Chapter 9. I/O Devices Management

อินพุต/เอาต์พุตฮาร์ดแวร์ (I/O Hardware)

- นักวิชาการแต่ละสาขาจะการมอง อินพุต/เอาต์พุตฮาร์ดแวร์ ในมุมที่แตกต่างกัน
- นักวิศวกรรมไฟฟ้าเห็นว่า อินพุต/เอาต์พุตฮาร์ดแวร์ คือ ตัวฮาร์ดแวร์ที่นำมาประกอบกันได้แก่ มอเตอร์ สายไฟ ชุดจ่ายกระแสไฟฟ้า
- อินพุต/เอาต์พุตฮาร์ดแวร์ คือ โปรแกรมที่ใช้ติดต่ออุปกรณ์ต่าง ๆ ทำงานร่วมกัน
- เป็นโปรแกรมที่ใช้ควบคุมอุปกรณ์แต่ละชนิดจะมีความแตกต่างกันขึ้นอยู่กับการทำงาน

อุปกรณ์อินพุต/เอาต์พุต (I/O Devices)

- อุปกรณ์อินพุต/เอาต์พุต แบ่งออกเป็น 2 ประเภท
  - บล็อกดีไวซ์ (Block Devices)
  - คาร์เรกเตอร์ดีไวซ์ (Character Devices)

อุปกรณ์อินพุต/เอาต์พุต (I/O Devices)

- บล็อกดีไวซ์ (Block Devices)
  - เป็นอุปกรณ์ที่จัดเก็บข้อมูลได้โดยทำการเก็บเป็นบล็อกเช่น ดิสก์เก็ต ฮาร์ดดิสก์ เทป โดยทั่วไปขนาดของบล็อกประมาณ 512 ไบต์ ถึง 32,768 ไบต์
  - สามารถอ่าน/เขียนข้อมูลลงในแต่ละบล็อกได้โดยอิสระ
  - อุปกรณ์ประเภทที่เป็นบล็อกจำพวกดิสก์สามารถกำหนดตำแหน่งลงไปในแผ่นดิสก์ได้ ประกอบไปด้วย
    - ไซเรนเดอร์ (Cylinder)
    - เซ็กเตอร์ (Sector)

อุปกรณ์อินพุต/เอาต์พุต (I/O Devices)

- บล็อกดีไวซ์ (Block Devices)
  - แต่ละบล็อกที่บรรจุข้อมูลภายในหัวอ่านสามารถนำข้อมูลออกมาได้
  - อุปกรณ์ที่มีการแอ็กเซสโดยตรง (direct access storage device) ได้แก่ ดิสก์
  - ข้อมูลจะถูกจัดไว้เป็นกลุ่มในระดับบล็อกหรือเซ็กเตอร์
  - แต่ละกลุ่มจะมีแอ็ดเดรสของตัวเอง
  - การแอ็กเซสทำได้โดยกำหนดแอ็ดเดรสของข้อมูลกลุ่มนั้น ซึ่งจะทำให้ทราบว่าข้อมูลกลุ่มนั้นอยู่ที่ไหน

อุปกรณ์อินพุต/เอาต์พุต (I/O Devices)

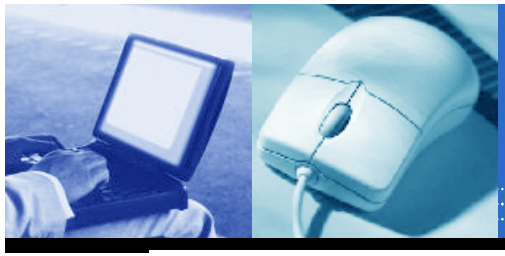
- บล็อกดีไวซ์ (Block Devices)
  - อุปกรณ์ที่มีการแอ็กเซสแบบลำดับ(serial access storage device) ได้แก่ เทป
  - ลักษณะของอุปกรณ์ประเภทนี้ การแอ็กเซสจะต้องเป็นไปตามลำดับ ตั้งแต่ต้นเทป เรียงไปจนถึงตำแหน่งที่ต้องการ
  - การเก็บข้อมูลจะเก็บเป็นกลุ่มๆ ไม่มีแอ็ดเดรสของแต่ละกลุ่ม การอ่านจะต้องอ่านเข้ามาทีละกลุ่ม

อุปกรณ์อินพุต/เอาต์พุต (I/O Devices)

- คาร์เรกเตอร์ดีไวซ์ (Character Devices)
  - เป็นอุปกรณ์ที่รับข้อมูลเป็นสาย
  - จำนวนข้อมูลขึ้นอยู่กับบัฟเฟอร์(Buffer) ที่มีแต่ไม่สามารถกำหนดตำแหน่งของบล็อกข้อมูลเหล่านี้ได้
  - อุปกรณ์ประเภทนี้ได้แก่ การ์ดเน็ตเวิร์คของเครื่องพิมพ์ สแกนเนอร์ และคีย์บอร์ด
  - อุปกรณ์ประเภทนี้มีการส่งข้อมูลเข้าออกเรียงเป็นลำดับก่อนหลัง
  - การแบ่งแยกข้อมูลทำได้โดยตรวจสอบลำดับของข้อมูล
  - เป็นอุปกรณ์ที่มีการจัดการได้ง่าย เพียงแต่จัดลำดับการรับ – ส่ง ข้อมูลที่ถูกต้อง เช่น ข้อมูลจากคีย์บอร์ดจะเรียงลำดับกันไป

อุปกรณ์อินพุต/เอาต์พุต (I/O Devices)

- อุปกรณ์ประเภทอื่น
  - อุปกรณ์ประเภทนี้ข้อมูลที่ส่งและรับไม่ขึ้นอยู่กับลำดับการส่ง
  - ต้องอาศัยข้อมูลเพิ่มเติมเพื่อที่จะแยกแยะข้อมูลแต่ละตัว
  - การจัดการอุปกรณ์ประเภทนี้ระบบปฏิบัติการจะต้องมีวิธีจัดการโดยเฉพาะขึ้นอยู่กับลักษณะของอุปกรณ์ชนิดนั้น
  - อุปกรณ์แต่ละประเภทมีความสามารถในการรับส่งข้อมูลในอัตราที่แตกต่างกัน
  - จอภาพของคอมพิวเตอร์จะต้องส่งให้ตรงตำแหน่งของตัวอักษรบนจอภาพซึ่งมีแอ็ดเดรสของตนเองที่แน่นอน



อุปกรณ์	ความเร็ว
คีย์บอร์ด	10 ไบต์/วินาที
เมาส์	100 ไบต์/วินาที
โมเด็ม 56K	7 กิโลไบต์/วินาที
แชนแนลโทรศัพท์	8 กิโลไบต์/วินาที
คู่สาย ISDN	16 กิโลไบต์/วินาที
เครื่องพิมพ์เลเซอร์	100 กิโลไบต์/วินาที
สแกนเนอร์	400 กิโลไบต์/วินาที
อีเธอร์เน็ต (Classic)	1.25 เมกะไบต์/วินาที
USB (Universal Serial Bus)	1.5 เมกะไบต์/วินาที
ดิจิตอลแคมคอร์ดเดอร์	4 เมกะไบต์/วินาที

ลำดับความเร็วในการรับ-ส่งข้อมูลของอุปกรณ์จากช้าไปเร็ว

อุปกรณ์	ความเร็ว
ดิสก์ IDE	5 เมกะไบต์/วินาที
ซีดีรอม 40X	6 เมกะไบต์/วินาที
อีเธอร์เน็ต (Fast)	12.5 เมกะไบต์/วินาที
บัส ISA	16.7 เมกะไบต์/วินาที
ดิสก์ EIDE (ATA-2)	16.7 เมกะไบต์/วินาที
FireWire (IEEE 1394)	50 เมกะไบต์/วินาที
มอนิเตอร์ XGA	60 เมกะไบต์/วินาที
เน็ตเวิร์ค SONET OC-12	78 เมกะไบต์/วินาที
ดิสก์ SCSI Ultra 2	80 เมกะไบต์/วินาที
อีเธอร์เน็ต (Gigabit)	125 เมกะไบต์/วินาที
เทป Ultrium	320 เมกะไบต์/วินาที
บัส PCI	528 เมกะไบต์/วินาที
Sun Gigaplane XB backplane	20 กิกะไบต์/วินาที

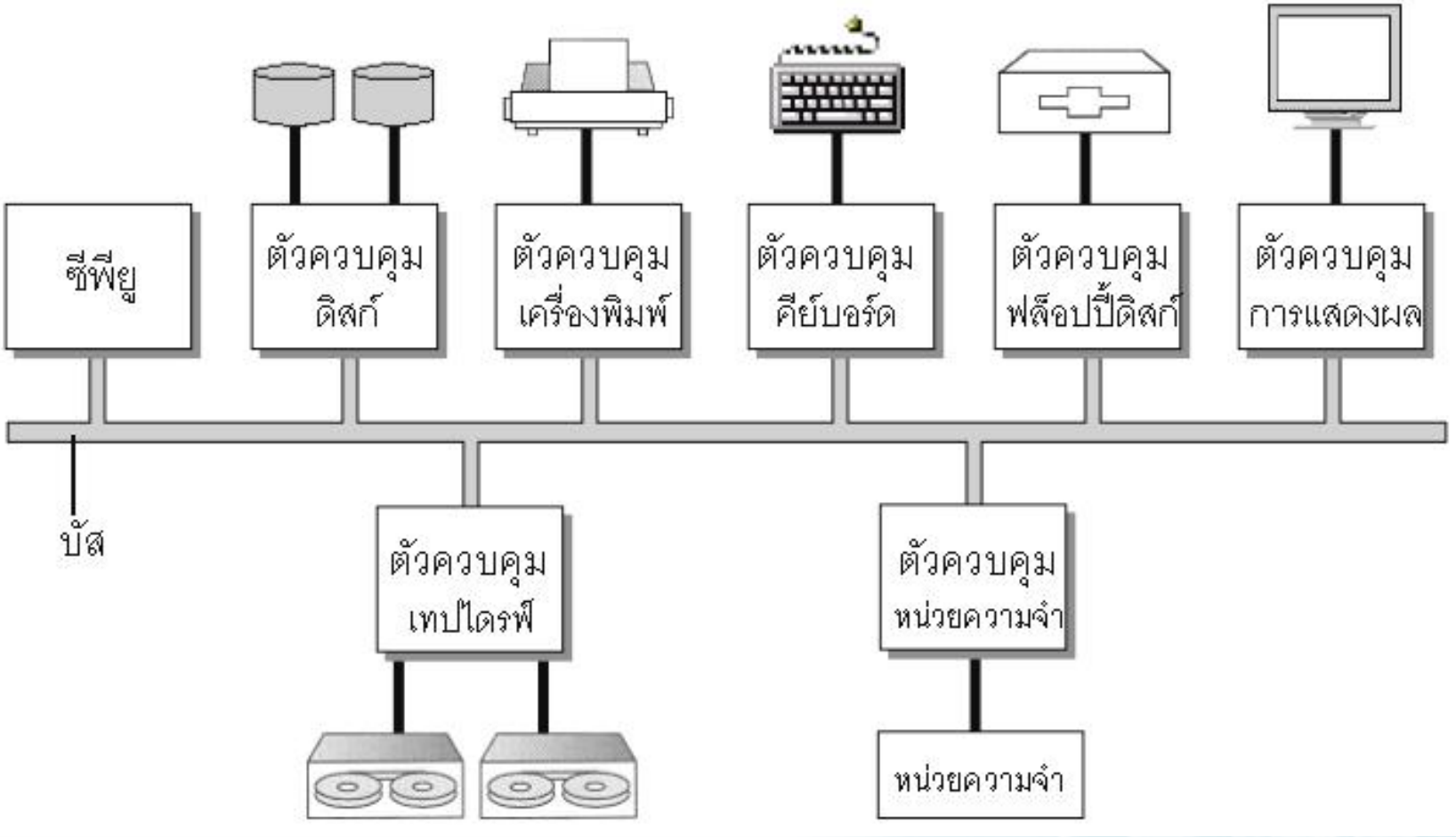
ลำดับความเร็วในการรับ-ส่งข้อมูลของอุปกรณ์จากช้าไปเร็ว(ต่อ)

