



## บทที่ 7

### การจัดการอุปกรณ์รับ-แสดงผล

#### 7.1 ความนำ

ในการทำงานของระบบคอมพิวเตอร์นั้น จะต้องมีการรับและแสดงผลข้อมูลของระบบ การจัดการอุปกรณ์ที่ใช้ในการรับและแสดงผลข้อมูลจึงมีความสำคัญและจำเป็น โดยหน้าที่ทั้งหมดเป็นหน้าที่ของระบบปฏิบัติการเป็นผู้จัดการอุปกรณ์เหล่านั้น ให้สามารถรับและแสดงผลข้อมูลได้อย่างมีประสิทธิภาพ

#### 7.2 หน้าที่ของระบบปฏิบัติการในการจัดการอุปกรณ์

หน้าที่พื้นฐานของระบบปฏิบัติการในการดูแลอุปกรณ์ต่างๆ ในระบบคอมพิวเตอร์ โดยทั่วไปมีดังนี้

##### 7.2.1 ติดตามสถานะของอุปกรณ์

ระบบปฏิบัติการมีหน้าที่ในการติดตามสถานะของอุปกรณ์ทุกชิ้น วิธีที่นิยมใช้คือ UCB (Unit Control Block) เพื่อเก็บข้อมูลที่สำคัญต่างๆ ของสำหรับอุปกรณ์ แต่ละชิ้น UCB คล้าย PCB ของโปรเซส

##### 7.2.2. กำหนดการใช้อุปกรณ์

เป็นหน้าที่หนึ่งที่ระบบปฏิบัติการต้องกำหนดว่าอุปกรณ์ชิ้นใด โปรเซสใดเป็นผู้ใช้ จะใช้นานเท่าไร และเริ่มใช้เมื่อไรมาตรการในการกำหนดสิ่งเหล่านี้เพื่อให้การใช้งานอุปกรณ์แต่ละชิ้นเกิดประโยชน์สูงสุด มีเทคนิคการจัดการอุปกรณ์อยู่ 3 ประเภท คือ

1. การยกให้ (Dedicated Device) เป็นการกำหนดให้อุปกรณ์ถูกใช้ได้โดยโปรเซสเพียงโปรเซสเดียว โปรเซสอื่นๆ จะเข้ามาใช้อุปกรณ์ตัวนี้ไม่ได้

2. การแบ่งปัน (Shared Device) เป็นการกำหนดให้อุปกรณ์ถูกใช้ได้โดยหลายโปรเซสร่วมกัน ไม่เป็นของโปรเซสใดโปรเซสหนึ่ง

3. การปลอม (Virtual Device) เป็นการจำลองอุปกรณ์ชิ้นหนึ่งให้เป็นอุปกรณ์อีกชิ้นหนึ่ง เช่น จำลองดิสก์เป็นเครื่องพิมพ์ ซึ่งมีในระบบปฏิบัติการบางตัวเท่านั้น



### 7.2.3 จัดสรร (Allocate) อุปกรณ์

ในการมอบหมายอุปกรณ์ให้แก่โปรเซสแต่ละโปรเซสนั้นระบบปฏิบัติการจะต้องจัดสรรอุปกรณ์ เพื่อมอบหมายอุปกรณ์ชิ้นนั้นให้โปรเซส

### 7.2.4 เรียกลิ้น (Deal locate) อุปกรณ์

เมื่อโปรเซสที่ครอบครองอุปกรณ์จบลง ระบบปฏิบัติการจะนำเอาอุปกรณ์ต่างๆ ที่ถูกครอบครองโดยโปรเซสนี้คืนให้กับระบบ หมายความว่าอุปกรณ์ไม่ได้ถูกครอบครองโดยโปรเซสที่เพิ่งจบสิ้นนี้อีกต่อไป

ที่กล่าวมานี้เป็นเพียงหน้าที่ทั่วไปของระบบปฏิบัติการกับอุปกรณ์ต่างๆ ระบบปฏิบัติการที่ถูกออกแบบมาเพื่อความคล่องตัวในการใช้งานสูงจะมีการใช้ตัวขับอุปกรณ์ (Device Driver) และมีความสามารถในการจัดการเรื่องการทำอุปกรณ์เทียม หรืออุปกรณ์เสมือน (Virtual Device) รวมอยู่ในระบบปฏิบัติการเองด้วย

## 7.3 ประเภทของอุปกรณ์ของระบบ

โดยทั่วไปเราสามารถแบ่งแยกประเภทของอุปกรณ์ของระบบออกได้เป็น 3 ประเภทใหญ่ๆ โดยในแต่ละประเภทก็จะมีลักษณะของฟังก์ชันการทำงานที่แตกต่างกันออกไป ดังนี้

### 7.3.1 Dedicated devices

อุปกรณ์ประเภทนี้จะถูกจัดสรรให้กับโปรเซสเพียง 1 โปรเซสเท่านั้น จะไม่มีการเปิดให้โปรเซสต่างๆ ใช้อุปกรณ์ร่วมกัน กล่าวคือ ในเวลาหนึ่งๆ จะมีเพียง 1 โปรเซสเท่านั้นที่สามารถใช้งานอุปกรณ์ประเภทนี้ได้

ตัวอย่างอุปกรณ์ของระบบที่ถูกจัดอยู่ในประเภทนี้ได้แก่ เทปไดรฟ์ เครื่องพิมพ์ พลอตเตอร์ เป็นต้น จะพบว่าอุปกรณ์เหล่านี้ไม่สะดวกต่อการใช้งานร่วมกันระหว่างโปรเซสหลายๆ โปรเซส ยกตัวอย่างเช่น พลอตเตอร์ซึ่งเป็นอุปกรณ์วาดลายเส้นโดยอาศัยแขนจับปากกาเพื่อลากลายเส้นในแกน X และแกน Y ของกระดาษ หากมีการเปิดให้ใช้งานพลอตเตอร์ร่วมกันได้ อาจก่อให้เกิดปัญหาคือ เมื่อพลอตเตอร์วาดรูปภาพไปได้ครึ่งหนึ่ง ในระหว่างนั้นมีอีกโปรเซสมาขอใช้งานพลอตเตอร์ร่วมด้วย พลอตเตอร์จึงไปวาดรูปภาพของโปรเซสใหม่ ในขณะที่ยังวาดภาพของ



โปรเซสเดิมยังไม่เสร็จสมบูรณ์ จึงมีผลทำให้ภาพที่วาดออกมา เป็นรูปภาพจากโปรเซส 2 โปรเซส ครั้งแรกเป็นภาพของโปรเซสเดิม และครั้งหลังเป็นภาพของโปรเซสใหม่ เป็นต้น

ข้อเสียของอุปกรณ์ประเภทนี้คือ ในเวลาหนึ่งๆ หากมีโปรเซสใดๆ ใช้งานอุปกรณ์ไม่เต็ม 100 เปอร์เซ็นต์ ก็จะส่งผลให้เกิดการใช้งานอุปกรณ์อย่างไม่มีประสิทธิภาพ เนื่องจาก ณ เวลาหนึ่งๆ จะมีโปรเซสนั้นเพียงโปรเซสเดียวเท่านั้นที่จะสามารถใช้งานอุปกรณ์นั้นๆ ได้

