

# บทที่ 1

## คอมพิวเตอร์ทำงานได้อย่างไร

### วัตถุประสงค์การเรียนรู้

อธิบายหลักการทำงานของคอมพิวเตอร์ในระดับไฮาร์ดแวร์ ซอฟต์แวร์ และระบบปฏิบัติการ

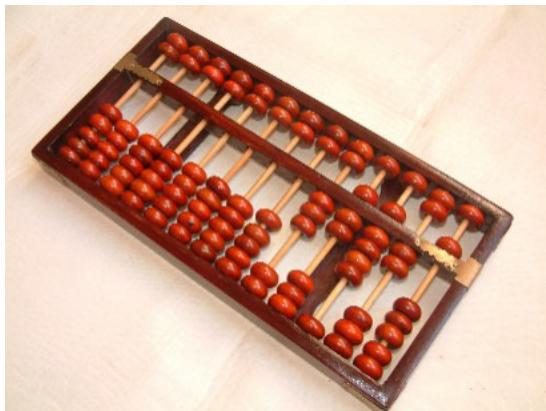
1. อธิบายวิวัฒนาการและความเป็นมาในการพัฒนาคอมพิวเตอร์ได้
2. อธิบายโครงสร้างคอมพิวเตอร์ส่วนขยายและหน้าที่ของแต่ละองค์ประกอบได้
3. รู้จักซอฟต์แวร์ชนิดต่างๆ และยาgardแวร์ที่มักพบในคอมพิวเตอร์ ระบุหน้าที่และความสัมพันธ์กับส่วนอื่นๆ ของคอมพิวเตอร์ได้
4. คาดเดาสาเหตุของปัญหาการทำงานของคอมพิวเตอร์ซึ่งเกิดจากยาgardแวร์หรือซอฟต์แวร์ได้

ใช้เวลา 3 คาบ

คอมพิวเตอร์มีพัฒนาการอย่างยาวนานจากอดีตมาจนปัจจุบัน และยังคงมีการพัฒนาอย่างต่อเนื่อง จุดมุ่งหมายหลักในการพัฒนาคอมพิวเตอร์นั้น คือการสร้างเครื่องทุนแรงมุ่งยึดในการคำนวณต่างๆ คอมพิวเตอร์ในปัจจุบันเป็นอุปกรณ์อิเล็กทรอนิกส์ ทำงานด้วยไฟฟ้า ภายใต้ระบบด้วยชิ้นส่วนต่างๆ เป็นจำนวนมากทำงานประสานกันโดยมีชุดคำสั่งควบคุม ซึ่งชับช้อนมากหากเทียบกับคอมพิวเตอร์ในยุคแรก การศึกษาวิวัฒนาการของคอมพิวเตอร์นับแต่อดีตจนปัจจุบัน จะทำให้เข้าใจกลไกการทำงานของคอมพิวเตอร์มากยิ่งขึ้น รวมถึงเห็นแนวทางการพัฒนาคอมพิวเตอร์ต่อไปในอนาคต

### 1.1 วิวัฒนาการของคอมพิวเตอร์

วิวัฒนาการของคอมพิวเตอร์จำแนกออกอย่างคร่าวๆ ตามเทคโนโลยีที่ใช้ในการสร้างเครื่องได้เป็น ยุคของเครื่องคิดเลขรูปแบบต่างๆ ยุคคอมพิวเตอร์เชิงกล ยุคคอมพิวเตอร์กึ่งไฟฟ้ากึ่งเชิงกล ยุคหลอดสูญญากาศ และยุคทรานซิสเตอร์ โดยสองยุคหลังเป็นยุคของคอมพิวเตอร์เชิงไฟฟ้า



รูปที่ 1.1: ลูกคิด

รูปที่ 1.2: เครื่องคิดเลขของชิกการ์ด<sup>1</sup>รูปที่ 1.3: เครื่องคิดเลขของปาสคาล<sup>2</sup>

### 1.1.1 เครื่องคิดเลข

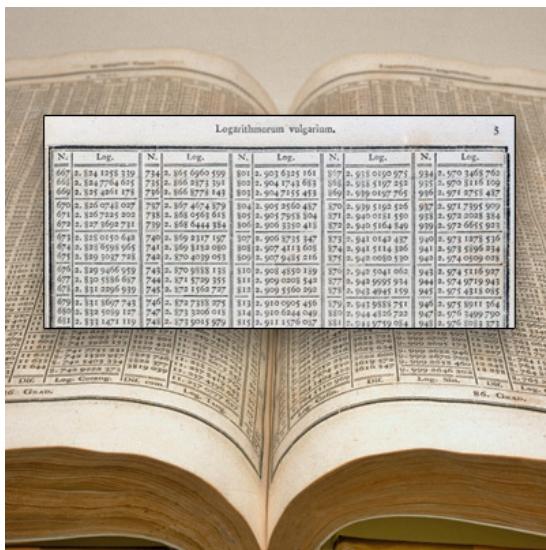
คอมพิวเตอร์ แปลตรงตัวหมายถึง เครื่องคำนวณ จุดเริ่มต้นในการสร้างคอมพิวเตอร์ขึ้นมาจึงใช้เพื่อการคำนวณต่างๆ เครื่องคำนวณอย่างง่ายที่สุดคือ เครื่องคิดเลข ยุคแรกๆ นั้นนับบัญชีได้ถึงยุคของลูกคิด (รูปที่ 1.1) ซึ่งมีใช้กันอย่างแพร่หลายมานานนับพันปี และยังคงใช้กันอยู่ในปัจจุบัน ลูกคิดเป็นอุปกรณ์เพื่อช่วยในการคำนวณหรือทดลอง ซึ่งยังต้องอาศัยคนในการติดลูกคิด แต่การคิดเลขจำนวนมากๆ นั้น ถึงแม้ลูกคิดจะช่วยทุ่นแรงในการทดลอง แต่ก็ยังต้องใช้เวลาและแรงคนในการติดลูกคิดอยู่