ตัวควบคุมอุปกรณ์ (Device Controllers)

- อุปกรณ์แต่ละประเภทจะประกอบด้วยกลไกต่างๆ และส่วนที่เป็นวงจรอีเล็กทรอนิกส์ ซึ่งประกอบเข้าเป็นตัวชิพ หรือประกอบรวมกันบนแผ่นเซอร์กิตบอร์ด (circuit board) ที่ถูกเรียกว่า “ตัวควบคุมอุปกรณ์” (device controller) หรืออะแดปเตอร์ (adapter)
- สำหรับบอร์ดและการ์ด ที่ประกอบเป็นคอมพิวเตอร์นั้นเรียกว่า กลไกบนบอร์ดคอมพิวเตอร์ ที่สามารถต่อการ์ดต่างๆ ได้เป็นจำนวนมาก
- ช่องที่เสียบเข้ากับบอร์ดเรียกว่า สล็อต (slot) สำหรับคอมพิวเตอร์พีซี (PC) สล็อตเหล่านี้จะมีชื่อเรียกต่าง ๆ กัน เช่นISA PCI AGP

ตัวควบคุมอุปกรณ์ (Device Controllers)

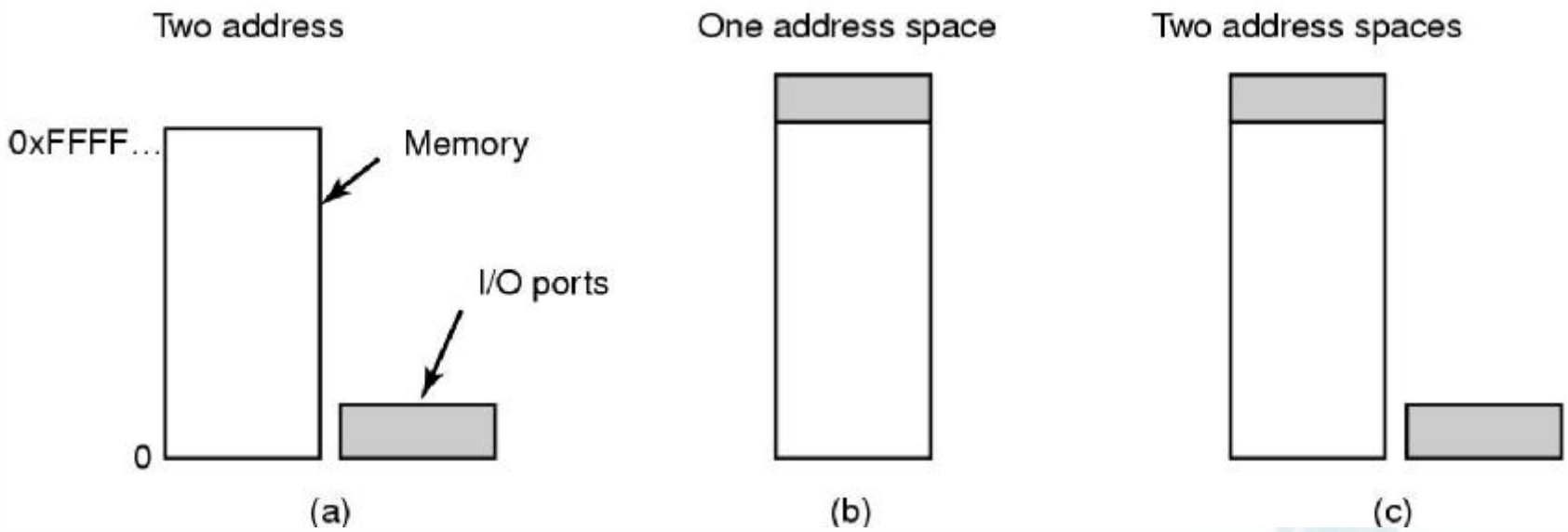
- การ์ดควบคุมอุปกรณ์แต่ละชนิดที่เสียบลงบนบอร์ดจะถูกต่อสายไปยังอุปกรณ์นั้นๆ แต่ละการ์ดสามารถควบคุมอุปกรณ์ได้ตั้งแต่1ชิ้น 2,4,8 ชิ้น
- การติดต่อระหว่างการ์ดควบคุมกับอุปกรณ์มักเป็นในรูปของภาษาเครื่องการโอนถ่ายข้อมูลจะกระทำในระดับบิต
- หน้าที่ของตัวควบคุมนอกจากจะคอยตรวจสอบข้อผิดพลาด แล้วยังมีหน้าที่ในการแปลข้อมูลจากรูปของสายข้อมูลไปเป็นบล็อกเพื่อนำไปจัดเก็บลงในสื่อข้อมูลหรือจากสื่อข้อมูลไปลงบนหน่วยความจำ
- ตัวอย่างของตัวควบคุมจอภาพหรือการ์ดจอจะรับข้อมูลในรูปของสายข้อมูลเพื่อนำมาใช้กับจอ CRT เพื่อสร้างสัญญาณข้อมูล โดยมีการแยกสัญญาณแต่ละตำแหน่งบนจอภาพที่เรียกว่าพิกเซล(pixel)



โครงสร้างระบบคอมพิวเตอร์ด้านอินพุต/เอาต์พุต

อินพุต/เอาต์พุตที่ใช้การแมพหน่วยความจำ (Memory mapped I/O)

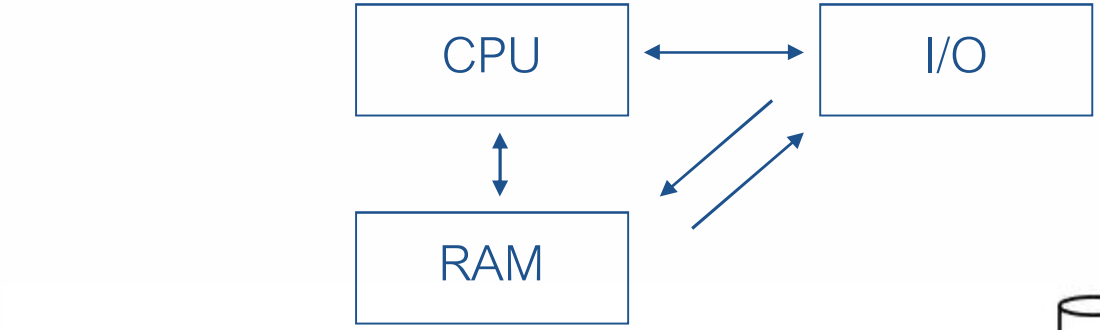
- ตัวควบคุมแต่ละตัวจะมีรีจิสเตอร์ที่ใช้ติดต่อกับซีพียูโดยตรง
- ระบบปฏิบัติการ จะเป็นตัวประสานการทำงานระหว่างอุปกรณ์ต่างๆ กับซีพียู แล้วแต่การร้องขอและความจำเป็นของอุปกรณ์นั้น ๆ
- ขณะที่ซีพียูทำการอ่านข้อมูลจากรีจิสเตอร์ ระบบปฏิบัติการที่ดีจะเป็นตัวกำหนดช่วงเวลาที่ติดต่อให้สอดคล้องและเหมาะสมกับสถานะของอุปกรณ์ในขณะนั้น
- อุปกรณ์หลายตัวจะมีบัฟเฟอร์ที่คอยทำหน้าที่เก็บข้อมูลเมื่อมีการติดต่อมายังอุปกรณ์นั้น ๆ โดยมีระบบปฏิบัติการเป็นตัวประสาน ตัวอย่างบัฟเฟอร์ คือ **วีดีโอแรม (Video RAM)**