### 7.3.2 Shared devices

เป็นอุปกรณ์ที่สามารถจัดสรรให้กับโปรเซสหลายๆ โปรเซสใช้งานร่วมกันได้ ตัวอย่างอุปกรณ์ของระบบที่ถูกจัดอยู่ในประเภทนี้ได้แก่ ดิสก์ต่างๆ เป็นต้น ในเวลาหนึ่งๆ โปรเซสต่างๆ สามารถส่งการร้องขอใช้งานดิสก์เข้ามาพร้อมกันได้

Device Manager ต้องควบคุมการใช้งานอุปกรณ์ร่วมกันระหว่างโปรเซสต่างๆ ไม่เช่นนั้น อาจก่อให้เกิดปัญหาการใช้งานอุปกรณ์ชนกันระหว่างโปรเซสได้ ตัวอย่างเช่น หากโปรเซสที่ 1 และโปรเซสที่ 2 ต้องการอ่านข้อมูลจากดิสก์เดียวกัน Device Manager ต้องจัดหานโยบายในการตัดสินใจว่าจะให้โปรเซสใดได้อ่านข้อมูลจากดิสก์นั้นก่อน

### 7.3.2 Virtual devices

สำหรับอุปกรณ์ประเภทนี้เป็นการนำ Dedicated devices มาแปลงให้อยู่ในรูปของ Shared devices ตัวอย่างเช่น เครื่องพิมพ์เป็น Dedicated devices และดิสก์เป็น Shared devices สามารถนำเครื่องพิมพ์มาแปลงเป็น Shared devices ได้ด้วยการนำเทคนิค Spooling มาใช้ ดังนั้นแทนที่ในเวลาหนึ่งๆ จะมีผู้ใช้งานเครื่องพิมพ์ได้แค่คนเดียว ก็สามารถให้ผู้ใช้งานหลายๆ คนสั่งพิมพ์งานมายังเครื่องพิมพ์เครื่องนั้นพร้อมๆ กันได้ โดยคำสั่งร้องขอพิมพ์งานจากผู้ใช้งานต่างๆ จะไม่ถูกส่งไปยังเครื่องพิมพ์โดยตรง แต่จะถูกพิมพ์ลงสู่ดิสก์ก่อน จากนั้นเมื่อเครื่องพิมพ์พร้อมทำงานก็จะส่งงานไปยังเครื่องพิมพ์เพื่อพิมพ์งานให้ผู้ใช้งานแต่ละคน ดังนั้น เครื่องพิมพ์เครื่องนี้จึงถูกเรียกว่า Virtual printer

## 7.4 ลักษณะการรับ-ส่งข้อมูลของอุปกรณ์รับ-แสดงผล

โดยทั่วไปแล้วอุปกรณ์รับ-แสดงผล (Input/Output Device) สามารถจำแนกตามลักษณะการจัดเก็บข้อมูลได้ 2 ประเภทดังนี้ (น.ท.ไพศาล โมลิสกุลมงคล และคนอื่นๆ, 2545)



### 7.4.1 Block Devices

คือ อุปกรณ์ที่จัดเก็บข้อมูลเป็นลักษณะบล็อกข้อมูล โดยมีขนาดของบล็อกข้อมูลประมาณ 512 ไบต์ ถึง 32,768 ไบต์ ตัวอย่างเช่น ดิสก์ (Disk) ต่างๆ ไม่ว่าจะเป็น ฮาร์ดดิสก์ ฟลอปปีดิสก์ ซีดี เป็นต้น ซึ่ง Block Devices มีคุณสมบัติดังนี้

1. สามารถที่จะอ่าน/เขียนข้อมูลลงในแต่ละบล็อกได้อย่างอิสระ ในอุปกรณ์ประเภทที่เป็นบล็อกจำพวกดิสก์
2. สามารถกำหนดตำแหน่งลงไปบนแผ่นดิสก์ได้ จะประกอบไปด้วยไซลินเดอร์ (Cylinder) และเซกเตอร์ (Sector) โดยมีหัวอ่านบนแขนกลทำหน้าที่อ่านข้อมูลในขณะที่หัวดิสก์หมุนรอบตัวไปด้วยความเร็วคงที่ หัวบริเวณพื้นที่ดิสก์หัวอ่านจะสามารถเลื่อนไปได้ทุกที่
3. แต่ละบล็อกที่บรรจุข้อมูลอยู่ภายในหัวอ่านสามารถนำข้อมูลออกมาได้ อุปกรณ์ที่มีการแอกเซสโดยตรง (Direct access storage device)



ภาพที่ 7.1 แสดงตัวอย่าง Block Devices

ที่มา (<http://www.kcw.ac.th/kcw5151/kraitong%201/p2.1.htm>, 2552)

### 7.4.2. Character Devices

คือ อุปกรณ์ที่จัดเก็บข้อมูลเป็นลักษณะสายอักขระ จำนวนข้อมูลขึ้นอยู่กับบัฟเฟอร์ (buffer) ที่มีแต่ไม่สามารถกำหนดตำแหน่งของบัฟเฟอร์เหล่านี้ได้ ตัวอย่างอุปกรณ์ประเภทนี้ ได้แก่ เครื่องพิมพ์ เครื่องอ่านภาพ การ์ดเครือข่าย เป็นต้น ซึ่ง Character Devices มีคุณสมบัติดังนี้

1. อุปกรณ์ประเภทนี้ข้อมูลที่ส่งเข้ามาออกจะเรียงมาเป็นลำดับก่อน-หลัง การแบ่งแยกข้อมูลทำได้โดยตรวจสอบลำดับของข้อมูล
2. อุปกรณ์ประเภทนี้สามารถจัดการได้ง่าย เพียงแต่จัดลำดับการรับ-ส่งข้อมูลให้ถูกต้อง



ภาพที่ 7.2 แสดงตัวอย่าง Character Devices

ที่มา (<http://pirun.ku.ac.th/~b4904281/page2.html>, 2552)

## 7.5 ตัวควบคุมอุปกรณ์

อุปกรณ์แต่ละอุปกรณ์จะมีองค์ประกอบ 2 ส่วน คือกลไกของอุปกรณ์นั้น และวงจรอิเล็กทรอนิกส์ที่ควบคุมกลไกการทำงานของอุปกรณ์ ซึ่งทั้งสององค์ประกอบนี้มักจะแยกออกจากกันโดยส่วนที่เป็นวงจรอิเล็กทรอนิกส์ที่ควบคุมกลไกการทำงานเรียกว่าตัวควบคุมอุปกรณ์ (Device Controller) หรือ อะแดปเตอร์ (Adapter) (นริรัตน์ นิยมไทย, 2549)

