐานเดียวกัน แสดงโครงสร้างการเชื่อมต่อคอร์เนล (Kernel) ของหน่วยรับและแสดงผลข้อมูล ดังรูปที่ 5.6



ดังรูปที่ 5.6 แสดงโครงสร้างการเชื่อมต่อคอร์เนลของหน่วยรับและแสดงผลข้อมูล
(A Kernel I/O Structure)

นอกจากนี้การพัฒนาอุปกรณ์ I/O ให้สอดคล้องกับลักษณะการทำงานร่วมกับระบบปฏิบัติการแต่ละชนิดซึ่งต่างก็มีมาตรฐานการใช้อุปกรณ์ในการเชื่อมต่อ (Device Driver interface) ที่แตกต่างกัน ขึ้นอยู่กับลักษณะการทำงาน ฟังก์ชันและรูปแบบการนำไปประยุกต์ใช้งานร่วมกับระบบคอมพิวเตอร์ ดังแสดงได้จากตารางที่ 5.1

ตารางที่ 5.1 แสดงความแตกต่างของลักษณะการทำงานของอุปกรณ์ I/O แต่ละประเภทในแง่มุมต่างๆ

แง่มุม (Aspect)	ความแตกต่าง (Variation)	อุปกร์รับและแสดงผล (I/O Device)
รูปแบบการส่งข้อมูล (Data Transfer Mode)	ตัวอักขระ (Character) บล็อก (Bock)	อุปกรณ์แสดงผล (Terminal) ดิสก์ (Disk)
วิธีการเข้าถึงข้อมูล (Access Method)	เรียงลำดับ (Synchronous) ไม่เรียงลำดับ (Asynchronous)	โมเด็ม (Modem) ซีดีรอม (CD-ROM)
การระบุการส่ง (Transfer Schedule)	ตามลำดับ (Sequential) แบบสุ่ม (Random)	เทป (Tape) แป้นพิมพ์ (Keyboard)
การใช้งานร่วมกัน (Sharing)	ใช้งานเฉพาะ (Dedicated) ทำงานร่วมกัน (Sharable)	เทป (Tape) แป้นพิมพ์ (keyboard)
ความเร็วของอุปกรณ์ (Device Speed)	Latency, Seek time Transfer rate, Delay between operations	
ณุปแบบการเข้าถึงอุปกรณ์ I/O (I/O Direction)	อ่านอย่างเดียว (Read Only) เขียนอย่างเดียว (Write Only) อ่านและเขียน (Read only)	ซีดีรอม (CD-ROM) ตัวควบคุมกราฟิก (Graphic Controller) ดิสก์ (Disk)

ประเภทของอุปกรณ์ที่ทำหน้าที่ติดต่อกับระบบรับและแสดงผลข้อมูล (I/O System)

1. อุปกรณ์ที่ใช้รับหรือส่งข้อมูลเป็นกลุ่ม (Block and Character Device) เป็นอุปกรณ์ที่ใช้รับหรือส่งข้อมูลเป็นกลุ่ม (Block) จะเกี่ยวข้องกับอุปกรณ์คือ ดิสก์ (Disk) ซึ่งบางครั้งอาจจะเรียกวิธีการเข้าถึงแบบนี้ว่าการรับหรือแสดงผลเป็นแถว (Raw I/O) โดยคำสั่ง System Calls เพื่อใช้ในการเชื่อมต่ออุปกรณ์ I/O ดังกล่าว เช่น คำสั่ง Read () คำสั่ง Write () และคำสั่ง Seek () เป็นต้น ส่วนอุปกรณ์ที่เกี่ยวข้องกับการรับหรือส่งข้อมูลที่ละตัวอักษร (Character) มักจะเกี่ยวข้องกับอุปกรณ์คือ แป้นพิมพ์ (Keyboard) เมาส์ (Mouse) และโมเด็ม (Modem) โดยคำสั่ง System Calls เพื่อใช้ในการเชื่อมต่ออุปกรณ์ I/O ดังกล่าวเช่น คำสั่ง (Get) และคำสั่ง (Put) เป็นต้น