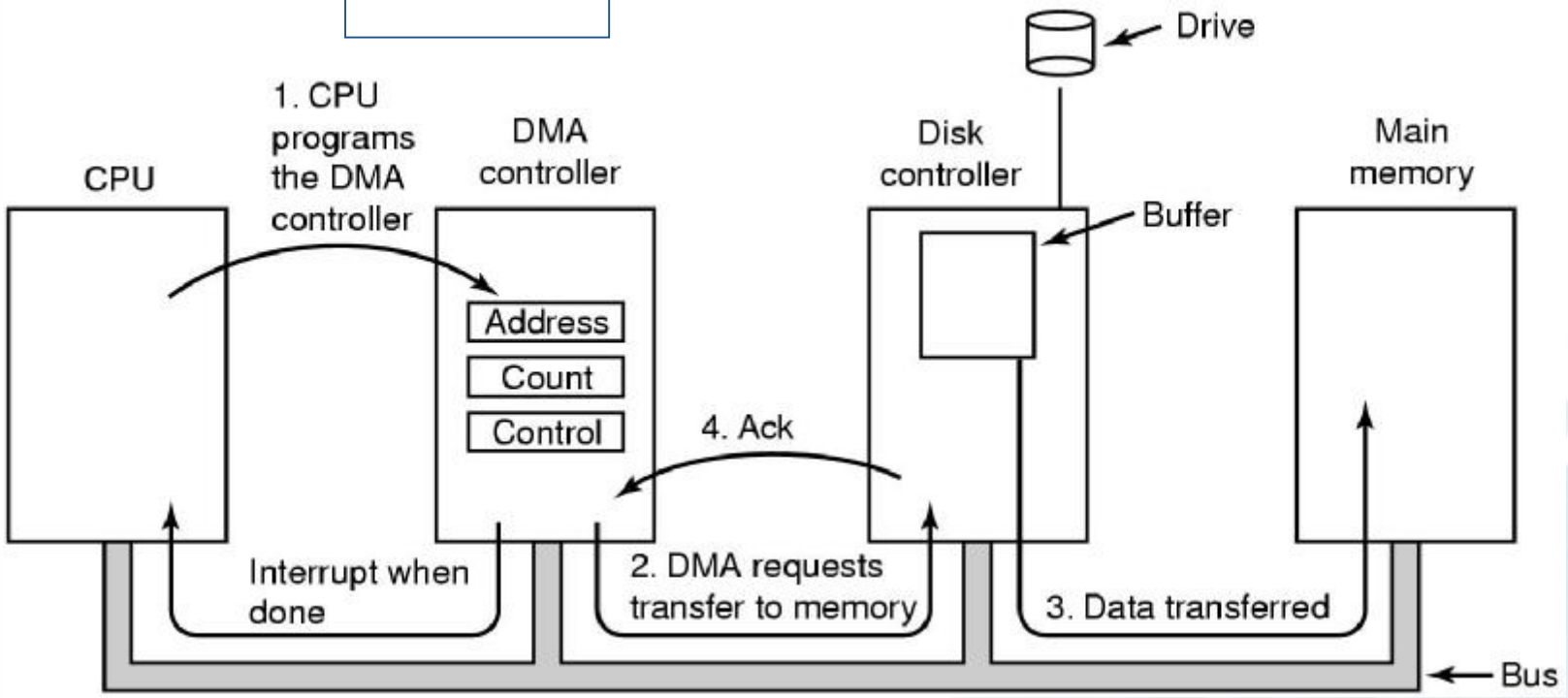


- Separate I/O and memory space
- Memory-mapped I/O
- Hybrid (ในเครื่อง Pentium จะใช้แบบนี้ โดยสงวนแอ็ดเดรสช่วง 640K - 1M เพื่อใช้เป็นบัฟเฟอร์ นอกเหนือจากพอร์ต I/O ช่วง 0 - 64K

การแอ็กเซสข้อมูลโดยตรง (DMA : Direct Memory Access)

- DMA เป็นการรับ-ส่งข้อมูลจากหน่วยความจำกับอุปกรณ์โดยตรงไม่ต้องผ่านซีพียู
- การรับ-ส่งข้อมูลได้รวดเร็วไม่เปลืองเวลาของซีพียู เพราะการรับ-ส่งข้อมูลแบบ DMA อาศัยแชนแนลหรือตัวควบคุม DMA (DMA controller)
- แชลแนล ทำหน้าที่แทนซีพียูเมื่อต้องการรับส่งข้อมูลแบบDMA
- แชลแนลจะส่งสัญญาณให้ ซีพียูรับรู้ ซีพียูจะสั่งให้แชนแนลทำรูทีนควบคุมการส่งข้อมูล
- จากนั้นซีพียูจะไปทำงานอื่นต่อและเมื่อการทำ DMA เสร็จสิ้นลง แชลแนลจะส่งสัญญาณบอกซีพียูได้รับรู้อีกครั้งว่าทำ DMA เสร็จแล้ว





ขั้นตอนการเกิดDMA

การแอ็กเซสข้อมูลโดยตรง (DMA : Direct Memory Access)

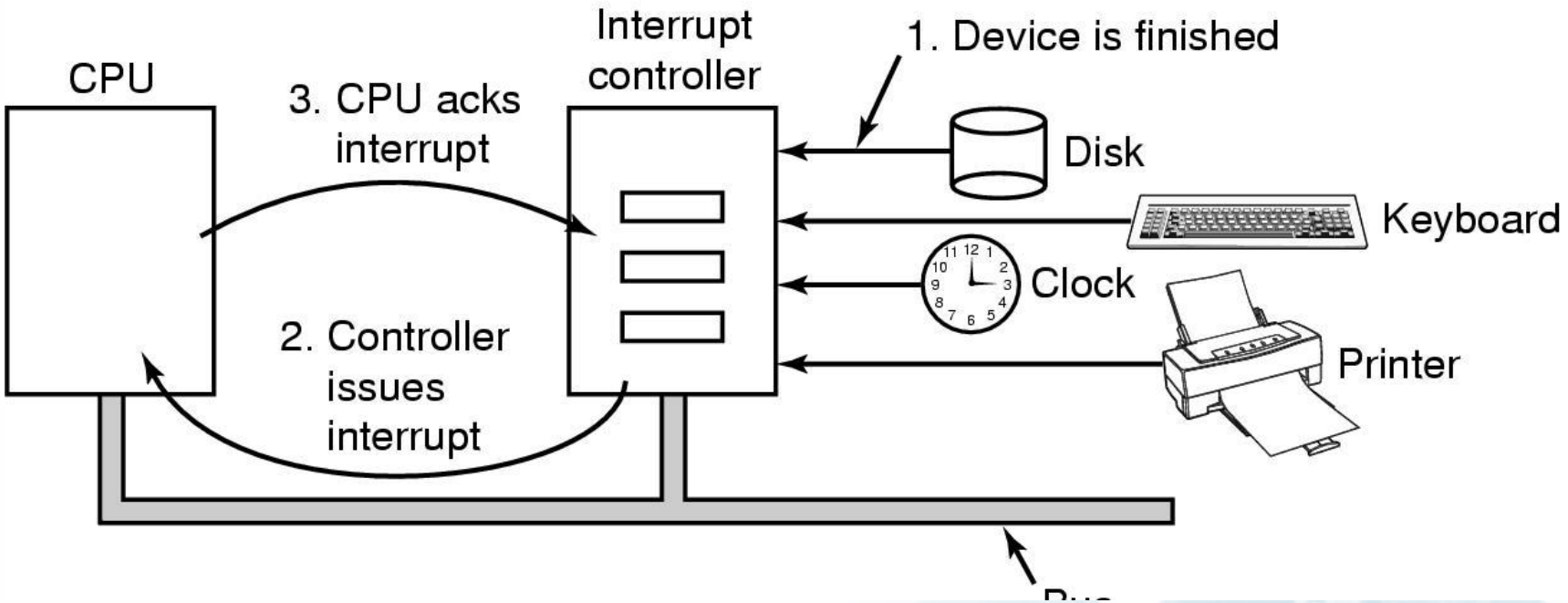
- อุปกรณ์ที่ใช้ตัวควบคุม DMA ในการรับ-ส่งข้อมูลโดยอิสระจากตัวซีพียู อุปกรณ์เหล่านี้ประกอบด้วยรีจิสเตอร์ที่ใช้ในการเก็บข้อมูลในระหว่างที่ทำการรับ-ส่งข้อมูลผ่านทางซีพียู
- การโอนถ่ายข้อมูลแต่ละครั้งจะทำครั้งละไบต์ หรือเวิร์ด หรือหลาย ๆ เวิร์ด ขึ้นอยู่กับจำนวนรีจิสเตอร์ที่มีอยู่ของอุปกรณ์นั้น ๆ
- นอกจากรีจิสเตอร์แล้ว อุปกรณ์เหล่านี้ยังมีรีจิสเตอร์ที่คอยทำหน้าที่ควบคุมการรับส่งข้อมูลผ่านทางแซลแนลต่าง ๆ อีกด้วย

การแอ็กเซสข้อมูลโดยตรง (DMA : Direct Memory Access)

- ซีพียูที่รันงานอื่น ๆ อยู่และมีการกระทำ DMA ขึ้นพร้อม ๆ กันในช่วงที่มีการรับ-ส่งข้อมูลระหว่างหน่วยความจำกับแชลแนล แชลแนลจำเป็นต้องใช้บัสข้อมูลและบัสแอ็ดเดรสของหน่วยความจำกับซีพียู
- กรณีการเลี่ยงช่วงเวลาที่ซีพียูใช้บัสข้อมูลและบัสแอ็ดเดรสของหน่วยความจำไม่ให้ตรงกัน ถ้าซีพียูต้องการใช้บัสทั้งสองDMA จะต้องหยุดทันทีเป็นการชั่วคราว เมื่อซีพียูทำงานเสร็จDMA ถึงจะกลับมาใช้บัสได้ตามเดิม ลักษณะที่ DMA ใช้บัสทั้งสองในขณะที่ซีพียูไม่ใช้นี้เรียกว่า “การขโมยรอบเวลา (cycle stealing)”

การอินเทอร์รัพต์ซีพียู (Interrupts Revisited)

- การอินเทอร์รัพต์ในระบบคอมพิวเตอร์ขณะที่ซีพียูกำลังทำงาน สามารถเกิดได้ตลาดเวลา ในระดับฮาร์แวร์เมื่ออุปกรณ์ใดๆ ทำงานที่ส่งเข้ามาเสร็จสิ้นแล้ว มันก็จะเกิดการอินเทอร์รัพต์ขึ้น
- สัญญาณอินเทอร์รัพต์จะถูกส่งผ่านไปตามบัสข้อมูลโดยมีอุปกรณ์ตรวจจับสัญญาณดังกล่าวเป็นชิพตัวหนึ่งที่เรียกว่า ตัวควบคุมอินเทอร์รัพต์
- ขณะที่ตัวควบคุมอินเทอร์รัพต์ตรวจดูว่ามีอินเทอร์รัพต์เข้ามาสัญญาณเดียวสัญญาณนี้ก็จะถูกส่งไปยังซีพียูทันทีเพื่อทำงานตามที่ร้องขอ
- แต่ถ้าสัญญาณที่ส่งเข้ามามีมากกว่าหนึ่งสัญญาณ ตัวควบคุมอินเทอร์รัพต์จะตรวจสอบความสำคัญของสัญญาณ แล้วจัดลำดับก่อนหลัง



การเกิดอินเทอร์รัพต์

การอินเทอร์รัพต์ซีพียู (Interrupts Revisited)

- ตารางอินเทอร์รัพต์เวกเตอร์ (Interrupt Vector) เป็นตารางที่เก็บรวบรวมแอ็ดเดรสต่างๆ ของสัญญาณอินเทอร์รัพต์ที่ร้องขอเข้ามาเพื่อขอรับบริการจากซีพียู
- การให้บริการของซีพียูจะถูกควบคุมโดยระบบปฏิบัติการเป็นตัวโหลดข้อมูลแอ็ดเดรสต่าง ๆ ตามที่บันทึกในอินเทอร์รัพต์เวกเตอร์แล้วทำงานไปตามวงรอบการทำงาน (machine cycle)
- คอมพิวเตอร์รุ่นเก่าบางรุ่น อาจไม่มีชิพที่ทำหน้าที่เป็นศูนย์รวมการรับสัญญาณอินเทอร์รัพต์จากอุปกรณ์ต่าง ๆ โดยตรง
- ดังนั้นตัวควบคุมอุปกรณ์แต่ละตัวจะต้องทำหน้าที่ร้องขออินเทอร์รัพต์ไปยังซีพียูโดยตรง