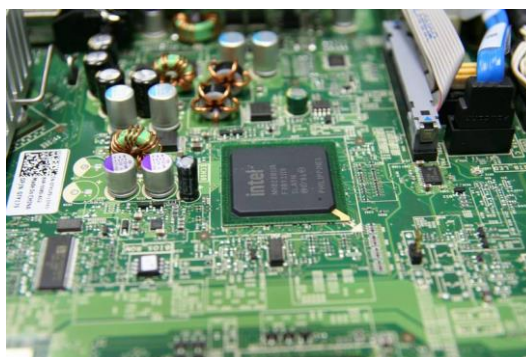
ตัวควบคุมอุปกรณ์จะอยู่ในรูปของแผงวงจร ซึ่งจะติดตั้งในเครื่องคอมพิวเตอร์แล้ว เชื่อมโยงเคเบิลไปยังตัวอุปกรณ์ ซึ่งตัวควบคุมอุปกรณ์ที่ใช้ในการควบคุมอุปกรณ์หนึ่ง จะใช้ควบคุมอุปกรณ์ได้เพียงอย่างเดียว ไม่สามารถนำไปใช้ควบคุมอุปกรณ์ชนิดอื่นได้ เช่น ตัวควบคุมดิสก์ จะได้สามารถนำไปใช้ควบคุมการทำงานของจอภาพ เป็นต้น

โดยปกติแล้วตัวควบคุมอุปกรณ์แต่ละตัวจะมีการติดต่อกับซีพียูโดยผ่านรีจิสเตอร์ โดยการใช้งานอุปกรณ์ต่างๆ นั้นระบบปฏิบัติการจะทำโดยการเขียนคำสั่งลงไปนรีจิสเตอร์ของตัวควบคุมอุปกรณ์นั้น โดยอาจจะมีการกำหนดค่าพารามิเตอร์ต่างๆ ไปพร้อมกับคำสั่งเพื่อควบคุมชนิดและรูปแบบของการใช้งานอุปกรณ์นั้นด้วย ซึ่งเมื่อทำงานเขียนคำสั่งเสร็จแล้วซีพียูก็จะปล่อยให้อุปกรณ์นั้นทำงานด้วยตัวเองโดยอาศัยข้อมูลต่างๆ ในการทำงานจากคำสั่งและพารามิเตอร์ที่เขียนลงไปนรีจิสเตอร์ หรือ ตำแหน่งความจำอุปกรณ์นั้น และซีพียูก็จะไปทำงานอื่น ๆ ต่อไป โดยไม่ต้องคอยการทำงานของอุปกรณ์ และเมื่อการทำงานของอุปกรณ์แล้วเสร็จ ตัวควบคุมอุปกรณ์จะส่งสัญญาณกลับไปยังให้ทางซีพียูทราบโดยผ่านระบบปฏิบัติการ เพื่อให้ซีพียูทำตรวจสอบผลลัพธ์ของการทำงาน โดยซีพียูจะอ่านข้อมูลจากรีจิสเตอร์ของตัวควบคุมอุปกรณ์นั้น เพื่อตรวจสอบผลลัพธ์ของการทำงานและสถานะของอุปกรณ์นั้นๆ เพื่อให้สามารถใช้งานในคราวต่อไปได้



แต่อย่างไรก็ตามในระบบคอมพิวเตอร์บางระบบตัวควบคุมอุปกรณ์อาจจะมีการติดต่อกับซีพียูด้วยวิธีการอื่น โดยไม่ต้องผ่านรีจิสเตอร์ก็ได้ ซึ่งจะสามารถทำได้โดยการเตรียมเนื้อที่หน่วยความจำเฉพาะไว้เพื่อทำหน้าที่เป็นช่องทางในการติดต่อกับซีพียูโดยเฉพาะ โดยไม่ต้องอาศัยการทำงานของรีจิสเตอร์เข้าช่วย โดยมีการกำหนดตำแหน่งของหน่วยความจำที่ใช้เพื่อการทำงานของตัวควบคุมอุปกรณ์ต่างๆ ซึ่งเรียกว่าการจับคู่หน่วยความจำกับอุปกรณ์ (Memory Map I/O) และในแต่ละตำแหน่งหน่วยความจำที่ใช้การทำงานนั้นจะมีหมายเลขประจำตัวหรือที่เรียกว่าหมายเลขการขัดจังหวะ (Interrupt Vector) กำหนดอยู่ เมื่อระบบต้องการใช้งานอุปกรณ์ใดก็จะทำงานส่งสัญญาณหมายเลขขัดจังหวะซึ่งเป็นหมายเลขประจำตัวของอุปกรณ์นั้นเพื่อแจ้งให้ซีพียูทราบว่าต้องการใช้งานอุปกรณ์ตัวนั้นๆ

อุปกรณ์ รับ-แสดงผลข้อมูล แต่ละประเภทนั้นจะประกอบไปด้วยกลไกต่างๆ และส่วนที่เป็นวงจรอิเล็กทรอนิกส์ ประกอบรวมกันเป็นตัวชิพ (Chip) หรือเป็นแผงวงจร (Circuit Board) ซึ่งเรียกว่า ตัวควบคุมอุปกรณ์ นั้นเอง



ภาพที่ 7.3 แสดงตัวอย่าง ตัวควบคุมอุปกรณ์

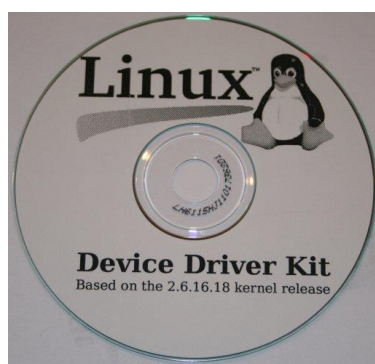
ที่มา ([http://www.mahamodo.com/server/R300\\_5\\_IntelChipControl.JPG](http://www.mahamodo.com/server/R300_5_IntelChipControl.JPG), 2552)

การติดต่อระหว่างตัวควบคุมอุปกรณ์ กับอุปกรณ์มักอยู่ในรูปของภาษาเครื่อง (Machine Language) โดยอาศัยระบบบัส (System Bus) ข้อมูลในการทำงาน ซึ่งมักจะมีข้อผิดพลาดเกิดขึ้น ต้องอาศัยการตรวจสอบข้อผิดพลาดของการทำงานดังกล่าวเรียกว่า Checksum Error



## 7.6 ตัวขับอุปกรณ์

อุปกรณ์แต่ละชนิดย่อมมีลักษณะที่แตกต่างกันออกไปทั้งในแง่การติดต่อส่งข้อมูล และการควบคุมการทำงานของตัวอุปกรณ์เอง ระบบปฏิบัติการจะต้องควบคุมและติดต่อกับอุปกรณ์ต่างๆ ทุกประเภท ถ้าเรานำโปรแกรมส่วนควบคุมอุปกรณ์ต่างๆ ทั้งหมดมารวมไว้ในตัวระบบปฏิบัติการ จะทำให้ระบบปฏิบัติการมีขนาดใหญ่มาก ระบบปฏิบัติการจำเป็นจะต้องรู้ลักษณะการทำงานของอุปกรณ์ทุกประเภท เพื่อสามารถควบคุมการทำงานของอุปกรณ์นั้นๆ ได้ถูกต้อง ซึ่งถ้าเราต้องการติดตั้งอุปกรณ์ชนิดใหม่เข้าไปในระบบคอมพิวเตอร์ของเรา เราไม่สามารถทำได้ ทั้งนี้เพราะในตัวระบบปฏิบัติการ ไม่มีโปรแกรมที่จะจัดการกับอุปกรณ์ชนิดใหม่นี้ มันจึงไม่สามารถติดต่อและควบคุมอุปกรณ์ใหม่ได้ ถ้าต้องการจะติดตั้งอุปกรณ์ตัวใหม่เข้าไปเราต้องแก้ไขปรับปรุงระบบปฏิบัติการใหม่ ซึ่งอาจเสียค่าใช้จ่ายสูงและการเปลี่ยนระบบปฏิบัติการทำได้ไม่สะดวกนัก คือเราจำเป็นต้องหยุดการทำงานของระบบทั้งหมดเพื่อเปลี่ยนเอาระบบปฏิบัติการตัวใหม่ที่ได้รับการปรับปรุงแล้วเข้าไปทำงานแทน