2. อุปกรณ์ที่ใช้รับหรือส่งข้อมูลบนเครือข่าย (Network Devices) เป็นอุปกรณ์ที่ใช้รับหรือส่งข้อมูลบนเครือข่าย ผ่านโปรแกรมที่เชื่อมต่อที่เรียกว่า Socket Interface โดยติดตั้งมากับระบบปฏิบัติการโดยเฉพาะ เช่น UNIX และ Windows NT เป็นต้น ภายใน Socket จะมีฟังก์ชันการทำงานในการเชื่อมต่อเพื่อรับและส่งข้อมูลอยู่ภายในเรียกว่าฟังก์ชัน Select () โดยหน้าที่หลักของฟังก์ชันนี้จะทำการส่งคืนค่าสถานะในขณะนั้นกลับไปยังส่วนที่มีการเรียกใช้งานโดยจะมี Socket Packet เพื่อคอยรับข้อมูลในฝังรับและมี Socket Room ทำหน้าส่งข้อมูลในฝังส่งตามที่กำหนด

3. อุปกรณ์สัญญาณนาฬิกาและอุปกรณ์บอกเวลา (Clock and Timers) เป็นอุปกรณ์ติดตั้งมากับเครื่องคอมพิวเตอร์โดยมีทำหน้าที่หลัก 3 อย่าง คือ

1. แสดงว่า ณ ปัจจุบัน
2. แสดงเวลาที่ผ่านพ้นไป
3. ตั้งเวลาเพื่อที่จะดำเนินการกระตุ้น (Trigger) เช่น เวลาที่จะใช้ในการขัดจังหวะ (Interrupt)

ทั้ง 3 หน้าที่อยู่ภายใต้การดำเนินการของระบบปฏิบัติการ โดยมีอุปกรณ์ที่ใช้แสดงเวลาที่ผ่านพ้นไปตั้งและตั้งเวลาเพื่อที่จะดำเนินการกระตุ้น (Trigger) เรียกว่า ”Programmable Interval Timer” โดยสามารถที่จะตั้งเวลารอคอย (Wait) และเวลาที่จะขัดจังหวะ (Interrupt) ในการดำเนินการของกระบวนการ (Process) ในแต่ละครั้งได้

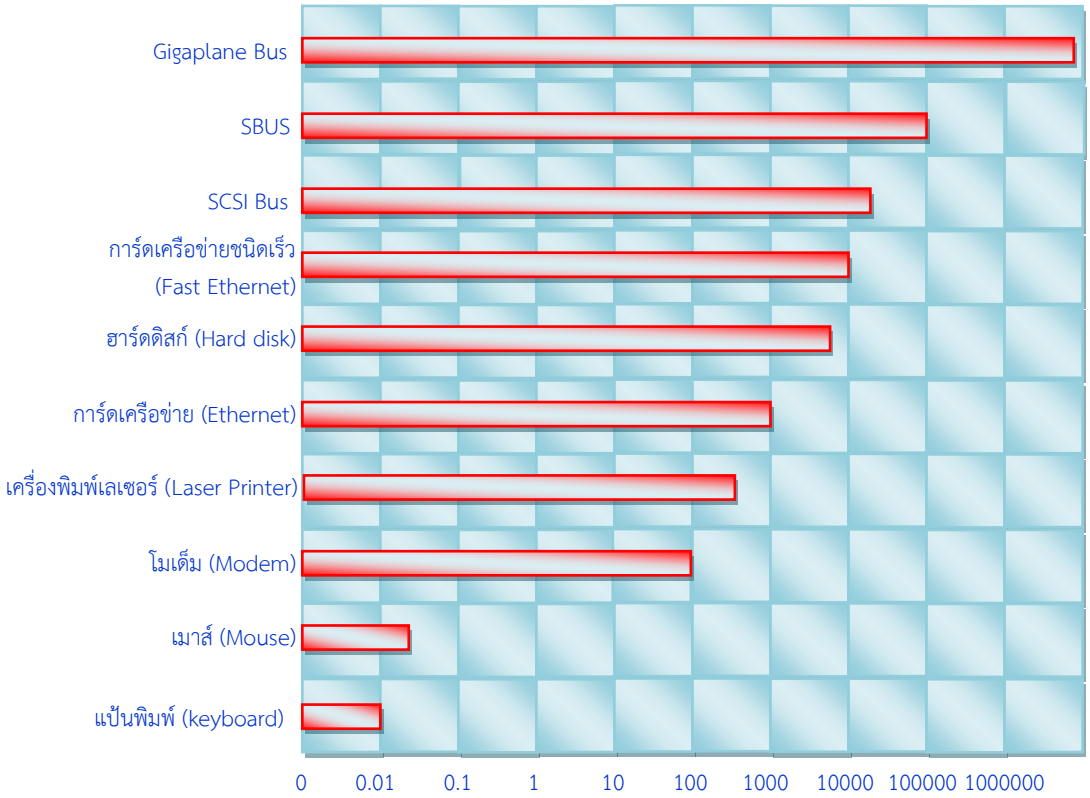
4. อุปกรณ์รับ/ส่งข้อมูลภายใต้กำหนดเวลาที่แน่นอนและไม่แน่นอน (Blocking and Nonblocking I/O) บางครั้งอาจจะเรียกอุปกรณ์ประเภทนี้ว่า Synchronous and Asynchronous Device โดยอุปกรณ์ประเภทที่รับหรือส่งข้อมูลที่กำหนดเวลาที่แน่นอน (Synchronous) เช่น เทป เป็นต้น หลักการทำงานแบบนี้อยู่ภายใต้คำสั่งระบบเรียกใช้งาน (System Call) โดยโปรเซส (Process) ที่กำลังประมวลผลอยู่จะหยุดการทำงานชั่วคราว เพื่อให้คำสั่งที่ระบบเรียกใช้งาน (System Call) ทำงานจนเสร็จสิ้นก่อน จึงจะสามารถกลับมาประมวลผลโปรเซส (Process) เดิมต่อไปได้ สำหรับอุปกรณ์ประเภทที่รับหรือส่งข้อมูลที่กำหนดเวลาที่ไม่แน่นอน (Asynchronous) เช่น แป้นพิมพ์ (Keyboard) เมาส์ (Mouse) เป็นต้น โปรเซส (Process) ที่กำลังประมวลผลอยู่ไม่จำเป็นต้องหยุดการทำงาน สามารถทำงานจนเสร็จสิ้นก่อน จึงคอยให้คำสั่งที่ระบบเรียกใช้งาน (System Call) เข้ามาประมวลผล ลักษณะการทำงานแบบนี้เรียกว่า เหตุการณ์ที่เกิดขึ้นต่อเนื่องกันหลายเหตุการณ์ (Multithreading) เช่น ขณะที่ใช้งานอุปกรณ์ในการเขียนข้อมูล (Writer) ในขณะเวลาเดียวกันก็สามารถประมวลผลอุปกรณ์ I/O ควบคู่กันไปด้วย เป็นต้น