การอินเทอร์รัพต์ซีพียู (Interrupts Revisited)

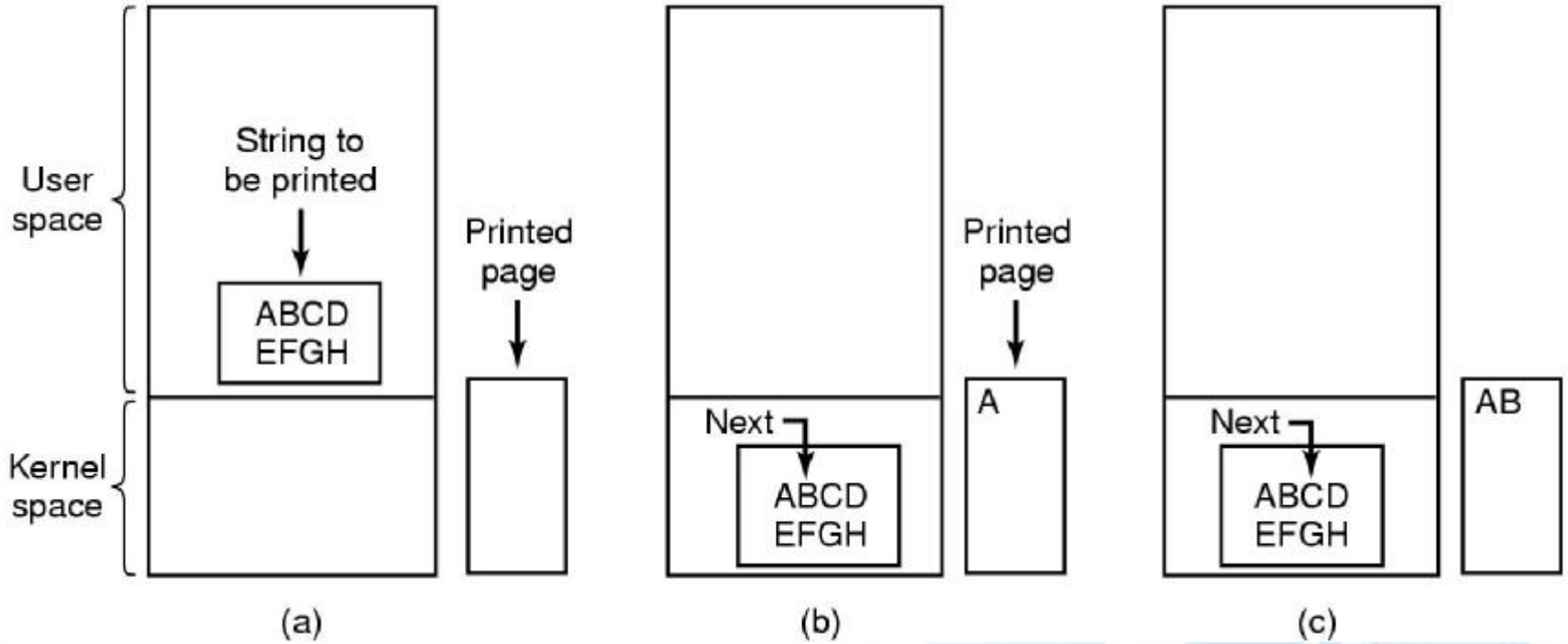
- ปัญหาของการเก็บแอ็ดเดรสเดิมเพื่อไปให้บริการอินเทอร์รัพต์ดังกล่าวคือตัวควบคุมอินเทอร์รัพต์ไม่สามารถรู้เลยว่าอินเทอร์รัพต์ที่ร้องขออันดับแรกได้รับบริการเสร็จหรือยัง
- ระบบแฮงค์ (Hang) ได้เมื่ออินเทอร์รัพต์ที่ร้องขอต้องรอคอยเป็นเวลานาน
- วิธีการแก้ การแฮงค์ (Hang) โดยใช้สแต็กในการเก็บแอ็ดเดรสต่าง ๆ ที่ซีพียูย้ายไปให้บริการ
- การใช้สแต็กยังคงมีปัญหาในกรณีที่มีการผิดหน้า(page fault) ระหว่างที่ซีพียูให้บริการอินเทอร์รัพต์ ทำให้โปรเซสไม่สามารถนำข้อมูลเก็บไว้ในหน่วยความจำได้ทำให้ระบบไม่สามารถบันทึกสถานะของเครื่องได้

หลักการอินพุต/เอาต์พุตซอฟต์แวร์ (I/O Software)

- หลักของอินพุต/เอาต์พุตซอฟต์แวร์
  - อุปกรณ์แต่ละชนิดเป็นอิสระต่อกันไม่เจาะจงประเภทหรือ ยี่ห้องของอุปกรณ์
  - ตัวอย่างเช่น ในการอ่านข้อมูล โปรแกรมที่เขียนขึ้นต้องสามารถอ่านข้อมูลได้ทั้งในฟล็อปปี้ ฮาร์ดดิสก์ หรือกระทั่งข้อมูลบนซีดีรอม โดยไม่ต้องปรับปรุงแก้ไขโปรแกรมแต่อย่างใด
  - สามารถตรวจสอบข้อผิดพลาดที่เกิดขึ้นได้โดยอัตโนมัติ หรือสามารถตรวจสอบหาสาเหตุของข้อผิดพลาดได้และพร้อมกับบอกแนวทางในการแก้ไขมาพร้อมกันด้วย
  - ในการส่งผ่านข้อมูลที่มีทั้งแบบวิธีการบล็อก (blocking) และการใช้อินเทอร์รัพต์ ซึ่งส่วนใหญ่จะใช้อินเทอร์รัพต์

หลักการอินพุต/เอาต์พุตซอฟต์แวร์ (I/O Software)

- Programmed I/O
  - เป็นวิธีที่ง่ายที่สุดใน 3วิธี คือ
    - Interrupt-driven I/O
    - I/O using DMA
    - Programmed I/O
  - การจัดการข้อมูลของ Programmed I/O สามารถแสดงได้จากตัวอย่างการส่งข้อมูลออกพิมพ์ทางเครื่องพิมพ์
  - ข้อมูลที่พิมพ์ขั้นแรกจะถูกเก็บในบัฟเฟอร์ของผู้ใช้งานก่อน หลังจากนั้นข้อมูลจะถูกส่งไปที่ kernel space
  - ในขณะที่ข้อมูลถูกส่องออกพิมพ์ถ้าเครื่องพิมพ์ว่างก็จะพิมพ์ทันทีแต่ถ้าเครื่องพิมพ์ไม่ว่าง ก็จะรอจนกว่าเครื่องพิมพ์ว่าง
  - จากนั้นจะทำการก็อปปี้ข้อมูลไปทีละตัวและสั่งพิมพ์ทางเครื่องพิม



Steps in printing a string

หลักการอินพุต/เอาต์พุตซอฟต์แวร์ (I/O Software)

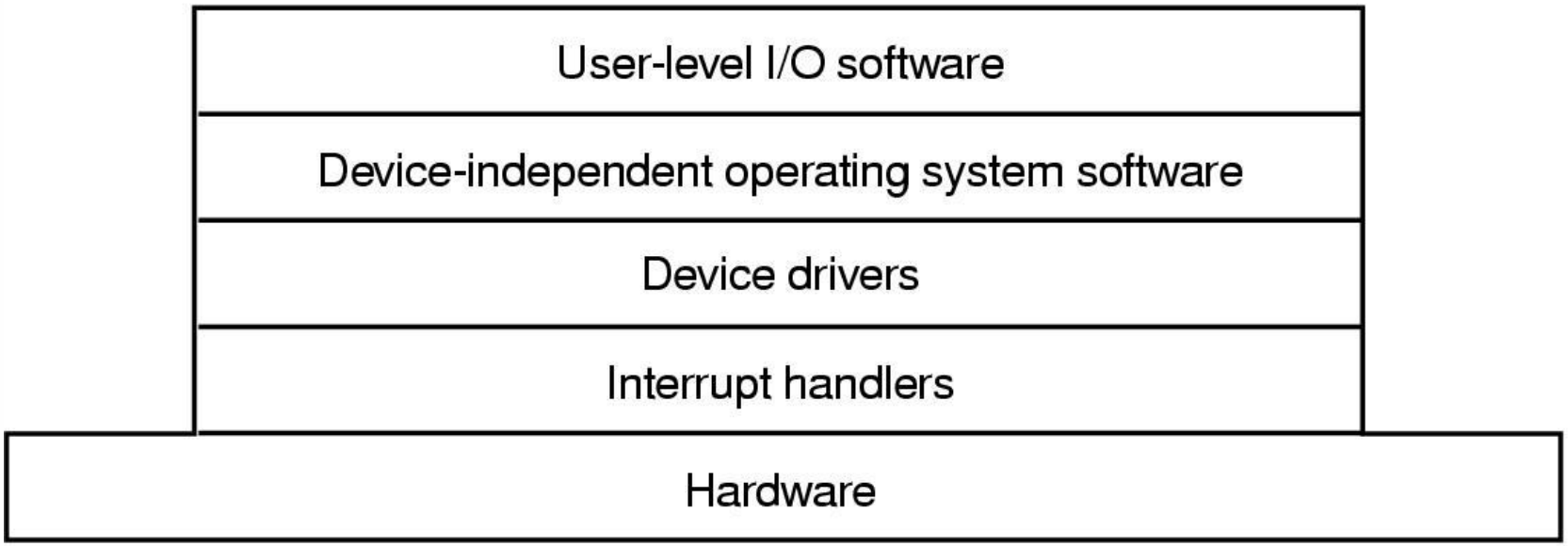
- Interrupt – Driven I/O
  - เกิดขึ้นในกรณีของการพิมพ์ข้อมูลผ่านทางเครื่องพิมพ์
  - ขั้นตอนการพิมพ์ข้อมูล 100 ตัวอักษร ซีพียูทำหน้าที่ส่งตัวอักษรไปยังเครื่องพิมพ์ทีละ 1 อักษร
  - จากนั้นจะคอยจนกว่าพิมพ์เสร็จแล้วส่งต่อไปจนกระทั่งครบทั้ง 100 ตัว
  - หนทางที่จะช่วยรักษาเวลาของซีพียูได้โดยในขณะที่ซีพียูรอการทำงานจากเครื่องพิมพ์ ก็คือหางานให้ซีพียูได้ทำงาน
  - ช่วงเวลาการพิมพ์ทีละตัวเป็นเวลาที่น้อยเกินไป
  - การจัดระบบจัดสรรบัฟเฟอร์ให้จำนวนหนึ่งจะทำให้ได้ช่วงเวลาที่มากกว่า
  - การให้งานกับซีพียูจะต้องมีตัวควบคุมการอินเทอร์รัพต์เพื่อร้องขออินเทอร์รัพต์ แล้วซีพียูจึงให้บริการอินเทอร์รัพต์นั้น