ภาพที่ 7.4 แสดงตัวอย่างซีดีสำหรับใช้ติดตั้งตัวขับอุปกรณ์ในระบบปฏิบัติการลินุกซ์

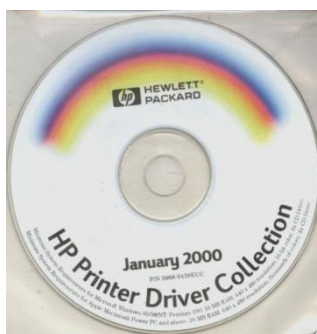
ที่มา (<http://www.kroah.com/log/2006/05/24/, 2552>)

เพื่อหลีกเลี่ยงปัญหาดังกล่าว ผู้ออกแบบระบบปฏิบัติการจึงมีความคิดที่จะแยกเอาส่วนควบคุมอุปกรณ์ทั้งหลายออกจากตัวระบบปฏิบัติการ โปรแกรมที่แยกออกมาต่างหากและมีหน้าที่ควบคุมติดต่อกับอุปกรณ์ต่างๆ นี้เรียกว่า ตัวขับอุปกรณ์ (Device Driver) อุปกรณ์แต่ละชนิดจะมีตัวขับอุปกรณ์ของมันเองและแตกต่างจากตัวขับอุปกรณ์ชนิดอื่นๆ ทั้งนี้เพราะการควบคุมติดต่อ



อุปกรณ์แต่ละชนิดไม่เหมือนกัน เช่น มีตัวขับอุปกรณ์ของเครื่องพิมพ์ ตัวขับอุปกรณ์ของเครื่องอ่านบัตร ตัวขับอุปกรณ์ของดิสก์ เป็นต้น

ถึงแม้ว่าเป็นอุปกรณ์ชนิดเดียวกันแต่ถ้าถูกผลิตจากผู้ผลิตต่างก็อาจมีการควบคุมติดต่อที่ต่างกันได้ เช่น เครื่องพิมพ์ยี่ห้อ EPSON มีการควบคุมต่างกับเครื่องพิมพ์ยี่ห้อ NEC ในกรณีนี้เครื่องพิมพ์ทั้งสองจะต้องมีตัวขับอุปกรณ์คนละตัวกัน เมื่อระบบปฏิบัติการต้องการจะติดต่อกับอุปกรณ์ชนิดใด ระบบปฏิบัติการก็จะติดต่อผ่านทางตัวขับอุปกรณ์ชนิดนั้น แล้วตัวขับอุปกรณ์จะติดต่อกับอุปกรณ์จริงๆ อีกทีหนึ่ง การติดต่อของระบบปฏิบัติการกับอุปกรณ์ไม่ว่าจะเป็นตัวขับอุปกรณ์ของอุปกรณ์ชนิดใดก็ตามจะเป็นรูปแบบเดียวกันหมด ระบบปฏิบัติการติดต่อกับอุปกรณ์ผ่านทางตัวขับอุปกรณ์ ในลักษณะเช่นนี้ ระบบปฏิบัติการไม่จำเป็นต้องรู้ลักษณะการทำงานของอุปกรณ์ต่างๆ เลย ปล่อยให้มันเป็นหน้าที่ของตัวขับอุปกรณ์ชนิดนั้นๆ ระบบปฏิบัติการรู้เพียงแต่การติดต่อและควบคุมอุปกรณ์ผ่านทางตัวขับอุปกรณ์เท่านั้น ซึ่งจะเหมือนกันหมด ดังนั้นการติดตั้งอุปกรณ์ชนิดใหม่เข้าไปในระบบก็ทำได้ง่ายมาก เพียงแต่นำโปรแกรมตัวขับอุปกรณ์ของอุปกรณ์ชิ้นนั้นเพิ่มเข้าไปในระบบ การแก้ไขระบบก็ทำเพียงแต่เขียนโปรแกรมตัวขับอุปกรณ์ขึ้นมาเท่านั้น ไม่จำเป็นต้องเข้าไปยุ่งเกี่ยวกับตัวระบบปฏิบัติการแต่อย่างใด เพราะระบบปฏิบัติการยังคงติดต่อกับอุปกรณ์ใหม่เหมือนกับอุปกรณ์อื่นเช่นกัน (นริรัตน์ นิยมไทย, 2549)



ภาพที่ 7.5 แสดง ตัวอย่างซีดีสำหรับใช้ติดตั้งตัวขับอุปกรณ์ของเครื่องพิมพ์

ที่มา (<http://cgi.ebay.com.my/ws/eBayISAPI.dll?ViewItem&item=280349097850>, 2552)





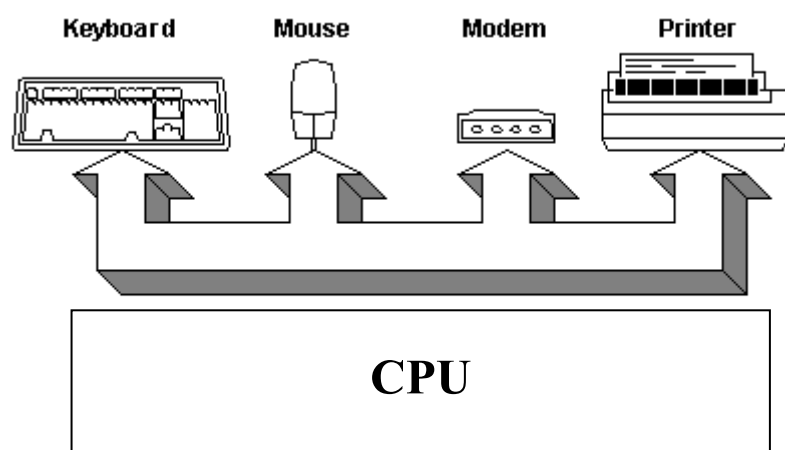
## 7.7 การรับส่งข้อมูลระหว่างซีพียูกับอุปกรณ์

เมื่อโปรเซสต้องการส่งข้อมูลให้กับอุปกรณ์ต่างๆ ข้อมูลที่ส่งจะถูกจัดเก็บไว้ในหน่วยความจำหลักที่ถูกครอบครองโดยโปรเซสนั้น การส่งข้อมูลจะกระทำโดยซีพียูเป็นผู้นำข้อมูลมาจากหน่วยความจำส่วนนี้และส่งไปให้อุปกรณ์ต่างๆ และในทางกลับกันเมื่ออุปกรณ์ต่างๆ ต้องการส่งข้อมูลไปให้กับโปรเซส ข้อมูลจะถูกส่งผ่านซีพียูไปไว้ในหน่วยความจำส่วนเดียวกันนี้ จากนั้นโปรเซสจึงจะสามารถนำข้อมูลไปใช้งานได้