การทดลองนั้นมีแบบแผนที่ซัดเจนแน่นอน เราจึงสามารถสร้างกลไกเพื่อทดลองอัตโนมัติได้ เทคโนโลยีในยุคเริ่มต้นคือกลไกการหมุนของเพ่อง เครื่องคิดเลขของชิกการ์ด (รูปที่ 1.2) เป็นเครื่องคิดเลขเชิงกล เครื่องแรกที่มีการบันทึกไว้ในประวัติศาสตร์ อาศัยกลไกการหมุนของเพ่องในการบวกลบ ร่วมกับกระดูกของนาเปียร์ (Napier's Bone) ในการคูณหาร ชิกการ์ดเสนอเครื่องคิดเลขนี้เพื่อช่วยในการคำนวณทางตารางศาสตร์ ตัวเครื่องจริงนั้นถูกทำลายหรือสูญหายไปแล้ว ส่วนเครื่องจำลองซึ่งสร้างจากการออกแบบโดยชิกการ์ดนั้นตั้งอยู่ในพิพิธภัณฑ์หลายแห่ง และมีเครื่องหนึ่งอยู่ที่องค์การพิพิธภัณฑ์วิทยาศาสตร์แห่งชาติ จังหวัดปทุมธานี

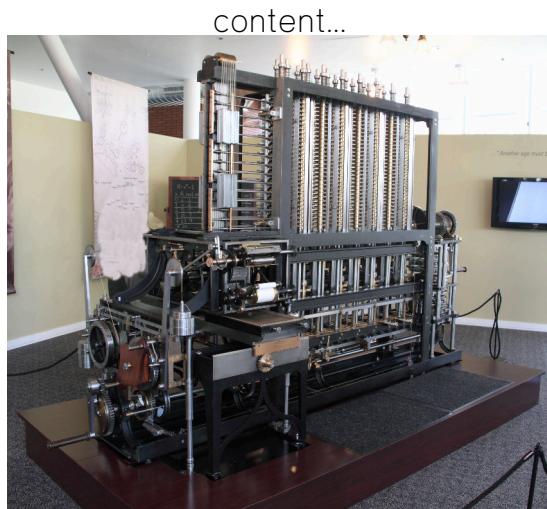
ความต้องการในการคำนวณที่ซับซ้อนนอกจากงานวิทยาศาสตร์ในกลุ่มตารางศาสตร์แล้ว การคิด

<sup>1</sup>โดย Herbert Klaerens [CC-BY-SA-3.0], ผ่าน Wikimedia Commons

<sup>2</sup>โดย © 2005 David Monniaux , [CC BY-SA 3.0], ผ่าน Wikimedia Commons



รูปที่ 1.4: ตารางล็อกและภาพขยายหน้าในหนังสือ<sup>3</sup>



รูปที่ 1.5: Difference engine<sup>4</sup>

คำนวนด้านการเงิน ภาษี ก็จะเป็นภาระที่หนัก เช่นเดียวกัน แบลร์ ปาสคาล (Blair Pascal) ออกแบบ และสร้างเครื่องคิดเลขขึ้นใช้ในสำนักงานเพื่อคำนวนภาษี เครื่องคิดเลขของปาสคาล (รูปที่ 1.3) ใช้เพื่องานในกระบวนการบัญชี ทำการคูณหารโดยอาศัยการบวกลบ นอกจากปาสคาลจะผลิตเครื่องคิดเลข เพื่อใช้ในสำนักงานของตนแล้ว ปาสคาลยังจดสิทธิบัตรและสร้างเครื่องคิดเลขเพื่อจำหน่ายเชิงพาณิชย์ อีกด้วย

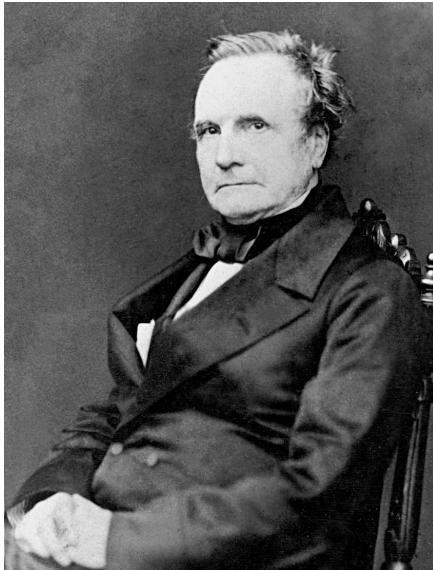
### 1.1.2 คอมพิวเตอร์เชิงกล (Mechanical computer)

เครื่องคิดเลขในอดีตนั้นสามารถทำการคำนวนพื้นฐาน ได้แก่ บวก ลบ คูณ หาร ได้ แต่จากความก้าวหน้า ทางวิทยาศาสตร์และวิศวกรรม นอกจากการด้านการค้าและบริการ ยังมีความต้องการตารางเพื่อใช้ในการคำนวนค่าต่างๆ ทางคณิตศาสตร์ ที่ซับซ้อน เช่น ตารางล็อก (รูปที่ 1.4) ตารางฟังก์ชันโพลีโนเมียล และอื่นๆ