หลักการอินพุต/เอาต์พุตซอฟต์แวร์ (I/O Software)

- I/O Using DMA
  - การรับ-ส่งข้อมูลแบบ DMA (Direct Memory Access) จำเป็นต้องอาศัยแชนแนล หรือตัวควบคุม DMA (DMA controller) แชนแนลจะทำหน้าที่แทนซีพียูเมื่อต้องการรับ-ส่งข้อมูลแบบ DMA
  - แชนแนลจะทำหน้าที่ส่งสัญญาณไปบอกซีพียูให้รับรู้ และพร้อมกับสั่งแชนแนลให้ทำรูทีนควบคุมการรับ-ส่งข้อมูล เพื่อให้ซีพียูไปทำงานอื่นก่อน
  - การทำงานของซีพียูบางครั้งจำเป็นต้องใช้บัสรับ-ส่งข้อมูล ถ้าเป็นช่วงเวลาเดียวกับเวลาที่เกิด DMA (เป็นไปได้น้อยมาก) กรณีนี้จะเลี่ยงการเกิด DMA โดยให้ซีพียูทำงานไปตามปกติ เมื่อทำงานเสร็จDMA จึงใช้บัสข้อมูล
  - โอกาสที่ DMA เกิดขึ้น และใช้บัสในการรับ-ส่งข้อมูลจะใช้สัญญาณนาฬิกา (Clock) เข้ามาช่วย เราเรียกช่วงเวลาดังกล่าวนี้ว่า“การขโมยรอบเวลา”

อินพุต/เอาต์พุตซอฟต์แวร์เลเยอร์ (I/O Software Layers)

- Interrupt Handlers หรือการแก้ไขเมื่อเกิดอินเทอร์รัพต์
  - เลเยอร์แรกเป็นเลเยอร์ที่ติดต่อกับตัวฮาร์ดแวร์โดยตรง
  - การติดต่อกับอุปกรณ์ใด ๆ จะใช้โปรแกรมเป็นตัวติดต่อ
  - ถ้ามีการใช้อินเทอร์รัพต์ถือว่าการขัดจังหวะงานที่กำลังทำอยู่ของอุปกรณ์นั้น จะต้องมีการกำหนดขั้นตอนการปฏิบัติตามลำดับ
  - ซึ่งลำดับขั้นตอนดังกล่าวในแต่ละอุปกรณ์จะแตกต่างกัน
  - เมื่อเกิดอินเทอร์รัพต์ซีพียูจะถูกพักงานไว้ก่อน และทำการบันทึกสถานภาพขณะนั้นไว้ก่อน
  - ย้ายไปทำการควบคุมที่อุปกรณ์ที่ร้องขออินเทอร์รัพต์นั้น
  - เมื่องานเสร็จ จะกลับทำงานเดิมตามสถานะที่บันทึกไว้



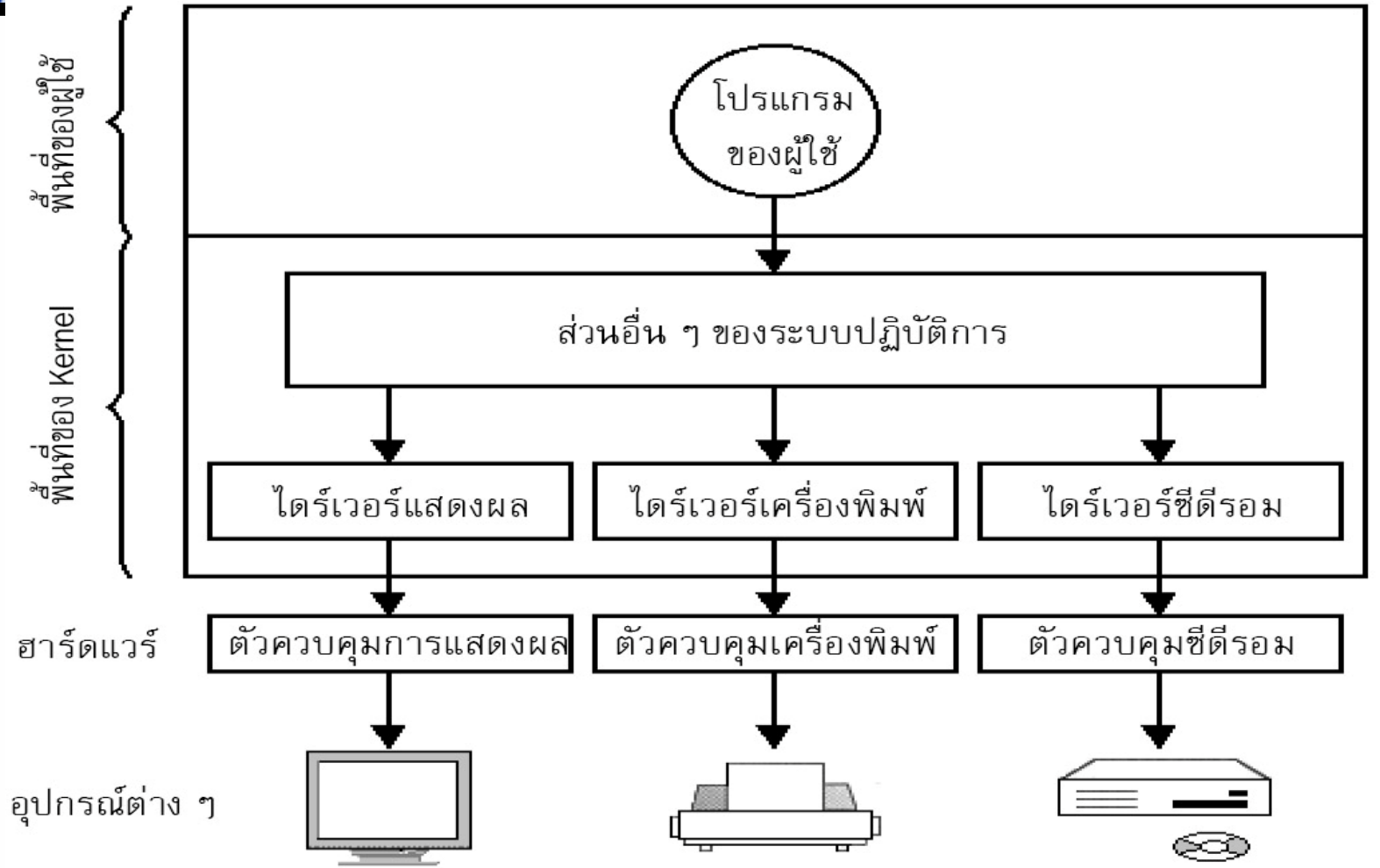
Layers of the I/O Software System

อินพุต/เอาต์พุตซอฟต์แวร์เลเยอร์ (I/O Software Layers)

- ดีไวซ์ไดร์เวอร์ (Device Drivers)
  - อุปกรณ์แต่ละชนิดมีคุณสมบัติในการติดต่อข้อมูล รวมทั้งความเร็วในการรับส่งข้อมูลที่แตกต่างกัน
  - ถ้าให้ระบบปฏิบัติการทำหน้าที่ควบคุมและติดต่ออุปกรณ์เหล่านั้น จะทำให้ระบบปฏิบัติการมีขนาดใหญ่มาก และต้องอัปเดทระบบปฏิบัติการบ่อย ๆ
  - ผู้ออกแบบระบบปฏิบัติการจึงเห็นควรแยกการควบคุมอุปกรณ์ออกจากระบบปฏิบัติการ
  - โปรแกรมที่แยกตัวออกมาเพื่อทำหน้าที่ควบคุมและติดต่ออุปกรณ์เรียกว่า “ดีไวซ์ไดร์เวอร์” (Device Drivers)
  - อุปกรณ์แต่ละชนิดจะมีดีไวซ์ไดร์เวอร์เป็นของมันเองซึ่งจะแตกต่างกับชนิดอื่น

อินพุต/เอาต์พุตซอฟต์แวร์เลเยอร์ (I/O Software Layers)