ในการทำงานของคอมพิวเตอร์นั้นมักจะมีการรับส่งข้อมูลระหว่างซีพียูกับอุปกรณ์ต่างๆ ซึ่งระบบปฏิบัติการมีหน้าที่ในการจัดการรับส่งข้อมูลเหล่านั้น วิธีในการรับส่งข้อมูลระหว่างซีพียูกับอุปกรณ์ มีอยู่ด้วยกันหลายวิธี ดังนี้ (พิรพร หมุนสนธิ และคนอื่น ๆ, 2553)

### 7.7.1 Polling

วิธีการรับ-ส่งข้อมูล แบบ Polling นั้นซีพียูจะต้องคอยสอบถามอุปกรณ์ทั้งหมด ในทุกๆ ช่วงเวลาว่าต้องการให้ซีพียูทำงานให้หรือไม่ ทำให้อุปกรณ์ต่างๆ ที่ต้องการให้ซีพียูทำการประมวลผลให้ต้องรอคอยการสอบถามจากซีพียูและเป็นการเสียเวลาการทำงานของซีพียูมากหากสอบถามอุปกรณ์แล้วไม่มีอุปกรณ์ใดต้องการให้ซีพียูประมวลผลให้



ภาพที่ 7.6 แสดงวิธีการรับ-ส่งข้อมูล แบบ Polling

ที่มา ([http://www.windowsnetworking.com/articles\\_tutorials/interrupt.html](http://www.windowsnetworking.com/articles_tutorials/interrupt.html), 2552)



เมื่อซีพียูส่งสัญญาณออกไปสอบถามสถานะการทำงานของอุปกรณ์ชิ้นใดชิ้นหนึ่ง จะได้รับสัญญาณตอบกลับจากอุปกรณ์ มีคำตอบ ดังนี้

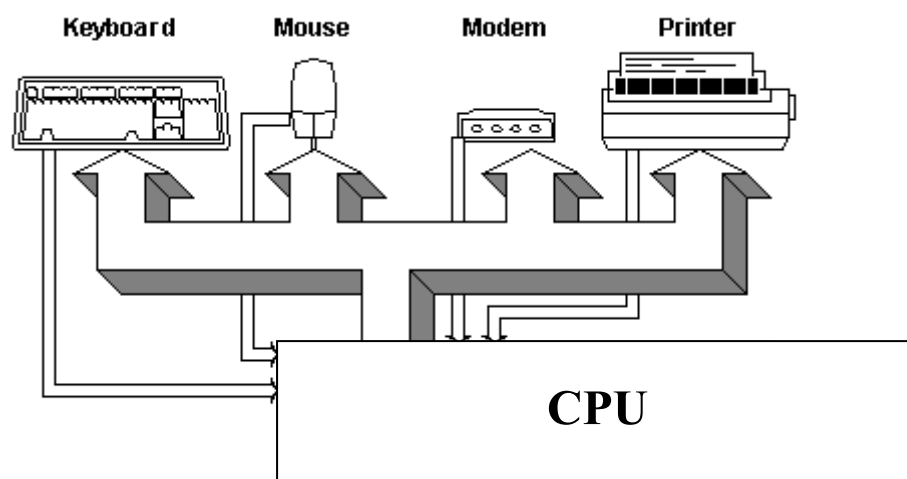
1. Command Ready หมายถึง สถานะที่อุปกรณ์พร้อมที่จะรับคำสั่งเพื่อปฏิบัติงาน
2. Busy หมายถึง สถานะที่อุปกรณ์ไม่พร้อมที่จะรับคำสั่งเพื่อปฏิบัติงาน เนื่องจากอุปกรณ์นั้นกำลังปฏิบัติงานตามคำสั่งอื่นอยู่
3. Error หมายถึง สถานะที่อุปกรณ์เกิดความผิดปกติในการทำงาน ไม่สามารถรับคำสั่งใดๆ ได้อีก เช่น กระดาษติดที่เครื่องพิมพ์ หมึกพิมพ์หมด ดิสก์เต็ม ดิสก์เสีย เป็นต้น

เมื่อซีพียูต้องรอการทำงานของอุปกรณ์ จะเกิดสภาวะการรอคอยเรียกว่า Busy Waite Cycle ซึ่งเป็นช่วงเวลาที่ซีพียูไม่สามารถทำงานอื่นได้ต้องรอนานกว่าช่วงเวลารอคอยนี้สิ้นสุดลง คือเมื่อสามารถเรียกใช้งานอุปกรณ์นั้นได้นั่นเอง

### 7.7.2 IRQ (Interrupt Request)

เป็นวิธีการหยุดการทำงานของซีพียู โดยอุปกรณ์รับ-แสดงผลเป็นผู้ส่งสัญญาณนี้ขึ้นมา และผู้ที่ทำหน้าที่หยุดการทำงานของซีพียูเรียกว่า Interrupt Handler โดยหลักการทำงานของ Interrupt Handler คือ มีโครงสร้างข้อมูลที่เรียกว่า Interrupt Vector ซึ่งเป็นที่เก็บตำแหน่งของโปรแกรมอินเทอร์รัพท์สำหรับสัญญาณอินเทอร์รัพท์ต่างๆ โดยที่สัญญาณอินเทอร์รัพท์ต่างๆ นี้จะมีหมายเลขสัญญาณที่ไม่ซ้ำกัน

วิธีการรับ-ส่งข้อมูล แบบ IRQ นั้นในขณะที่ซีพียูกำลังทำงานอยู่ อุปกรณ์ต่างๆ สามารถร้องขอมาที่ซีพียูให้ช่วยประมวลผลได้ทุกช่วงเวลา โดยการส่งสัญญาณอินเทอร์รัพท์มายังซีพียู ซึ่งอุปกรณ์แต่ละประเภทจะมีหมายเลข IRQ ที่ต่างกัน โดยหมายเลข IRQ ที่มีค่าความสำคัญ (Priority) สูงสุดคือ IRQ หมายเลข 0 (NMI = Non Maskable Interrupt) เป็นสัญญาณอินเทอร์รัพท์ที่ไม่สามารถรอกอยได้ซีพียูต้องทำการประมวลผลให้กับงานที่มี IRQ เป็น 0 ก่อนเสมอ เมื่อเกิดอินเทอร์รัพท์ ขึ้นสัญญาณการอินเทอร์รัพท์จะถูกส่งผ่านไปตามบัสของข้อมูลและหากมีสัญญาณการอินเทอร์รัพท์เกิดขึ้นมากกว่า 1 สัญญาณ อุปกรณ์ควบคุมอินเทอร์รัพท์จะตรวจสอบค่าความสำคัญของสัญญาณแล้วจัดลำดับงานก่อน-หลัง ส่งต่อไปยังซีพียู