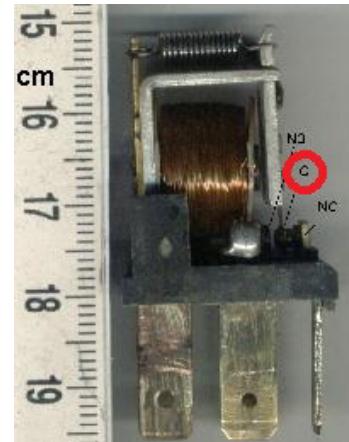
ในอดีต ตารางเหล่านี้สร้างจากการคำนวนโดยมนุษย์ ส่งโรงพิมพ์เพื่อเรียงพิมพ์ และพิมพ์ออกมานเป็นเล่ม ซึ่งมีโอกาสผิดพลาดในขั้นตอนการเรียงพิมพ์ได้มาก เนื่องจากตารางนั้นมีเฉพาะตัวเลข ไม่เหมือนกับข้อความซึ่งสามารถพิสูจน์อักษรได้ง่ายกว่า ชาร์ลส แบบบจ (Charles Babbage, รูปที่ 1.6) จึงเสนอโมเดลของ Difference engine ซึ่งเป็นเครื่องพิมพ์ที่สามารถพิมพ์ตารางค่าทางคณิตศาสตร์ได้โดยอัตโนมัติ การทำงานของ Difference engine นั้นใช้หลักการทodorobของเพื่องต่างๆ ในการคิดเลข เรียกคอมพิวเตอร์ที่ใช้หลักการทำงานของกลไกเชิงกลแบบนี้ว่า “คอมพิวเตอร์เชิงกล”

<sup>3</sup>โดย Georg von Vega, 1794 จาก Science Museum

<sup>4</sup>โดย Canticle at en.wikipedia [CC-BY-SA-3.0 or GFDL], จาก Wikimedia Commons



รูปที่ 1.6: ชาร์ลส์ แบบเบจ: บิดาของคอมพิวเตอร์<sup>5</sup>



รูปที่ 1.7: รีเลย์<sup>6</sup>

Difference engine นั้นไม่เคยถูกสร้างจนสำเร็จใช้งานได้จริงในยุคของแบบเบจเนื่องจากข้อจำกัดทางด้านงบประมาณและเทคโนโลยี ราวกว่าปี 1990 จึงมีการสร้าง Difference engine ซึ่งทำงานได้จริง โดยอาศัยร่างแบบของแบบเบจและเป็นการยืนยันว่าแนวคิดของแบบเบจนั้นใช้งานได้จริง ปัจจุบัน Difference engine ที่ทำงานได้จริงตั้งอยู่ที่พิพิธภัณฑ์วิทยาศาสตร์ ณ กรุงลอนדון (รูปที่ 1.5) และพิพิธภัณฑ์ประวัติศาสตร์คอมพิวเตอร์ ณ เมืองเมาน์เทนวิว สหรัฐอเมริกา

นอกจากโมเดลของ Difference engine แล้ว แบบเบจยังเสนอโมเดลของ Analytical engine ซึ่งถือเป็นโมเดลคอมพิวเตอร์เครื่องแรกที่สามารถโปรแกรมหรือสั่งให้ทำงานได้ต่างๆ ได้ตามต้องการ แต่จนปัจจุบัน ก็ยังไม่มีการสร้างเครื่องคอมพิวเตอร์ที่ทำงานได้จริงจากร่างการออกแบบ Analytical engine ของแบบเบจ อย่างไรก็ดี จากผลงานการออกแบบนี้ทำให้แบบเบจได้รับการยกย่องให้เป็น “บิดาของคอมพิวเตอร์”

ในอดีต เครื่องคำนวนแต่ละเครื่องนั้นจะถูกออกแบบมาเพื่อการคำนวนเฉพาะอย่าง เช่น การคำนวนทางพิชิต การคำนวนค่าตารางล็อก เป็นต้น ทำให้เครื่องที่สร้างขึ้นนั้นสามารถทำงานได้เพียงอย่างเดียว หากคอมพิวเตอร์สามารถถูกปรับแต่งให้ทำงานได้หลากหลายตามแต่ผู้ใช้ต้องการได้ ก็จะทำให้คอมพิวเตอร์มีความสามารถมากขึ้น ใช้งานได้หลากหลายสถานการณ์ขึ้น รวมถึงประยุกต์ค่าใช้จ่ายโดยรวมในการสร้างเครื่องเพื่อทำงานหลาย ๆ อย่างมากขึ้นด้วย

<sup>5</sup>Public Domain, <https://commons.wikimedia.org/w/index.php?curid=15392737>

<sup>6</sup>ตัดแปลงจาก Public Domain, <https://commons.wikimedia.org/w/index.php?curid=186888>

### 1.1.3 คอมพิวเตอร์เชิงไฟฟ้า

ปัญหาของเครื่องคำนวนเชิงกลคือความเร็วในการทำงานซึ่งติดข้อจำกัดที่รับการหมุนของกลไก ทำให้ทำความเร็วได้ไม่มาก จอร์จ สติบิตซ์ (George Stibitz) จากเบลล์แล็บ (Bell Labs) จึงออกแบบและสร้างคอมพิวเตอร์โดยใช้หลักการทำงานของรีเลย์

รีเลย์ (รูปที่ 1.7) เป็นสวิตช์รูปแบบหนึ่งซึ่งมีสองสถานะคือ ปิด และ เปิด ทำงานโดยใช้หลักการแม่เหล็กไฟฟ้า อย่างไรก็ได้ รีเลย์ก็ยังมีส่วนแขนสวิตช์หน้าล้มผัสด (แขน C ในรูปที่ 1.7) ซึ่งเป็นการทำงานเชิงกลอยู่ด้วย คอมพิวเตอร์ที่อาศัยรีเลย์ในการทำงานจึงเป็นคอมพิวเตอร์กึ่งเชิงกลกึ่งเชิงไฟฟ้า (electro-mechanical computer) สถานะติด (วงจรปิด-กระแสไหล) และดับ (วงจรเปิด-กระแสไม่ไหล) ของสวิตช์สามารถแทนได้ด้วยเลขฐานสอง 1 หลัก นั่นคือ 0 แทนสถานะปิด และ 1 แทนสถานะเปิด การออกแบบการคำนวนของคอมพิวเตอร์ซึ่งอาศัยรีเลย์ในการทำงานจึงอิงกับเลขฐานสองเป็นหลัก เรียกจำนวนข้อมูลที่เก็บด้วยเลขฐานสอง 1 หลักนี้ว่า จำนวน 1 บิต (Binary digit - bit)