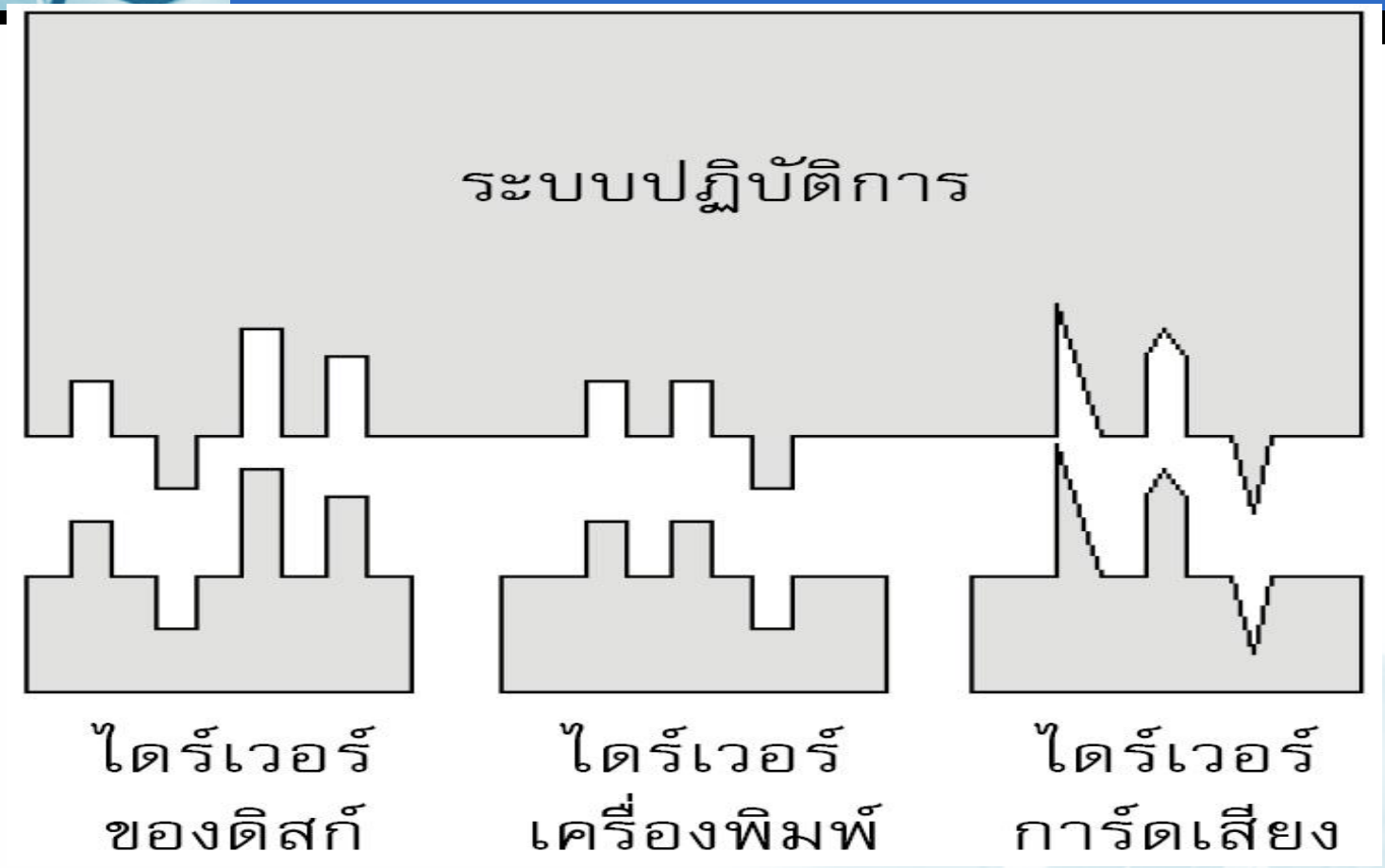
- ดีไวซ์ไดร์เวอร์ (Device Drivers)
  - ถึงแม้จะเป็นอุปกรณ์ชนิดเดียวกันก็ใช้ไดร์เวอร์คนละตัวกัน เช่น เครื่องพิมพ์ของ Epson จะใช้ไดร์เวอร์ของ NEC ไม่ได้ (แม้แต่คนละรุ่นก็ใช้คนละตัวกัน)
  - ระบบปฏิบัติการที่ติดต่อกับอุปกรณ์ชนิดใด จะทำการติดต่อผ่านทางไดร์เวอร์ชนิดนั้น แล้วไดร์เวอร์ก็จะติดต่อกับอุปกรณ์จริง ๆ อีกทีหนึ่ง
  - การติดต่อของระบบปฏิบัติการกับอุปกรณ์ไม่ว่าจะเป็นไดร์เวอร์ชนิดใด ๆ ก็ตามจะมีรูปแบบเดียวกันทั้งหมด
  - การติดตั้งอุปกรณ์จึงทำได้ง่าย โดยการเสียบปลั๊กและติดตั้งโปรแกรมไดร์เวอร์ ซึ่งในไดร์เวอร์ยังมีตัวควบคุมอุปกรณ์(Device Controller)
  - ปัจจุบันฟีเจอร์ที่เรียกว่าPnP (plug and Play) ช่วยให้การติดตั้งอุปกรณ์ชิ้นใหม่ทำได้สะดวกและรวดเร็วมากยิ่งขึ้น



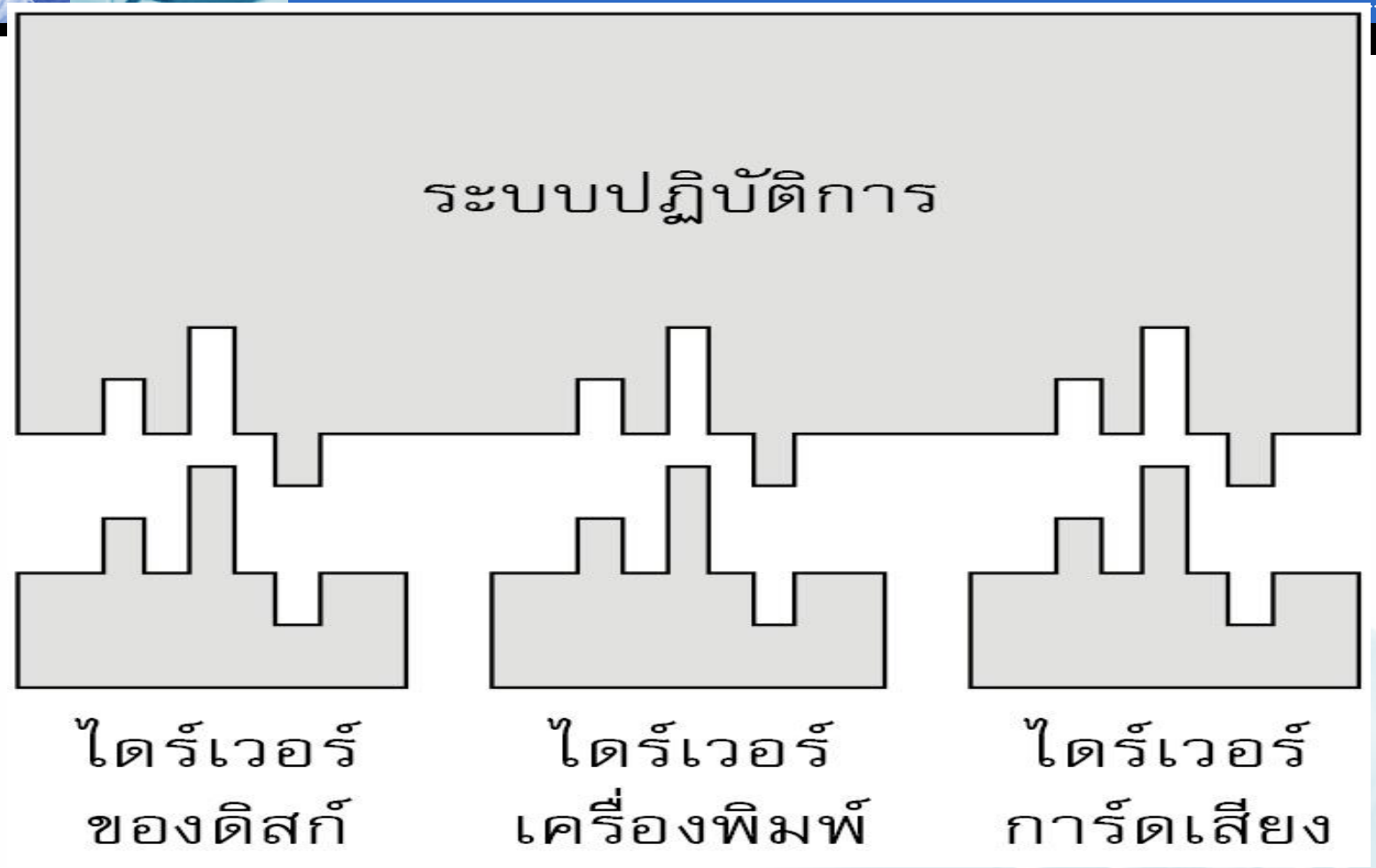
รูปภาพแสดงตำแหน่งของดีไวซ์ไดร์เวอร์ที่อยู่กลางระหว่างอุปกรณ์กับระบบปฏิบัติการโดยเชื่อมด้วยบัส

อินพุต/เอาต์พุตซอฟต์แวร์เลเยอร์ (I/O Software Layers)

- Device Independent I/O Software
  - การอินเทอร์เฟซที่เป็นลักษณะเฉพาะของดีไวซ์ไดร์เวอร์ : ความต้องการของระบบปฏิบัติการคือต้องการให้ระบบ สามารถติดต่อกับอุปกรณ์ด้วยมาตรฐานเดียวกัน ไม่ว่าจะติดต่อกับดิสก์ เครื่องพิมพ์ คีย์บอร์ด



รูปภาพ (ก) แสดงการติดต่อระหว่างอุปกรณ์กับระบบด้วยไดร์เวอร์ที่แตกต่างกัน เมื่อระบบต้องการติดต่อกับอุปกรณ์อื่น



รูปภาพ (ข) แสดงการติดต่อระหว่างอุปกรณ์กับระบบนั้นใช้การติดต่อที่มีมาตรฐานเดียวกัน

อินพุต/เอาต์พุตซอฟต์แวร์เลเยอร์ (I/O Software Layers)

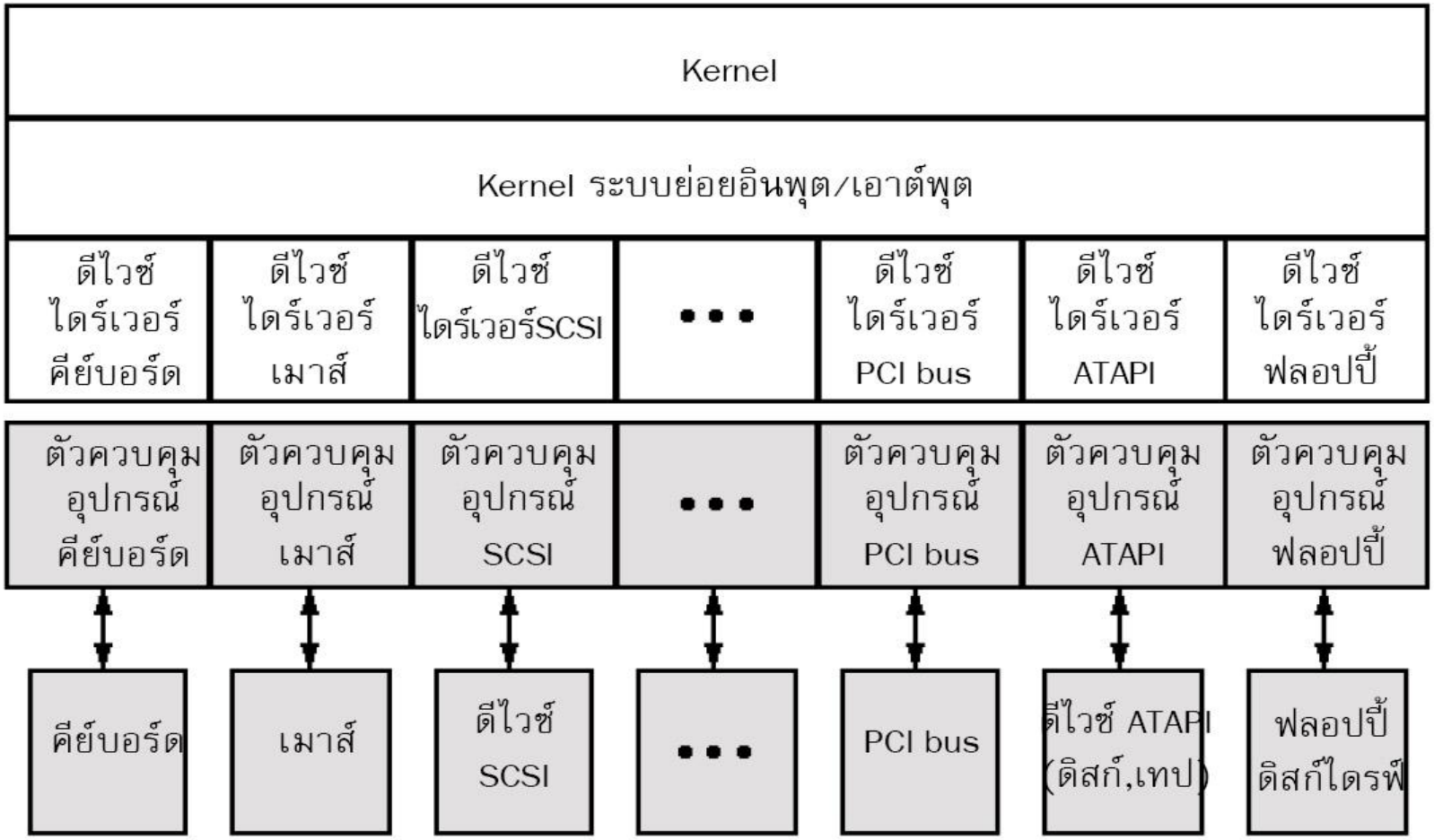
- Device Independent I/O Software
  - การจัดบัฟเฟอร์ : หน่วยความจำบัฟเฟอร์ถือว่ามีความสำคัญในการติดต่อระหว่างอุปกรณ์ เนื่องจากความสามารถของอุปกรณ์แต่ละชนิดมีความรวดเร็วในการรับ-ส่งข้อมูลที่แตกต่างกัน
  - การรายงานข้อผิดพลาด : ระบบปฏิบัติการจะมีฟังก์ชันที่ทำหน้าที่ในการตรวจสอบและรายงานข้อผิดพลาดที่เกิดขึ้นได้
  - การผูกและปล่อยอุปกรณ์ : เช่น CD-ROM ประกอบด้วย 2 ฟังก์ชันที่ทำหน้าที่ในการอ่านข้อมูลจากสื่อ และทำหน้าที่ในการเขียนข้อมูล