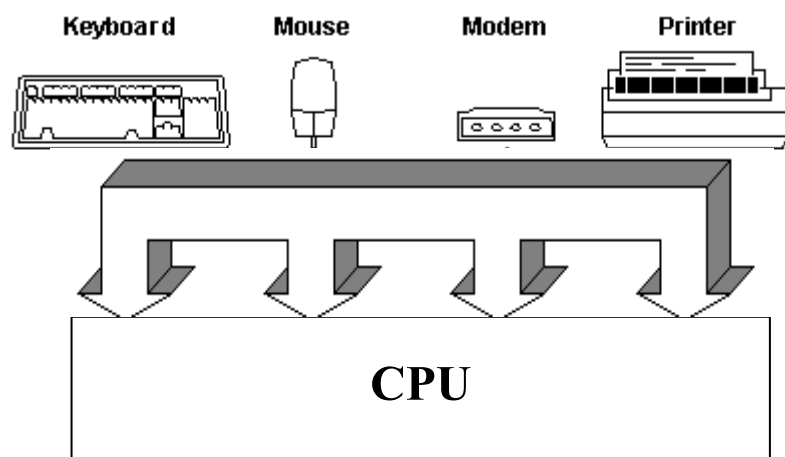


ภาพที่ 7.7 แสดงวิธีการรับ-ส่งข้อมูล แบบ IRQ

ที่มา ([http://www.windowsnetworking.com/articles\\_tutorials/interrupt.html](http://www.windowsnetworking.com/articles_tutorials/interrupt.html), 2552)

### 7.7.3 Mail Box

วิธีการ รับ-ส่ง ข้อมูลแบบ Mail Box นั้นซีพียูจะกันเนื้อที่หน่วยความจำส่วนหนึ่งเอาไว้รอรับข้อความ (Message) จากอุปกรณ์ที่ต้องการให้ซีพียูช่วยประมวลผล หากมีอุปกรณ์ใดต้องการให้ซีพียูช่วยประมวลผล อุปกรณ์นั้นต้องส่งข้อความ มายังซีพียู โดยหากมีข้อความของอุปกรณ์ต่างๆ มากกว่า 1 ข้อความ ซีพียูก็จะจัดลำดับงานก่อน-หลัง ตามลำดับข้อความที่ซีพียูได้รับมาจากอุปกรณ์



ภาพที่ 7.8 แสดงวิธีการรับ-ส่งข้อมูล แบบ Mail Box

ที่มา ([http://www.windowsnetworking.com/articles\\_tutorials/interrupt.html](http://www.windowsnetworking.com/articles_tutorials/interrupt.html), 2552)

#### 7.7.4 DMA (Direct Memory Access)

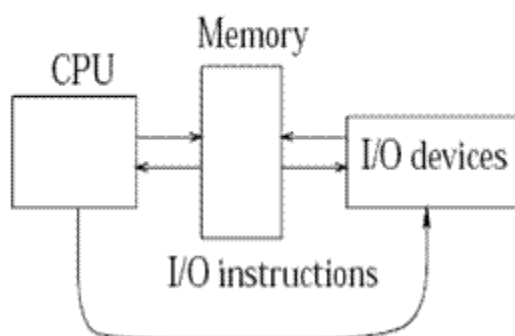
DMA เป็นการรับ – ส่ง ข้อมูลระหว่างอุปกรณ์ โดยไม่ต้องส่งผ่านซีพียู ซึ่ง DMA สามารถเกิดได้กับทุกอุปกรณ์และทุกช่วงเวลาที่มีซีพียูกำลังคำนวณผลทางตรรกะ ซึ่งเป็นช่วงที่ซีพียูกำลังคำนวณผลโดยใช้ ALU ในการทำงานซึ่งซีพียูจะไม่ใช้หน่วยความจำในการทำงาน

เมื่อโปรเซสต้องการส่งข้อมูลให้อุปกรณ์ต่างๆ ข้อมูลที่ถูกส่งจะถูกเก็บไว้ในหน่วยความจำที่ถูกครอบครองโดยโปรเซส การส่งข้อมูลจะกระทำโดยซีพียูเป็นผู้ดึงข้อมูลมาจากหน่วยความจำส่วนนี้และส่งไปให้อุปกรณ์ต่างๆ ในทางกลับกันเมื่ออุปกรณ์ต่างๆ ต้องการส่งข้อมูลให้โปรเซส ข้อมูลจะถูกส่งผ่านซีพียูไปไว้ในหน่วยความจำส่วนนี้ จากนั้นโปรเซสจึงนำข้อมูลไปใช้ได้ วิธีการรับส่งข้อมูลต้องผ่านซีพียู เป็นไปได้ช้าและเปลืองเวลาของซีพียู ดังนั้น ถ้าเราสามารถรับส่งข้อมูลได้โดยไม่ต้องผ่านซีพียูก็จะทำให้การรับข้อมูลเร็วขึ้น และยังสามารถใช้ซีพียูสำหรับประมวลผลโปรเซสอื่นได้ต่อไป

การรับส่งข้อมูลแบบ DMA จำเป็นต้องอาศัยแชนแนล (Chanel) หรือตัวควบคุม DMA (DMA controller) แชนแนลจะทำหน้าที่แทนซีพียู เมื่อต้องการรับส่งข้อมูลแบบ DMA แชนแนลจะส่งสัญญาณไปบอกให้ซีพียูรับรู้ ซีพียูจะสั่งให้แชนแนลทำธุรกรรมควบคุมการส่งข้อมูล จากนั้นซีพียูจะไปทำงานอื่น และเมื่อการทำ DMA เสร็จสิ้น แชนแนลจะส่งสัญญาณบอกให้ซีพียูรับรู้อีกครั้งว่าการทำ DMA เสร็จสิ้นลงแล้ว



## Direct Memory Access (DMA) Structure



ภาพที่ 7.9 แสดงวิธีการรับ-ส่งข้อมูล แบบ DMA

ที่มา ([http://www.windownetworking.com/articles\\_tutorials/interrupt.html](http://www.windownetworking.com/articles_tutorials/interrupt.html), 2552)

ในขณะที่ซีพียูกำลังทำงานอื่นอยู่ และมีการทำ DMA เกิดขึ้นพร้อมกันด้วย แต่การทำงานของ DMA ไม่ได้เกิดขึ้นตลอดเวลา เพราะในช่วงการรับส่งข้อมูลระหว่างหน่วยความจำกับแชนแนลนั้น แชนแนลจำเป็นต้องใช้บัสข้อมูลและแอดเดรสบัสเท่านั้น เช่น ซีพียูกำลังคำนวณฟังก์ชันทางคณิตศาสตร์ ซึ่งไม่จำเป็นต้องดึงข้อมูลจากหน่วยความจำไปใช้งาน ในช่วงเวลานี้ บัสข้อมูลและแอดเดรสบัสไม่ถูกใช้งานโดยซีพียู ดังนั้นการทำ DMA จึงสามารถทำได้ และเมื่อใดที่ซีพียูต้องการใช้หน่วยความจำ การทำ DMA จะต้องหยุดการทำงาน รอจนกว่าซีพียูจะหยุดใช้งาน บัสข้อมูลและแอดเดรสบัส จึงจะสามารถทำ DMA ต่อได้ การทำงานที่ DMA ใช้บัสทั้งสองในขณะที่ซีพียูไม่ใช้งานเรียกว่า การขโมยรอบเวลา (Cycle Stealing)