5.4 ระบบรับหรือแสดงผลย่อย (Kernel I/O Subsystem)

ระบบปฏิบัติการโดยทั่วไปจะมีส่วนคอร์เนล (Kernel) ไว้คอยสนับสนุนการให้บริการต่างๆ ที่เกี่ยวพันกับอุปกรณ์ I/O หลายอย่าง เช่น การจัดตารางอุปกรณ์ (I/O Scheduling) ที่พักข้อมูล (Buffering) การเก็บข้อมูลไว้ในหน่วยความจำ (Caching) การจัดการข้อผิดพลาด (Error Handing) การจัดโครงสร้างข้อมูลภายในคอร์เนล (Kernel Data Structures) เป็นต้น ซึ่งถูกจัดการโดยระบบรับหรือแสดงผลย่อย (Kernel I/O Subsystem)

1. การจัดตารางอุปกรณ์ (I/O Scheduling) เป็นการกำหนดความสำคัญในการใช้งานอุปกรณ์ต่างๆ อย่างเป็นระบบ โดยการบันทึกข้อมูลการทำงานของอุปกรณ์ I/O แต่ละตัวไว้ใน ตารางแสดงสถานะการทำงานขแงแต่ละอุปกรณ์ (Device-Status Table) เพื่อตรวจสอบสถานะว่าอุปกรณ์ I/O นั้นพร้อมที่จะทำงานหรือไม่ ถ้าพร้อม ระบบปฏิบัติการก็จะทำการตรวจสอบสัญญาณขัดจังหวะ (Interrupt) ว่ามาจากอุปกรณ์ I/O ตัวใดก็จะทำการปรับเปลี่ยนค่าให้ถูกต้องตามสัญญาณขัดจังหวะนั้น และพร้อมกับทำการตรวจสอบว่ามีอุปกรณ์ I/O ใดเข้าคิว (Queue) ก็จะทำงานตามคำร้องขอ (Request) ถัดไป เมื่อดำเนินการเสร็จสิ้นแล้วก็จะทำการคืนค่าการควบคุมเพื่อกลับไปประมวลผลงานเดิมก่อนที่จะถูกขัดจังหวะ