รีเลย์เองยังมีข้อจำกัดด้านความเร็วจากชิ้นส่วนเชิงกลในอุปกรณ์ ทำให้ทำความเร็วในการประมวลผลได้ไม่สูง ยุคถัดมาของคอมพิวเตอร์จึงเปลี่ยนจากการใช้สวิตช์ซึ่งทำจากรีเลย์มาเป็นหลอดสูญญากาศ (รูปที่ 1.8) แทน หลอดสูญญากาศนี้ไม่มีส่วนเชิงกล การทำงานเป็นเชิงไฟฟ้าทั้งหมด จึงทำงานได้เร็วกว่ารีเลย์มาก ตัวอย่างของหลอดสูญญากาศ เช่น หลอดไดโอด

ข้อเสียของหลอดสูญญากาศคือความร้อนสูง กินไฟ อายุการใช้งานสั้น มีขนาดใหญ่ จึงมีการพัฒนาทรานซิสสเตอร์ขึ้นมา ทรานซิสเตอร์ (รูปที่ 1.9) มีลักษณะการทำงานเป็นสวิตช์สองสถานะ เช่นเดียวกับรีเลย์และหลอดสูญญากาศ มีขนาดเล็ก กินไฟต่ำ ทำให้การสร้างคอมพิวเตอร์ขนาดใหญ่ทำได้ง่ายขึ้น เทคโนโลยีคอมพิวเตอร์จึงพัฒนาต่อมาอย่างรวดเร็ว

เทคโนโลยีในยุคถัดๆ มาจนถึงปัจจุบันนี้เป็นการลดขนาดของทรานซิสเตอร์ และการรวมทรานซิสเตอร์หลายๆ ตัวเข้าเป็นวงจรรวม (Integrated Circuit: IC) หรือชิป (chip) หรือไมโครชิป (microchip) ดังรูปที่ 1.10 ซึ่งทำให้คอมพิวเตอร์มีขนาดเล็กลง แต่มีสมรรถนะการคำนวนสูงขึ้น



รูปที่ 1.8: หลอดสูญญากาศ<sup>7</sup>



รูปที่ 1.9: ทรานซิสเตอร์

ปัจจุบันนี้ก็ยังคงเป็นยุคของเทคโนโลยีทรานซิสเตอร์อยู่ แต่อยู่ในรูปของวงจรรวมขนาดใหญ่ (Very Large Scale Integration: VLSI) เทคโนโลยีปัจจุบัน (พ.ศ.2558) ทำงานจรรยาที่ใช้ทรานซิสเตอร์ขนาดหลักลิบนาโนเมตร ในชิปหนึ่งตัวประกอบด้วยทรานซิสเตอร์นับพันล้านตัว ทำให้คอมพิวเตอร์ปัจจุบันมีขนาดเล็กกว่าในยุคแรกในขณะที่มีประสิทธิภาพการคำนวนสูงกว่ามาก ทั้งนี้ ทรานซิสเตอร์

<sup>7</sup>โดย Chemical Heritage Foundation [CC-BY-SA-3.0], ผ่าน Wikimedia Commons

ในวงจรรวมก็ยังคงทำหน้าที่เป็นสวิตซ์ปิดเปิดเช่นเดียวกับในรีเลย์หรือหลอดสูญญากาศในยุคก่อน การทำงานกับคอมพิวเตอร์จึงยังอาศัยเลขฐานสองเป็นหลักเช่นเดิม

ปัจจุบันนี้ขนาดของทรานซิสเตอร์เล็กลงเรื่อยๆ แต่กำลังจะถึงขีดจำกัดในไม่ช้า เทคโนโลยีการสร้างคอมพิวเตอร์ที่เริ่มได้รับความสนใจคือคอมพิวเตอร์ควบคุมตั้ม โดยอาศัยหลักการทำงานตามพิสิกส์ในการทำงาน ซึ่งแตกต่างจากสวิตซ์ที่ใช้ในคอมพิวเตอร์เชิงไฟฟ้าในปัจจุบัน จุดเด่นของคอมพิวเตอร์ควบคุมตั้มคือสามารถคำนวณค่าต่างๆ ได้รวดเร็วกว่าระบบคอมพิวเตอร์ในปัจจุบันมาก แต่เทคโนโลยีควบคุมตั้มคอมพิวเตอร์ปัจจุบันยังอยู่ในห้องทดลอง ยังไม่พร้อมจะนำมาใช้งานจริง



รูปที่ 1.10: วงจรรวม<sup>8</sup>

## ลองคิด

จากแนวโน้มการเปลี่ยนแปลงของเทคโนโลยีคอมพิวเตอร์จากการแบบเชิงกลมาเป็นระบบไฟฟ้า และอนาคตอาจจะเป็นระบบควบคุมตัมนั้น นิสิตคิดว่า หลังจากระบบควบคุมตัมแล้ว จะมีการเปลี่ยนแปลงอีกหรือไม่ เพราะเหตุใด