### 7.8 ระบบย่อยรับ-แสดงผล (Input Output Subsystems)

Kernel ในระบบปฏิบัติการเป็นส่วนที่ติดกับฮาร์ดแวร์ สามารถให้บริการอุปกรณ์ต่างๆ ได้หลายชนิดด้วยกัน ภายใน Kernel จะมีระบบย่อยอินพุต/เอาต์พุต (I/O Subsystem) ซึ่งเป็นตัวสำคัญในการให้บริการอุปกรณ์ต่างๆ ดังนี้ (น.ท. ไพศาล โมลิสกุลมงคล และคนอื่น ๆ, 2545)



### 7.8.1 การจัดเวลาอินพุต/เอาต์พุต (I/O Scheduling)

I/O Scheduling เป็นส่วนหนึ่งของ Kernel ซึ่งช่วยจัดลำดับขั้นตอนต่างๆ ให้กับซีพียู เพื่อให้การทำงานของซีพียูเต็มประสิทธิภาพ การจัดการขั้นตอนที่ดีจะช่วยเพิ่มประสิทธิภาพ ของระบบได้เป็นอย่างดีโดยไม่ต้องเสียค่าใช้จ่ายแต่อย่างใด จะขอยกตัวอย่างการอ่านบล็อกข้อมูลจากดิสก์ สมมุติว่ามีโปรเซสที่ต้องการทำอยู่ 3 โปรเซสด้วยกัน โปรเซสที่ 1 บล็อกข้อมูลอยู่ตอนท้ายของโปรเซส ส่วนที่ 2 บล็อกข้อมูลอยู่ตอนต้นของโปรเซส ส่วนโปรเซสที่ 3 บล็อกข้อมูลอยู่ตอนกลางของโปรเซส ถ้าระบบทำงานตามลำดับโปรเซสที่ 1, 2, 3 จะพบว่าอ่านจะต้องเลื่อนไหว่ที่ไปมาเป็นอย่างมาก แต่ถ้าจัดลำดับเสียใหญ่ เป็นโปรเซสที่ 3, 2, 1 หัวอ่านจะเคลื่อนที่น้อยลงมากนอกจากประสิทธิภาพของระบบจะสูงขึ้นแล้ว หัวอ่านยังไม่เสียพลังงานในการเคลื่อนที่และไม่สึกหรองง่าย

### 7.8.2 การใช้บัฟเฟอร์

บัฟเฟอร์เป็นหน่วยความจำประเภทหนึ่งที่ใช้เก็บข้อมูลขณะที่มีการโอนข้อมูลกันระหว่างอุปกรณ์ทั้งสองชนิด ที่มาจากการสร้างบัฟเฟอร์มีเหตุผล 3 ประการ คือประการแรกสำหรับอุปกรณ์ที่มีความเร็วที่แตกต่างกันมากๆ จะช่วยให้อุปกรณ์ที่มีความเร็วมากไม่ต้องเสียเวลาคอย ตัวอย่างเช่น ไฟล์ที่ข้อมูลที่อ่านจากโมเด็มซึ่งมีความเร็วช้ากว่าดิสก์ ถ้าสามารถอ่านข้อมูลจากโมเด็มมาเก็บไว้ในหน่วยความจำหลักซึ่งใช้เป็นบัฟเฟอร์ก่อนแล้วเขียนลงในดิสก์ครั้งเดียวจะช่วยให้การทำงานเร็วยิ่งขึ้น ในกรณีของโมเด็มอาจจะมีบัฟเฟอร์ 2 ชุดด้วยกัน ชุดแรกเมื่อทยอยอ่านข้อมูลจากโมเด็มจนเต็มแล้วเขียนลงดิสก์ ในระหว่างเขียนลงดิสก์นั้นจำเป็นต้องใช้บัฟเฟอร์อีกชุดหนึ่งเพื่อเก็บข้อมูลที่กำลังอ่านจากโมเด็มอย่างต่อเนื่องข้อมูลจะได้ไม่มีการสูญเสีย

ประเภทที่สองสำหรับการใช้บัฟเฟอร์เพื่อปรับปรุงอุปกรณ์ที่มีความสามารถในการถ่ายโอนข้อมูลที่มีขนาดต่างกันให้สามารถทำงานควบคู่กันได้อย่างมีประสิทธิภาพ

ประการสุดท้ายเพื่อใช้สำหรับการทำแบ็คอัพข้อมูลซึ่งมักจะใช้เป็นประจำในคอมพิวเตอร์ เป็นการป้องกันการสูญเสยข้อมูล ทำให้อุปกรณ์ต่างๆ เป็นอิสระในขณะที่มีการจัดสรรเนื้อที่บนดิสก์ให้เป็นระเบียบ นอกจากนั้นยังนำไปใช้ในการทำหน่วยความจำเสมือนทำให้ระบบมีประสิทธิภาพสูงขึ้น



### 7.8.3 การใช้แคช

แคช คือหน่วยความจำประเภทหนึ่งที่มีความเร็วสูงมาก ประโยชน์ของการใช้แคช การเข้าถึงข้อมูลในแคชสามารถทำได้รวดเร็วกว่าในดิสก์ จุดประสงค์ของการใช้แคชสำหรับอุปกรณ์มีจุดประสงค์เดียวกันกับบัฟเฟอร์ คือไว้สำหรับทำเบ็คอัพ แต่เนื่องจากแคชทำงานได้เร็วกว่า สำหรับงานบางประเภทการใช้แคชจะดีกว่า แต่เนื่องจากแคชมีราคาแพงกว่าบัฟเฟอร์มาก ดังนั้นการทำเบ็คอัพเนื้อที่มากๆ จึงจำเป็นต้องใช้บัฟเฟอร์แทน

### 7.8.4 สพูลลิงและการสงวนดีไวซ์ (Spooling and Device Reservation)

สพูล (spool) คือบัฟเฟอร์ชนิดหนึ่งสำหรับการเก็บข้อมูลหลายๆ ไ่ว้รวมกัน ตัวอย่างของสพูลที่ถูกนำมาใช้ได้แก่ เครื่องพิมพ์ โดยเฉพาะระบบที่ใช้เครื่องพิมพ์ร่วมกันในขณะส่งพิมพ์ ระบบปฏิบัติการจะใช้สพูลลิงเข้ามาจัดการข้อมูล แล้วนำไปเก็บในสพูลตามลำดับที่ส่งเข้ามา สพูลจะเป็นตัวที่ทำหน้าที่จัดคิวให้กับเครื่องพิมพ์ไม่ให้ปะปนกัน ขณะทำงานต่างๆ เก็บอยู่ในสพูล ผู้ใช้สามารถควบคุมผ่านระบบปฏิบัติการ ให้สามารถยกเลิกการพิมพ์งานใดๆ ได้ หรือแม้กระทั่งจัดลำดับการพิมพ์ใหม่ได้เช่นกัน