อินพุต/เอาต์พุตซอฟต์แวร์เลเยอร์ (I/O Software Layers)

- User Space I/O Software
  - จัดเนื้อที่สำหรับเก็บค่าข้อมูลต่าง ๆ เช่น Library Procedure เพื่อเชื่อมกับโปรแกรม
  - และสามารถทำเป็น สพูลลิ่ง(Spooling เพื่อพักข้อมูลไว้ชั่วขณะในระหว่างที่รอการส่งต่อไปยังอุปกรณ์)

แอปพลิเคชันอินพุต/เอาต์พุตอินเทอร์เฟซ (Application I/O Interface)

- การติดต่อระหว่างตัวควบคุมอุปกรณ์กับระบบย่อยอินพุต/เอาต์พุต (I/O Subsystem) ซึ่งเป็นส่วนหนึ่งของKernel จะใช้ไดร์เวอร์เป็นตัวควบคุม
- ในอุปกรณ์แต่ละประเภทจะมีไดร์เวอร์เป็นของตัวเอง
- ในเลเยอร์ของไดร์เวอร์ (Drivice Driver Layer) จะถูกปิดซ่อนไว้ระหว่างระบบย่อยอินพุต/เอาร์พุตกับตัวควบคุมอุปกรณ์



โครงสร้าง Kernel ของอินพุต/เอาร์พุต

แอปพลิเคชันอินพุต/เอาต์พุตอินเทอร์เฟซ (Application I/O Interface)

- การแยกประเภทต่างๆของอุปกรณ์แบ่งตามลักษณะการส่งข้อมูลได้ดังนี้
  - อุปกรณ์ที่มีการส่งข้อมูลครั้งละตัวอักษร กับครั้งละบล็อกของข้อมูล
  - การเข้าถึงข้อมูลของอุปกรณ์แบบเรียงลำดับ (Sequential access) หรือการเข้าถึงแบบสุ่ม (Random access)
  - อุปกรณ์ที่มีการจัดเตรียมข้อมูลแบบกำหนดช่วงเวลาได้ (Synchronous) กับกำหนดช่วงเวลาไม่ได้ (Asynchronous)
  - อุปกรณ์ที่สามารถใช้ร่วมกันในช่วงเวลาเดียวกันได้ กับอุปกรณ์ที่ต้องใช้ต่างช่วงเวลากัน
  - ความเร็วที่แตกต่างกันของอุปกรณ์แต่ละประเภท
  - ทิศทางในการถ่ายโอนข้อมูลของอุปกรณ์

Aspect	Variation	Example
Data-transfer mode	Character Block	เทอร์มินอล ดิสก์
Access method	Sequential Random	โมเด็ม ซีดีรอม
Transfer schedule	Synchronous Asynchronous	เทป คีย์บอร์ด
Sharing	Dedicate Sharable	เทป คีย์บอร์ด
Device speed	Latency Seek time Transfer rate Delay between operations	
I/O direction	Read only Write only Read-write	ซีดีรอม ตัวควบคุมการแสดงผล ดิสก์

คุณลักษณะของอุปกรณ์อินพุต / เอาต์พุต

แอปพลิเคชันอินพุต/เอาต์พุตอินเทอร์เฟซ (Application I/O Interface)

- บล็อกดีไวซ์ และคาแรกเตอร์ดีไวซ์ (Block and Character Devices)
  - การติดต่อกับอุปกรณ์ที่จัดการในลักษณะของบล็อกข้อมูล จะใช้คำสั่งซึ่งตัวอุปกรณ์เองจะคุ้นเคยกับคำสั่งต่าง ๆ อยู่แล้ว เช่น การอ่าน, การเขียน, การค้นหา เป็นคำสั่งพื้นฐานสำหรับอุปกรณ์ประเภทดิสก์ และซีดีรอม
  - การแอ็กเซสไฟล์แบบการแมพหน่วยความจำ(Memory mapped) สามารถแอ็กเซสได้ในเลเยอร์สูงสุดของไดร์เวอร์ที่ใช้ในอุปกรณ์นั้นๆ
  - การแมพหน่วยความจำจะใช้เนื้อที่บนดิสก์เสมือนเนื้อที่ของหน่วยความจำ เมื่อต้องการบล็อกข้อมูลใดก็ทำการโหลดข้อมูลนั้นมาทันที
  - การแมพหน่วยความจำเป็นการให้บริการอีกอย่างหนึ่งของKernel
  - วิธีการเรียกการแมพหน่วยความจำยังสามารถใช้สำหรับจัดสรรเนื้อที่ดิสก์ (swapping) เพื่อให้สะดวกในการค้นหา

แอปพลิเคชันอินพุต/เอาต์พุตอินเทอร์เฟซ (Application I/O Interface)

- เน็ตเวิร์คดีไวซ์ (Network Devices)
  - การติดต่อกับอุปกรณ์เน็ตเวิร์คจะใช้แอปพลิเคชันชนิดหนึ่งที่มีชื่อว่า “ซ็อกเก็ต” (Socket)
  - ใช้โปรแกรมฟังก์ชัน Read-Write-Seek สำหรับติดต่อกับดิสก์
  - การใช้ซ็อกเก็ตติดต่อที่เห็นกันโดยทั่วไปในระบบปฏิบัติการหลายตัว เช่น Windows NT, UNIX
  - ภายในซ็อกเก็ตมีฟังก์ชันที่ใช้ติดต่อกับ Remote Address ชื่อว่า Select ฟังก์ชัน select เป็นตัวจัดการในการติดต่อประสานงาน รับ-ส่งข้อมูล
  - เมื่อระบบเรียกฟังก์ชัน select ฟังก์ชันจะส่งค่าเพื่อแสดงสถานะภาพขณะนั้น กลับไปยังผู้เรียก จากนั้นจะทำการรับ-ส่งข้อมูลกันตามที่เรียกเข้ามา

แอปพลิเคชันอินพุต/เอาต์พุตอินเทอร์เฟซ (Application I/O Interface)

- Clock and Timers
  - ทำหน้าที่หลัก3 ประการ คือ
    - บอกเวลาปัจจุบัน
    - บอกเวลาที่ใช้งานไปแล้ว
    - การตั้งเวลา
  - 3 ฟังก์ชันดังกล่าวมีการทำงานที่ต่างกัน
  - ฟังก์ชัน Programmable interval timer เป็นฟังก์ชันที่สามารถบอกเวลาที่ข้อมูลถูกอ่านแล้ว และยังสามารถบอกเวลาที่เหลือ เพื่อให้ระบบปฏิบัติการสร้างสัญญาณอินเทอร์รัพต์เมื่อโหลดข้อมูลเสร็จ
  - กลไกในเรื่องการกำหนดเวลาเป็นสิ่งสำคัญและจำเป็นในระบบคอมพิวเตอร์ที่มีอุปกรณ์หลายชนิดทำงานร่วมกัน

แอปพลิเคชันอินพุต/เอาต์พุตอินเทอร์เฟซ (Application I/O Interface)

- ในคอมพิวเตอร์หลายๆ เครื่องสัญญาณอินเทอร์รัพต์ที่เกิดขึ้น จะถูกแสดงออกมาในช่วงความละเอียด 18-60 ครั้งวินาที ด้วยตัว clock ของนาฬิกาเอง
- ในขณะที่ซีพียูสามารถเอ็กซิคิวต์คำสั่งต่างๆได้ถึงล้านๆ คำสั่งใน วินาที
- แสดงให้เห็นว่าความละเอียดของ clock มีความละเอียดน้อยมากเมื่อเทียบกับซีพียู
- ปัจจุบันคอมพิวเตอร์บางตัวสามารถใช้counter clock ซึ่งมีความละเอียดสูงเข้ามาแทน ทำให้สร้างสัญญาณอินเทอร์รัพต์แสดงออกได้ละเอียดมากขึ้น

แอปพลิเคชันอินพุต/เอาต์พุตอินเทอร์เฟซ (Application I/O Interface)

- Blocking I/O and Nonblocking I/O
  - มี 2 แนวทางในการเรียก System call คือ
    - Blocking I/O
    - Nonblocking I/O (Asynchronous)
  - เมื่อโปรแกรมใช้ Blocking I/O ระบบที่กำลังทำงานขณะนั้นจะหยุดรอ โปรแกรมจะสั่งให้ระบบย้ายการควบคุมไปที่ผู้เรียก System call ไปทำงานที่ถูกอินเทอร์รัพต์เข้ามาก่อน
  - เมื่อทำงานเสร็จโปรแกรมจะย้ายการควบคุมกลับไปยังที่เดิมก่อนที่จะเรียก System call
  - ลักษณะการเกิด Blocking ทางกายภาพแล้วช่วงเวลาตั้งแต่การเรียกSystem call จนกระทั้งกลับเข้าสู่โปรเซสปกติ