2. ที่พักข้อมูล (Buffering) เป็นพื้นที่ในหน่วยความจำ (Memory) ที่ใช้เก็บข้อมูลขณะที่มีการถ่ายโอนระหว่างอุปกรณ์ 2 ชนิดกันหรือระหว่างอุปกรณ์กับงานที่ทำอยู่ เนื่องจากอัตราความเร็วในการทำงานของแต่ละอุปกรณ์กับหน่วยประมวลผลกลาง (CPU) มีความเร็วไม่เท่ากัน ดังนั้นจำเป็นจะต้องมีที่พักข้อมูลชั่วคราวเพื่อให้การถ่ายโอนข้อมูลระหว่าง 2 อุปกรณ์มีความต่อเนื่อง และเพิ่มประสิทธิภาพการทำงานให้กับระบบคอมพิวเตอร์ แสดงดังรูปที่ 5.7



ดังรูปที่ 5.6 แสดงความเร็วในการโอนข้อมูลของแต่ละอุปกรณ์
(Sun Enterprise 6000 Device-Transfer Rate: Logarithmic)

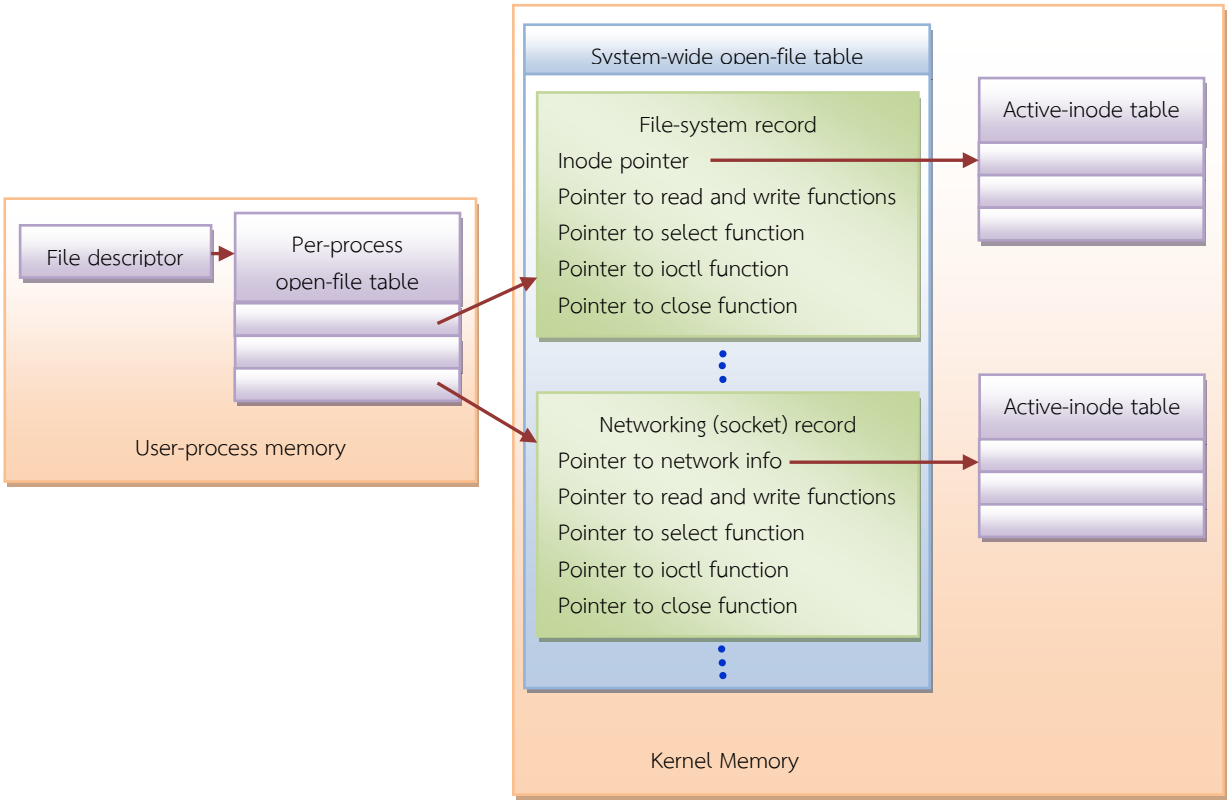
3. การเก็บข้อมูลไว้ในหน่วยความจำ (Caching) โดยที่แคลช (Cache) เป็นหน่วยความจำความเร็วสูงที่อยู่ระหว่างหน่วยประมวลผลกลาง (CPU) และหน่วยความจำหลัก (Main Memory) ทำหน้าที่ในการพักข้อมูลไว้ชั่วคราว โดยการนำข้อมูลบางส่วนมาใส่ไว้ในแคลช (Cache) เนื่องจากการอ่านข้อมูลได้เร็วกว่า แต่ถ้าไม่มีข้อมูลในแคลช (Cache) ก็จะไปอ่านข้อมูลในหน่วยความจำแทน การใช้งานของแคลช (Cache) มีรูปแบบการทำงานที่ไม่แตกต่างจากที่พักข้อมูล (Buffering) มากนัก เพียงแต่ทำงานได้เร็วกว่า แต่มีราคาค่อนข้างแพง ดังนั้นงานที่ต้องการความเร็วมักจะนิยมใช้ แคลช (Cache) เป็นตัวจัดการ