## 1.2 เครื่องคอมพิวเตอร์ที่สำคัญในประวัติศาสตร์

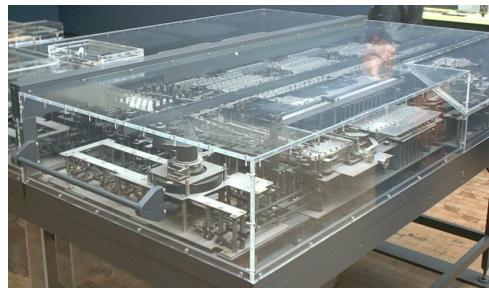
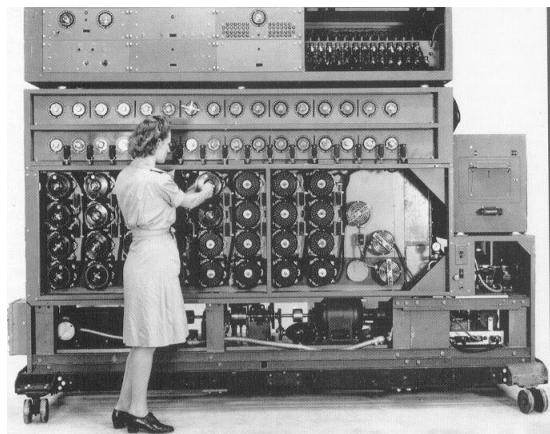
เครื่องคอมพิวเตอร์ที่สำคัญในประวัติศาสตร์มักเป็นเครื่องที่แสดงจุดเปลี่ยนทางเทคโนโลยีที่สำคัญๆ เช่น เทคโนโลยีการสร้างเครื่อง จากกลไกเชิงกล เป็นกลไกเชิงไฟฟ้า หรือมีความสามารถในการปรับแต่งเพื่อทำงานได้หลากหลาย เรียกว่า สามารถโปรแกรมได้

การปรับแต่งเครื่องคอมพิวเตอร์เพื่อให้ใช้งานได้หลากหลายอาจทำได้ในหลายรูปแบบ เช่น ปรับเปลี่ยนชิ้นส่วนสำหรับการทำงานเฉพาะอย่าง สลับสายการเชื่อมต่ออุปกรณ์ต่างๆ หรือใช้ชุดรหัสคำสั่งบรรจุไว้ในหน่วยเก็บหรือหน่วยความจำ เป็นต้น

### 1.2.1 Z1

คอนราด ซูส (Konrad Zuse) ออกแบบและสร้าง Z1 ซึ่งเป็นคอมพิวเตอร์เชิงกลที่สามารถโปรแกรมได้แบบจำกัด Z1 เป็นคอมพิวเตอร์ซึ่งสร้างโดยเงินสนับสนุนของครอบครัวซูส ซึ่งต่างจากเครื่องคอมพิวเตอร์อื่นในยุคสมัยส่วนใหญ่ที่ใช้กระดาษ punched tape ในการเขียนโปรแกรม ซูสสร้าง Z1 ขึ้นใหม่อีกครั้งหลังสงครามโลก ปัจจุบัน Z1 ที่สร้างขึ้นใหม่จัดแสดงอยู่ที่พิพิธภัณฑ์เทคโนโลยีเยอรมัน ณ กรุงเบอร์ลิน (รูปที่ 1.11)

<sup>8</sup>โดย Zephyris ที่ en.wikipedia [GFDL หรือ CC-BY-SA-3.0], จาก Wikimedia Commons

รูปที่ 1.11: Z1 ที่สร้างขึ้นใหม่<sup>9</sup>รูปที่ 1.12: Enigma<sup>10</sup>รูปที่ 1.13: เครื่อง Bombe ของสหราชอาณาจักร<sup>11</sup>

นอกจาก Z1 แล้ว ชูสังกอกแบบและสร้าง Z2 และ Z3 ตามมาด้วย โดยที่ Z3 เป็นคอมพิวเตอร์กึ่งไฟฟ้ากึ่งเชิงกลและเป็นคอมพิวเตอร์เครื่องแรกที่สามารถโปรแกรมได้โดยสมบูรณ์หลังการปรับแต่งบางอย่าง แต่ Z3 ก็ถูกทำลายลงจากการทิ้งระเบิดในระหว่างสงครามโลกครั้งที่ 2 เช่นกัน

### 1.2.2 Bombe กับ Enigma และ Colossus กับ Lorenz SZ

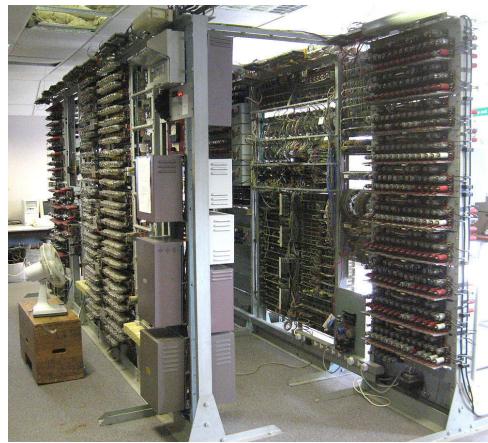
ช่วงสงครามโลกนั้นความต้องการทางการทหารเป็นแรงผลักดันที่ทำให้เทคโนโลยีคอมพิวเตอร์พัฒนาไปอย่างรวดเร็ว ปัจจัยหนึ่งที่ลุ้นผลต่อการแพ็คชนะในสงครามคือข้อมูล การรับส่งข้อมูลทางการทหารระหว่างฝ่ายเดียวกันนั้นจำเป็นต้องเข้ารหัสเพื่อป้องกันความลับรั่วไหล ในทางกลับกัน หากสามารถถอดรหัสของฝ่ายตรงข้ามได้ ก็มีแนวโน้มที่จะชนะได้มากขึ้น