### 7.8.5 Error Handling

ตัวระบบปฏิบัติการเองที่ทำหน้าที่ประสานการติดต่อระหว่างอุปกรณ์กับแอปพลิเคชันต่างๆ ของผู้ใช้ ทำให้ระบบปฏิบัติการมีความสามารถในการป้องกันความเสียหายที่อาจจะเกิดขึ้นได้ สำหรับอุปกรณ์โดยเฉพาะดิสก์ หลายครั้งที่เกิดความล้มเหลวในขณะอ่านข้อมูลจากดิสก์ ซึ่งมาจากหลายสาเหตุ เช่น อุปกรณ์ชำรุด ระบบคอมพิวเตอร์เกิดโอเวอร์โหลดขึ้น หรือแม้กระทั่งตัวควบคุมดิสก์เกิดขัดข้อง ทำให้ไม่สามารถอ่านข้อมูลได้ ระบบปฏิบัติการสามารถช่วยประสานให้ดิสก์ทำงานได้ด้วยวิธีต่างๆ เช่น สั่งให้อ่านข้อมูลใหม่ ถ้าอ่านไม่ได้ระบบปฏิบัติการจะฟ้องข้อผิดพลาดของการทำงานออกมา ผู้ใช้จึงจะดำเนินการแก้ไขตามคำแนะนำของระบบ อย่างไรก็ตามถ้าเกิดปัญหาจากอุปกรณ์ หรือแอปพลิเคชัน ระบบปฏิบัติการสามารถฟ้องข้อผิดพลาดได้ แต่ถ้าเกิดขึ้นในตัวระบบปฏิบัติการเองเราไม่สามารถแก้ไขปัญหานี้ได้

ตามหลักการโดยทั่วไป เมื่ออุปกรณ์เรียก System call ระบบจะส่งค่าขนาด 1 บิตกลับมาเพื่อเป็นการตอบรับตัวอย่างเช่นในระบบปฏิบัติการ UNIX ถ้าค่าที่ส่งกลับมาเป็น “errno” แสดงว่ามีข้อผิดพลาดเกิดขึ้น นอกจากนี้ยังมีรหัสที่แสดงค่าต่างๆ กว่า 100 รหัส ซึ่งระบบจะมีข้อแนะนำและ



แนวทางในการแก้ไขให้ผู้ใช้ได้ปฏิบัติตาม ในทางที่ถูกต้องระบบสามารถบอกถึงสาเหตุของระบบได้โดยการโต้ตอบระหว่างผู้ใช้กับระบบปฏิบัติการ และระหว่างระบบปฏิบัติการกับอุปกรณ์ เช่น ถ้าผู้ใช้แสดงคำสั่งไม่ถูกต้องระบบสามารถตรวจสอบได้จากโปรแกรม แล้วดึงคำสั่งที่ไม่ถูกต้องขึ้นมาแสดงให้ผู้ใช้ทราบ ในตัวอย่าง ข้อผิดพลาดในตัวอุปกรณ์เช่น จุดตรวจสอบ จุดเชื่อมต่อต่างๆ ระบบสามารถตรวจสอบได้จากไค์เวอร์ว่ามีข้อผิดพลาดเกิดขึ้นที่ใดบ้าง (วิชญ์ พลายมาศ, 2552)

## บทสรุป

การจัดการอุปกรณ์ที่ใช้ในการรับและแสดงผลข้อมูลมีความสำคัญและจำเป็น โดยระบบปฏิบัติการเป็นผู้จัดการอุปกรณ์เหล่านั้น ให้สามารถรับและแสดงผลข้อมูลได้อย่างมีประสิทธิภาพ หน้าที่พื้นฐานของระบบปฏิบัติการที่มีต่ออุปกรณ์ได้แก่ การติดตามสถานะของอุปกรณ์ กำหนดการใช้อุปกรณ์ จัดสรรอุปกรณ์ และ เรียกคืนอุปกรณ์ ในการรับส่งข้อมูลระหว่างซีพียูกับอุปกรณ์ต่างๆ นั้น อาศัยวิธีการหลายวิธี เช่น Polling, IRQ, Mail Box และ DMA และภายใน Kernel ของระบบปฏิบัติการจะมีระบบย่อยอินพุต/เอาต์พุต (I/O Subsystem) ซึ่งเป็นตัวสำคัญในการให้บริการอุปกรณ์ต่างๆ





---

### คำถามท้ายบทที่ 7

1. ระบบปฏิบัติการมีหน้าที่ในการจัดการอุปกรณ์อย่างไรบ้าง
2. จงอธิบายความแตกต่างของอุปกรณ์ชนิด Character Device และ Block Device
3. จงอธิบายหน้าที่ของตัวควบคุมอุปกรณ์
4. จงอธิบายหน้าที่ของตัวขับอุปกรณ์
5. จงอธิบายการรับส่งข้อมูลระหว่างซีพียูกับอุปกรณ์แบบ IRQ
6. จงอธิบายการรับส่งข้อมูลระหว่างซีพียูกับอุปกรณ์แบบ DMA

### ปฏิบัติการ

ให้นักศึกษาทำปฏิบัติการที่ 6

เรื่อง ศึกษาวิธีการจัดการอุปกรณ์ Input/Output ของระบบปฏิบัติการ Windows

โดยดาวน์โหลดแบบฟอร์มปฏิบัติการได้ที่ <http://roongrote.crru.ac.th/CT2403.html>



## เอกสารอ้างอิง

- น.ท.ไพศาล โมลิตกุลมงคล, ผศ.น.ท.ดร.ประสงค์ ปราณีตพลกรัง, น.ต.เมธา สุนทรสารทูล,  
น.ต.สุชาติ วีรกุลวัฒนา, ร.ท.อนุโชติ วุฒิพรพงษ์. (2545). **ระบบปฏิบัติการ (Operating Systems)**. กรุงเทพฯ : ไทยเจริญการพิมพ์.
- นริรัตน์ นิยมไทย. (2549). **ระบบปฏิบัติการ**. กรุงเทพฯ : ศูนย์ส่งเสริมวิชาการ.
- พิรพร หมุนสนิท, สุธีพงศาสกุลชัย และอัจจิมา เลี้ยงอยู่. (2553). **ระบบปฏิบัติการ (Operating Systems)**. กรุงเทพฯ : เคทีพี.
- วิษณุ ปลายมาศ. (2552). **ระบบปฏิบัติการ (Operating Systems)**. กรุงเทพฯ : ซีเอ็ดดูเกชั่น.  
(2552). [Online]. Available HTTP:  
<http://www.kroah.com/log/2006/05/24/>.
- (2552). [Online]. Available HTTP:  
<http://cgi.ebay.com.my/ws/eBayISAPI.dll?ViewItem&item=280349097850>.
- (2552). [Online]. Available HTTP:  
<http://www.srisamut.com/index.php?lay=show&ac=article&Id=5337331>.
- (2552). [Online]. Available HTTP:  
<http://pirun.ku.ac.th/~b4904281/page2.html>.
- (2552). [Online]. Available HTTP:  
[http://www.mahamodo.com/server/R300\\_5\\_IntelChipControl.JPG](http://www.mahamodo.com/server/R300_5_IntelChipControl.JPG).
- (2552). [Online]. Available HTTP:  
[http://www.windownetworking.com/articles\\_tutorials/interupt.html](http://www.windownetworking.com/articles_tutorials/interupt.html).