4. การพักข้อมูลชั่วคราว (Spooling and Device Reservation) เป็นที่พักข้อมูลชัวครวสำหรับจัดเก็บข้อมูลที่จะถูกส่งออกไปยังอุปกรณ์ I/O เพื่อแก้ปัญหาของอุปกรณ์ที่ไม่สามารถสลับการทำงานหรือทำงานพร้อมกันระหว่างโปรเซส (Process) ได้ การใช้เครื่องพิมพ์ร่วมกัน (Share Printer) ระบบปฏิบัติการจะวิธีการพักข้อมูลชั่วคราว (Spooling) เพื่อจัดลำดับคิวงานในการส่งข้อมูลเข้ามาให้กับเครื่องพิมพ์ และใน

ขณะเดียวกันก็สามารถที่จะยกเลิกการพิมพ์ได้เพราะลำดับของคิวต่างๆ ที่เก็บอยู่ในที่พักข้อมูลชั่วคราว (Spooling) ผู้ใช้สามารถควบคุมการทำงานผ่านระบบปฏิบัติการได้

5. การจัดการข้อผิดพลาด (Error Handing) ระบบปฏิบัติจะทำหน้าที่เป็นตัวกลางในการประสานการทำงานระหว่างอุปกรณ์ต่างๆและโปรแกมสำเร็จรูป (Application Program) ของผู้ใช้ ดังนั้นระบบปฏิบัติจะต้องคอยป้องกันการใช้งานหน่วยความจำ (Protected Memory) และจัดการข้อผิดพลาดที่อาจจะเกิดขึ้นในขณะส่งผ่านข้อมูลหรือที่มีการเรียกใช้งานอุปกรณ์ I/O นั้นๆ อยู่เพื่อให้ระบบไม่เกิดการติดขัดและสามารถทำงานต่อไปได้ เช่น ข้อผิดพลาดที่เกิดขึ้นกับเครื่องพิมพ์กรณีการที่ใช้พิมพ์อยู่หมด ระบบปฏิบัติจะต้องมีการจ้าข้อผิดพลาดในการใช้งานให้กับผู้ใช้ทราบได้ทันที่ เป็นต้น