ฝ่ายเยอรมนีใช้แบบวิธีการเข้ารหัสและสร้างเครื่องเข้ารหัสกึ่งไฟฟ้ากึ่งเชิงกลชื่อ Enigma (รูปที่ 1.12) มีลักษณะเป็นเหมือนพิมพ์ดีด เมื่อกดแป้นอักขระตามปกติแล้ว จะได้ข้อความที่เข้ารหัสแล้วพิมพ์

<sup>9</sup>โดย Stahlkocher [GFDL หรือ CC-BY-SA-3.0], ผ่าน Wikimedia Commons

<sup>10</sup>โดย Greg Goebel (Web page Image) [Public domain], ผ่าน Wikimedia Commons

<sup>11</sup>จาก CryptoMuseum.com ที่ <http://www.cryptomuseum.com/crypto/bombe/>



รูปที่ 1.14: Colossus ที่สร้างขึ้นใหม่<sup>12</sup>

นอกจาก หลักการเข้ารหัสของ Enigma คือการสลับอักขระจริงกับอักขระอื่น แต่รูปแบบการสลับนั้นเปลี่ยนไปเรื่อยๆ ในแต่ละครั้งของการส่งข้อมูล การถอดรหัสด้วยมือจึงเป็นเรื่องยุ่งยากและใช้เวลามาก อีกทั้งยังมีข้อความใหม่ๆ ที่ต้องถอดรหัสกุลส่งมาตลอดเวลา

ฝ่ายสัมพันธมิตรจึงพยายามสร้างเครื่องช่วยถอดรหัสอัตโนมัติขึ้น เครื่อง Bombe เป็นเครื่องถอดรหัสกึ่งไฟฟ้ากึ่งเชิงกล ออกแบบโดยอลัน ทัวริง (Alan Turing) เพื่อใช้ในการถอดรหัส Enigma หลักการทำงานของเครื่อง Bombe คือ ทดลองหาวิธีการสลับอักขระของ Enigma ไปเรื่อยๆ ว่าเป็นไปได้หรือไม่ การหาวิธีการเข้ารหัสนั้นสามารถทำได้ด้วยมือคนก็จริง แต่จะใช้เวลานานมากเนื่องจากจำนวนรูปแบบที่เป็นไปได้มีจำนวนมาก การใช้ Bombe มาช่วยจำลองการเข้ารหัสเพื่อทดสอบจะย่นเวลาการหารหัสที่ถูกต้องได้มาก เครื่อง Bombe ถูกสร้างขึ้นหลายเครื่องโดยสมาชิกของฝ่ายสัมพันธมิตร รูปที่ 1.13 เป็นตัวอย่างของเครื่อง Bombe ที่สร้างโดยสหรัฐอเมริกา

นอกจาก Enigma แล้วฝ่ายเยอรมนียังมีเครื่องเข้ารหัสชุด Lorenz SZ อีกด้วย ซึ่งเข้ารหัสด้วยการนำข้อความมา XOR กับกุญแจ และถอดรหัสด้วยวิธีเดียวกัน ฝ่ายสัมพันธมิตรพัฒนาเครื่อง Colossus ขึ้น เพื่อช่วยในการถอดรหัสชนิดนี้ โดยที่ Colossus เป็นคอมพิวเตอร์อิเล็กทรอนิกส์เต็มตัว และยังเป็นคอมพิวเตอร์อิเล็กทรอนิกส์เครื่องแรกที่สามารถโปรแกรมได้ แม้จะทำได้อย่างจำกัดก็ตาม ข้อมูลของ Colossus ถือเป็นความลับทางการทหาร แผนการอุบัติเหตุและตัวเครื่อง Colossus ส่วนมากจึงถูกแยกชิ้นส่วนหรือทำลายหลังจากจบสงคราม Colossus ถูกสร้างขึ้นใหม่ภายหลังสงครามสงบ (รูปที่ 1.14) ปัจจุบันจัดแสดงอยู่ที่พิพิธภัณฑ์การคอมพิวเตอร์แห่งชาติ ณ บักกิงแฮมเชอร์ สาธารณรัฐอเมริกา

### 1.2.3 ENIAC

ENIAC (รูปที่ 1.15) เป็นที่รู้จักในฐานะคอมพิวเตอร์อิเล็กทรอนิกส์ที่สามารถโปรแกรมได้โดยสมบูรณ์ เครื่องแรกของโลก ENIAC ถูกออกแบบมาเพื่อคำนวณตารางสำหรับการยิงปืนใหญ่ในช่วงสงครามโลก

<sup>12</sup>โดย MaltaGC [GFDL, CC-BY-SA-3.0 หรือ FAL], ผ่าน Wikimedia Commons



รูปที่ 1.15: ENIAC<sup>13</sup>

ครั้งที่ 2 ส่วนอิเล็กทรอนิกส์ของ ENIAC เป็นหลอดสูญญากาศ จึงทำให้ทำงานได้เร็วกว่าคอมพิวเตอร์รุ่นก่อนๆ ซึ่งอาศัยเทคโนโลยีรีเลย์

ทั้ง ENIAC และ Colossus เป็นคอมพิวเตอร์อิเล็กทรอนิกส์ที่สามารถโปรแกรมได้ แต่ Colossus นั้นสามารถโปรแกรมได้จำกัดกว่า ENIAC

### ลองคิด

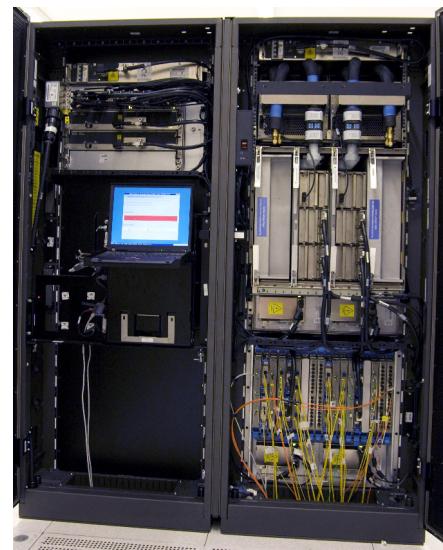
หากคอมพิวเตอร์ที่สำคัญคือเครื่องที่เป็นจุดเปลี่ยนต่างๆ ทั้งด้านเทคโนโลยีและวิธีการใช้งาน เครื่องที่น่าจะเป็นคอมพิวเตอร์ที่สำคัญในช่วง 20 ปีที่ผ่านมานี้มีเครื่องอะไรบ้าง จงให้เหตุผลประกอบ