แอปพลิเคชันอินพุต/เอาต์พุตอินเทอร์เฟซ (Application I/O Interface)

- System call แบบ Nonblocking I/O (Asynchronous) มีประโยชน์ยิ่งในด้านความเร็วในระหว่างที่เรียก System call
- ในระหว่างที่เรียก System call งานที่กำลังทำขณะนั้นก็ยังคงดำเนินต่อไปจนกระทั่งเสร็จสิ้น แล้วจึงเรียกอินเทอร์รัพต์ เพื่อนำงานทั้งหมดที่เรียก System call มาเชื่อมเข้าด้วยกัน
- ตัวอย่างของNonblocking I/O คือฟังก์ชัน Selectในระบบเน็ตเวิร์ค เมื่อมีการเรียก System call ด้วยฟังก์ชัน Select ระบบไม่สามารถทราบได้ว่าฟังก์ชัน Select จะรีเทิร์นกลับเมื่อไร
- ถ้าเลยเวลาที่กำหนดไปแล้ว แสดงว่าSystem call ทำงานไม่สำเร็จ

ระบบย่อยอินพุต/เอาต์พุตใน Kernel (I/O Subsystem)

- การจัดเวลาอินพุต/เอาต์พุต (I/O Scheduling)
  - I/O Scheduling เป็นส่วนหนึ่งของ Kernel ช่วยจัดลำดับขั้นตอนต่างๆ ให้กับซีพียู เพื่อให้ซีพียูทำงานเต็มประสิทธิภาพ
  - ตัวอย่างการอ่านบล็อกข้อมูลจากดิสก์ ถ้ามีโปรเซสที่ต้องการทำอยู่ โปรเซสด้วยกัน
    - 1. บล็อกข้อมูลอยู่ตอนท้ายของโปรเซส
    - 2. บล็อกข้อมูลอยู่ตอนต้นของโปรเซส
    - 3. บล็อกข้อมูลอยู่ตอนกลางของโปรเซส

2	3	1

ระบบย่อยอินพุต/เอาต์พุตใน Kernel (I/O Subsystem)

- การใช้บัฟเฟอร์
  - บัฟเฟอร์เป็นหน่วยความจำประเภทหนึ่งที่ใช้เก็บข้อมูลขณะที่มีการถ่ายโอนข้อมูลกันระหว่างอุปกรณ์ทั้งสองชนิด ที่มาของการสร้างบัฟเฟอร์มีเหตุผล3 ประการคือ
    - สำหรับอุปกรณ์ที่มีความเร็วที่แตกต่างกันมาก ๆ จะช่วยให้อุปกรณ์มีความเร็วมากไม่ต้องเสียเวลาคอย
    - การใช้บัฟเฟอร์เพื่อปรับปรุงอุปกรณ์ที่มีความสามารถในการถ่ายโอนข้อมูลที่มีขนาดต่างกันให้สามารถทำงานควบคู่กันได้อย่างมีประสิทธิภาพ
    - ใช้สำหรับทำการแบ็คอัพข้อมูลซึ่งมักจะใช้เป็นประจำในคอมพิวเตอร์เป็นการป้องกันการสูญหายของข้อมูล ทำให้อุปกรณ์ต่าง ๆ เป็นอิสระในขณะที่มีการจัดสรรเนื้อที่บนดิสก์ให้เป็นระเบียบ

ระบบย่อยอินพุต/เอาต์พุตใน Kernel (I/O Subsystem)

- การใช้แคช
  - แคช คือ หน่วยความจำประเภทหนึ่งที่มีความเร็วสูงมาก การเข้าถึงข้อมูลในแคชสามารถทำได้รวดเร็วกว่าในดิสก์
  - จุดประสงค์ในการใช้แคช สำหรับอุปกรณ์มีจุดประสงค์เดียวกันกับบัฟเฟอร์ คือไว้สำหรับทำแบ็คอัพ
  - แคชมีราคาแพงกว่าบัฟเฟอร์ ดังนั้นการทำแบ็คอัพเนื้อที่มากๆ จึงจำเป็นต้องใช้บัฟเฟอร์แทน

ระบบย่อยอินพุต/เอาต์พุตใน Kernel (I/O Subsystem)

- สพูลลิ่งและการสงวนดีไวซ์ (Spooling and Device Reservation)
  - สพูล (Spool) คือบัฟเฟอร์ชนิดหนึ่งที่ใช้สำหรับเก็บข้อมูลมาก ๆ ไว้รวมกัน ตัวอย่างได้แก่ เครื่องพิมพ์
  - ระบบปฏิบัติการจะใช้สพูลลิ่งเข้ามาจัดการข้อมูล แล้วนำไปเก็บในสพูลตามลำดับที่ส่งเข้ามา
  - สพูลจะเป็นตัวที่ทำหน้าที่จัดคิวให้กับเครื่องพิมพ์ไม่ให้ปะปนกัน
  - งานต่าง ๆ ที่เก็บอยู่ในสพูล ผู้ใช้สามารถควบคุมผ่านระบบปฏิบัติการ ให้สามารถยกเลิกการพิมพ์งานใดๆ ได้ หรือแม้กระทั่งจัดลำดับการพิมพ์ใหม่ได้เช่นกัน

ระบบย่อยอินพุต/เอาต์พุตใน Kernel (I/O Subsystem)

- Error Handling
  - ตัวระบบปฏิบัติการทำหน้าที่ประสานการติดต่อระหว่างอุปกรณ์กับแอปพลิเคชันต่างๆ ของผู้ใช้ ทำให้ระบบปฏิบัติการมีความสามารถในการป้องกันความเสียหายที่อาจจะเกิดขึ้นได้สำหรับอุปกรณ์โดยเฉพาะดิสก์
  - สาเหตุที่ทำให้เกิดความล้มเหลวในขณะอ่านข้อมูลจากดิสก์ เช่น อุปกรณ์ชำรุด ระบบคอมพิวเตอร์เกิดโอเวอร์โหลดขึ้น
  - หากเกิดปัญหาจากอุปกรณ์ หรือแอปพลิเคชัน ระบบปฏิบัติการสามารถฟ้องข้อผิดพลาดได้
  - แต่ถ้าเกิดขึ้นในตัวระบบปฏิบัติการเอง ไม่สามารถแก้ไขปัญหานี้ได้
  - เมื่ออุปกรณ์เรียก System call ระบบจะส่งค่าขนาด 1 บิต กลับมาเพื่อเป็นการตอบรับ

ระบบย่อยอินพุต/เอาต์พุตใน Kernel (I/O Subsystem)

- การแปลคำสั่งจากการร้องขอให้ฮาร์ดแวร์ทำงาน(Transforming I/O Requests to Hardware Operation)
  - หัวข้อผ่านมาเป็นวิธีการติดต่อโปรแกรมควบคุมอุปกรณ์กับแอปพลิเคชันโดยมีระบบปฏิบัติการเป็นตัวกลาง แอปพลิเคชันสามารถอ้างชื่อไฟล์ในการติดต่อกับไฟล์ข้อมูลที่อยู่บนดิสก์ได้
  - ชื่อไฟล์เหล่านี้ถูกอ้างถึงโดยผู้ใช้แล้วส่งผลให้ตัวควบคุมดิสก์เพื่อสามารถเข้าไปจัดการข้อมูลที่เก็บในดิสก์ได้
  - ระบบปฏิบัติการ MS-DOS การแอ็กเซสดิสก์สามารถแอ็กเซสโดยใช้ Colon (:)

ระบบย่อยอินพุต/เอาต์พุตใน Kernel (I/O Subsystem)

- ประสิทธิภาพ (Performance)
  - อุปกรณ์ต่างๆ ที่ต่อเข้ากับคอมพิวเตอร์เป็นองค์ประกอบสำคัญที่มีผลต่อประสิทธิภาพของระบบคอมพิวเตอร์ ไม่ว่าจะเป็นการจัดวางของอุปกรณ์ในตำแหน่งที่เหมาะสม การจัดคิวของงานให้กับซีพียู มีการบล็อกข้อมูลหรือไม่ ล้วนมีผลต่อประสิทธิภาพของระบบทั้งสิ้น
  - การออกแบบสถาปัตยกรรมคอมพิวเตอร์จึงถือว่ามีความสำคัญต่อระบบโดยตรง
  - แม้ปัจจุบันคอมพิวเตอร์จะสามารถรองรับการอินเทอร์รัพต์ได้เป็นจำนวนนับร้อยครั้งต่อวินาที แต่การเกิดอินเทอร์รัพต์ก็ยังต้องใช้เวลาในการดูแลระบบให้ทำงานได้อย่างต่อเนื่อง

ระบบย่อยอินพุต/เอาต์พุตใน Kernel (I/O Subsystem)

- เน็ตเวิร์คเป็นหัวใจสำคัญในการเชื่อมต่อคอมพิวเตอร์เข้าไว้ด้วยกัน ยิ่งระบบมีคอมพิวเตอร์ต่อพ่วงเข้าด้วยกันมากเท่าไร ยิ่งทำให้เกิดความคับคั่งกับคอมพิวเตอร์มากเท่านั้น
- อุปกรณ์ที่ใช้ในการเชื่อมเน็ตเวิร์คที่ใช้คือคอนเท็กซ์สวิตซ์ (Context Switch)
- หน้าที่ของ Context Switch จะเชื่อมการทำงานระหว่าง Kernel กับเน็ตเวิร์ค

ระบบย่อยอินพุต/เอาต์พุตใน Kernel (I/O Subsystem)

- เทคนิควิธีการต่าง ๆ ที่ใช้เพิ่มประสิทธิภาพของระบบคอมพิวเตอร์ ดังนี้คือ
  - ลดการเชื่อมต่อคอนเท็กซ์สวิตซ์ ตามจุดต่างๆ ให้ได้มากที่สุด
  - ลดจำนวนครั้งของการก็อปปี้ข้อมูลไปเก็บในหน่วยความจำในขณะการส่งข้อมูลระหว่างอุปกรณ์กับตัวโปรแกรม
  - ลดจำนวนอินเทอร์รัพต์ที่เกิดขึ้นโดยใช้บัฟเฟอร์เข้ามาเก็บข้อมูลให้มีจำนวนมากก่อน แล้วเรียกอินเทอร์รัพต์ในครั้งเดียว
  - เพิ่มการทำงานในทางขนานโดยใช้วิธีการของDMA
  - ย้ายโปรเซสที่เป็นพื้นฐานที่ทำเป็นประจำ ไปเป็นหน้าที่ของฮาร์ดแวร์แทน
  - สร้างสมดุลให้กับระบบโดยรวมโดยเฉพาะความเร็วของอุปกรณ์แต่ละชนิดที่แตกต่างกัน ให้สามารถทำงานร่วมกันได้อย่างคล่องตัว