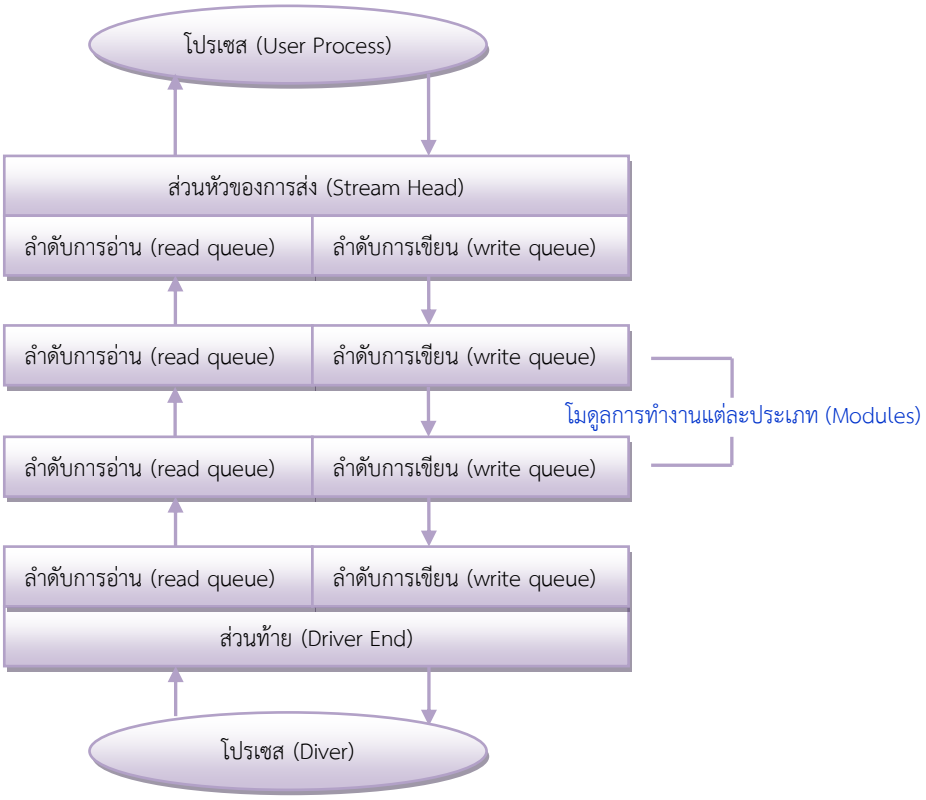
6. การจัดโครงสร้างข้อมูลภายในคอร์เนล (Kernel Data Structures) ภายในคอร์เนล (Kernel) ต้องการที่จะเก็บสถานะการทำงานที่เกี่ยวข้องกับการใช้งานอุปกรณ์ I/O แต่ละชนิด วิธีการหนึ่งคือการจัดโครงสร้างข้อมูลภายในคอร์เนล (Kernel Data Structures) เช่น การเปิดไฟล์ (Open-File) ดังรูปที่ 5.7 แสดงตารางโครงสร้างภายในคอร์เนล (Kernel) ในการติดตาม (Track) การเชื่อมต่อระบบเครือข่าย การติดต่อสื่อสารกับอุปกรณ์ส่งผ่านข้อมูลอักขระ (Character-Device) หรือกิจกรรมอื่นที่เกี่ยวข้องกับการใช้อุปกรณ์ I/O เป็นต้น



ดังรูปที่ 5.7 แสดงความเร็วในการโอนข้อมูลของแต่ละอุปกรณ์
(UNIX I/O Kernel Structure)

5.5 การส่งข้อมูลอย่างต่อเนื่อง (Streams)

เป็นวิธีการจัดการในติดต่อและการส่งข้อมูลอย่างต่อเนื่องระหว่างอุปกรณ์ตัวขับ (Device Driver) กับผู้ใช้ในระดับโปรเซส (User-Level Process) ซึ่งโครงสร้างการส่งข้อมูลอย่างต่อเนื่อง (Stream) ประกอบด้วยส่วนหัวของการส่ง (Stream Head) โดยทำการเชื่อมต่อกับส่วนของโปรเซสผู้ใช้และตัวขับสุดท้ายหรือส่วนสิ้นสุด (Driver end) ทำการควบคุมอุปกรณ์สตรีมโมดูล (Stream module) แต่และตัว โดยที่ส่วนหัวของการส่ง (Stream Head) ตัวขับสุดท้ายหรือตัวสิ้นสุด (Driver end) และแต่ละโมดูลประกอบด้วยคู่ของแถวลำดับจำนวนหลายคู่ (Pair of Queues) เช่น คู่ของแถวลำดับการอ่าน (Read Queue) กับแถวลำดับการเขียน (Write Queue) โดยการส่งผ่านข้อมูลจะกระทำโดยคู่แถวลำดับในแต่ละแถวลำดับ แสดงดังรูปที่ 5.8



ดังรูปที่ 5.8 แสดงโครงสร้างการส่งข้อมูลอย่างต่อเนื่อง (The Streams Structure)

5.6 ประสิทธิภาพการทำงาน (Performance)

ปัจจัยที่มีผลกับประสิทธิภาพการทำงานของอุปกรณ์ I/O ขึ้นอยู่กับเหตุการณ์หรืองานที่เข้ามาประมวลผลในหน่วยประมวลผลกลาง (CPU) ควบคู่กับโปรแกรมขับอุปกรณ์ (Device-Driver Code) และการจัดตารางการทำงานให้แต่ละโปรเซสว่ามีประสิทธิภาพดีมากน้อยเพียงไร ตลอดจนประสิทธิภาพของการรับหรือส่งข้อมูลภายใต้กำหนดเวลาที่แน่นอนและไม่แน่นอน (Block and Unblocked) ตลอดจนยังขึ้นอยู่กับสถาปัตยกรรมของระบบคอมพิวเตอร์ที่เลือกใช้งานอีกด้วย