## 1.3 ประเภทของคอมพิวเตอร์

การจำแนกประเภทของคอมพิวเตอร์ทำได้หลายรูปแบบ ในประวัติศาสตร์คอมพิวเตอร์อาจจะแบ่งตามเทคโนโลยีที่ใช้ เช่น รีเลย์ หลอดสูญญากาศ ทรานซิสเตอร์ สำหรับคอมพิวเตอร์ในยุคปัจจุบัน อาจแบ่งตามลักษณะการใช้งาน คือเครื่องคอมพิวเตอร์ส่วนบุคคล เช่น คอมพิวเตอร์ตั้งโต๊ะ คอมพิวเตอร์แบบพกพา และคอมพิวเตอร์ขนาดใหญ่สำหรับการประมวลผลจำนวนมาก เช่น เครื่องคอมพิวเตอร์ที่ใช้ในการพยากรณ์อากาศ ซึ่งเรียกชนิดของคอมพิวเตอร์ต่างๆ ที่นำเสนอ มีดังต่อไปนี้

1. เครื่องบริการ (server) เป็นการเรียกคอมพิวเตอร์ที่ทำหน้าที่ให้บริการผู้ใช้ หรือคอมพิวเตอร์อื่นๆ ที่เชื่อมต่อเข้ามา การให้บริการเป็นได้หลากหลายรูปแบบ เช่น เว็บไซต์ แฟ้มข้อมูล จดหมายอิเล็กทรอนิกส์ เครื่องบริการหนึ่งเครื่องสามารถให้บริการได้มากกว่าหนึ่งบริการ เครื่องบริการ

<sup>13</sup>โดย ช่างภาพกองทัพสหรัฐอเมริกา, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=978770>

รูปที่ 1.16: เครื่องบริการ<sup>14</sup>รูปที่ 1.17: เมนเฟรม IBM Z9<sup>15</sup>

สำหรับบางงาน เช่น เว็บไซต์ธุรกิจ ไม่จำเป็นต้องมีประสิทธิภาพการประมวลผลสูงนัก โดยทั่วไปใช้คำว่า “เครื่องบริการ” ในสองความหมาย หนึ่งคือเป็นเครื่องที่ให้บริการ ไม่ว่ารูปแบบใดก็ตาม ของเครื่องนั้นจะเป็นคอมพิวเตอร์ตั้งโต๊ะ หรือรูปแบบอื่นก็ตาม อีกความหมายหนึ่งของเครื่องบริการคือตัวยาardแวร์ที่ออกแบบมาเฉพาะเพื่อใช้เป็นเครื่องบริการ เช่นรูปที่ 1.16

2. เมนเฟรม (mainframe) หมายถึงเครื่องคอมพิวเตอร์ขนาดใหญ่ รองรับผู้ใช้งานจำนวนมากได้พร้อมกัน มีความแม่นยำเชื่อถือได้สูง จึงถูกใช้ในการประมวลผลธุกรรมในระบบธนาคารและธุรกิจขนาดใหญ่ เช่นเครื่อง IBM Z9 (รูปที่ 1.17) เราอาจมองว่าเมนเฟรมเป็นเครื่องบริการขนาดใหญ่ได้ เพราะเมนเฟรมมีหน้าที่บริการประมวลผลธุกรรมให้ผู้ใช้ แต่ขนาดของตัวเครื่องเมนเฟรมมักจะใหญ่กว่าเครื่องบริการโดยทั่วไปมาก
3. ซูเปอร์คอมพิวเตอร์ (supercomputer) หมายถึงคอมพิวเตอร์ขนาดใหญ่ สมรรถนะการประมวลผลสูงมาก เช่น IBM Blue Gene/P (รูปที่ 1.18) ที่สถาบันวิจัยแห่งชาติอาร์กอนน์ ซูเปอร์คอมพิวเตอร์มักถูกใช้ในการประมวลผลทางวิทยาศาสตร์และวิศวกรรมศาสตร์ เช่น การพยากรณ์อากาศ การสร้างแบบจำลองทางวิทยาศาสตร์ ซูเปอร์คอมพิวเตอร์เป็นเครื่องคอมพิวเตอร์ขนาดใหญ่เช่นเดียวกับเมนเฟรม แต่จุดมุ่งหมายในการใช้งานต่างกัน เมนเฟรมต้องการรองรับการทำธุรกรรมจำนวนมาก จึงมีการติดต่อกับลูกค้า เช่น เขียนข้อมูลสูง รองรับผู้ใช้งานจำนวนมาก แต่อาจไม่จำเป็นต้องมีการคำนวณมาก ในขณะที่ซูเปอร์คอมพิวเตอร์อาจไม่จำเป็นต้องอ่านเขียนมากนัก แต่ต้องคำนวณค่าได้รวดเร็ว
4. คอมพิวเตอร์ส่วนบุคคล (Personal Computer - PC) เป็นคอมพิวเตอร์เพื่อใช้งานทั่วไป เช่น คอมพิวเตอร์ตั้งโต๊ะ และคอมพิวเตอร์รวมตั้งก็ มีขนาดไม่ใหญ่มาก ราคาไม่สูง เหมาะสมสำหรับ