สรุป

พื้นฐานของระบบคอมพิวเตอร์แต่ละส่วนจะเกี่ยงข้องและติดต่อประสานการทำงานระบบรับและแสดงผลข้อมูล (I/O System) ซึ่งมีตัวควบคุมการทำงาน (Device Controllers) เป็นของตัวเอง โดยหลักการทำงานจะเป็นการโอยย้ายข้อมูลระหว่างอุปกรณ์และหน่วยความจำหลัก (Main Memory) ถูกจัดการโดยหน่วยประมวลผลกลาง (CPU) ผ่านโปรแกรมที่ใช้จัดการอุปกรณ์ I/O หรือจัดการอยู่ภายในการควบคุมการเข้าถึงหน่วยความจำหลักโดยตรง (Direct Access Memory) โดยมีระบบปฏิบัติการเข้ามาควบคุมเพื่อให้การดำเนินงานต่างเป็นไปอย่างราบรื่นและเกิดประสิทธิภาพ ระบบการทำงานย่อยที่ทำหน้าที่ในการควบคุมการเข้าถึงอุปกรณ์รับและแดงผลข้อมูลที่เรียกว่า “I/O Subsystem”

ซึ่งในปัจจุบันมีการพัฒนาอุปกรณ์ที่ทำหน้าที่ในการรับหรือส่งข้อมูลขึ้นมาหลากหลายประเภท ดังนี้

1. อุปกรณ์ประเภทรับข้อมูลเข้า เช่น เมาส์ (Mouse) แป้นพิมพ์ (Keyboard) จอสัมผัส (Touch Screen) และไมโครโฟน (Microphone) เป็นต้น
2. อุปกรณ์ประเภทแสดงผลข้อมูลออก เช่น จอภาพ (Monitor) ลำโพง (Speaker) และเครื่องพิมพ์ (Printer) เป็นต้น
3. อุปกรณ์ประเภทบันทึก เช่น ดิสก์ (Disk) เทป (Tape) และซีดีรอม (CD-ROM) เป็นต้น

ซึ่งแต่ละอุปกรณ์ที่เชื่อมต่อกับระบบปฏิบัติการจะมีฮาร์ดแวร์ที่ใช้ควบคุมการทำงานของอุปกรณ์ต่างๆ โดยเฉพาะ ประกอบด้วย ตัวควบคุมอุปกรณ์ (Device Controller) ทำหน้าที่รับและส่งข้อมูลและตัวขับ (Device Driver) เป็นซอฟต์แวร์ทำหน้าที่ติดต่อกับอุปกรณ์ I/O แต่ละประเภท

วิธีรับ-ส่งข้อมูลระหว่างหน่วยประมวลผลกลางกับอุปกรณ์นำเข้าและแสดงผลข้อมูล (CPU with I/O Device) สามารถทำได้หลายวิธี เช่น วิธีการทำ Polling วิธีการขัดจังหวะ (Interrupts) วิธีการเข้าถึงหน่วยความจำหลักโดยตรง (Direct Access Memory) เป็นต้น นอกจากนี้อุปกรณ์ I/O ยังสามารถแบ่งตามฟังก์ชันการทำงานของโปรแกรมประยุกต์ที่แต่ละบริษัทผลิตขึ้นมาใช้งาน เช่น อุปกรณ์ที่ใช้รับหรือส่งข้อมูลเป็นกลุ่ม (Block and Character Device) อุปกรณ์ที่ใช้รับหรือส่งข้อมูลบนเครือข่าย (Network Devices) อุปกรณ์สัญญาณนาฬิกาและอุปกรณ์บอกเวลา (Clock and Timers) อุปกรณ์รับ/ส่งข้อมูลภายใต้กำหนดเวลาที่แน่นอนและไม่แน่นอน (Blocking and Nonblocking I/O) เป็นต้น โดยแต่ละวิธีจะมีรูปแบบการทำงานที่แตกต่างกันขึ้นอยู่กับสถาปัตยกรรมของผู้ออกแบบระบบปฏิบัติการที่ใช้อยู่ในปัจจุบัน