<sup>14</sup>โดย Rodzilla ที่ en.wikipedia [GFDL, CC-BY-SA-3.0 หรือ CC-BY-2.5], จาก Wikimedia Commons

<sup>15</sup>โดย Ing. Richard Hilber (Self-photographed) [Public domain], ฝ่าย Wikimedia Commons



รูปที่ 1.18: ซูเปอร์คอมพิวเตอร์ IBM Blue Gene/P<sup>16</sup>

การใช้งานคนเดียว คอมพิวเตอร์ส่วนบุคคลแบ่งออกเป็นสองรูปแบบ ได้แก่ แบบไม่เคลื่อนที่ เช่น คอมพิวเตอร์ตั้งโต๊ะ เน็ตท็อป เป็นต้น และแบบพกพาได้ มีขนาดเล็ก น้ำหนักเบา มีแบตเตอรี่ ในตัว เช่น คอมพิวเตอร์ว่างตัก (laptop, notebook) แท็บเล็ตพีซี (tablet pc) เป็นต้น

5. คอมพิวเตอร์ฝังตัว (embedded computer) เป็นคอมพิวเตอร์ขนาดเล็กที่ถูกออกแบบมาเพื่อ ทำงานเฉพาะทาง และติดตั้งอยู่ในอุปกรณ์อื่น เช่น หน่วยควบคุมเครื่องยนต์ (Engine Control Unit - ECU) ในรถยนต์ ระบบการทำงานในหม้อหุงข้าว ตู้เย็น เป็นต้น

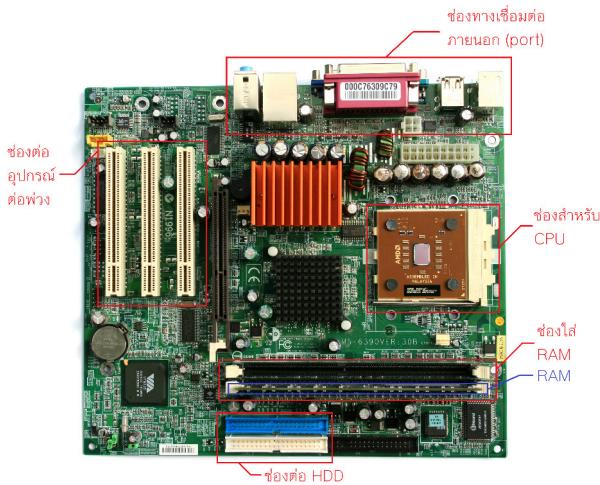
## ลองคิด

สมาร์ตโฟน นับเป็นคอมพิวเตอร์หรือไม่ หากเป็นควรจัดอยู่ในหมวดใดข้างต้น หรือควรแยกเป็นหมวด ใหม่เฉพาะ

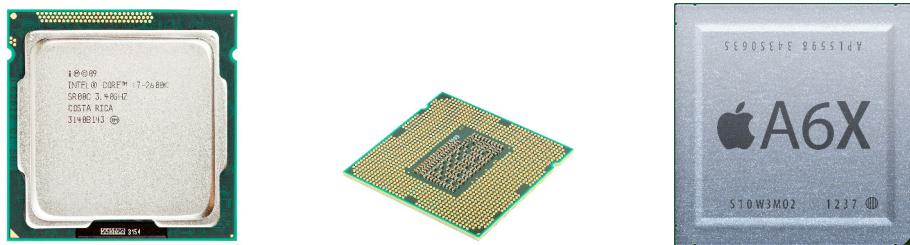
## 1.4 องค์ประกอบและการทำงานของชาร์ดแวร์คอมพิวเตอร์ในปัจจุบัน

ระบบคอมพิวเตอร์ในปัจจุบันแบ่งออกเป็นสองส่วน ได้แก่ ชาร์ดแวร์ และ ซอฟต์แวร์ โดยชาร์ดแวร์คือ อุปกรณ์ต่างๆ ที่จับต้องได้ และซอฟต์แวร์คือรหัสคำสั่งเพื่อการทำงานต่างๆ การแยกชาร์ดแวร์ออกจาก ซอฟต์แวร์เพื่อให้การออกแบบคอมพิวเตอร์สำหรับการทำงานตามความต้องการเฉพาะทางต่างๆ ลดความซับซ้อน และให้ซอฟต์แวร์เป็นตัวคำสั่งระบุขั้นตอนวิธีในการทำงาน และให้ชาร์ดแวร์มีหน้าที่ทำงานคำสั่งต่างๆ เหล่านั้นโดยใช้สัญญาณไฟฟ้า

<sup>16</sup>โดย Argonne National Laboratory ที่ Flickr [CC-BY-SA-2.0], ผ่าน Wikimedia Commons



รูปที่ 1.19: ແຜງງຈຮລັກຂອງຄອມພິວເຕອຣ໌ຕັ້ງໂດຍອຽມດາ<sup>17</sup>



รูปที่ 1.20: ທ່າງປະລາດຄວາມປັບປຸງຂອງອິນເທັດ (ຊ້າຍແລກລາງ<sup>18</sup>) ແລະ ແອປເປີລ (ຂວາ<sup>19</sup>)

ຍັງແລ້ວເວັບໄຊຂອງຮະບບຄອມພິວເຕອຣ໌ໄໝວ່າຈະເປັນນາດໃໝ່ຫຼືເລື້ອມມັກຈະມີແຜງງຈຮລັກ (mainboard, motherboard) ເຊັ່ນຮູບທີ 1.19 ເປັນແກນໜັກ ແລ້ວປະກອບອຸປະກອນສ່ວນຕ່າງໆ ຫຼືເລື້ອມຕ່ອງອຸປະກອນ ອື່ນໆ ກັບແຜງງຈຮລັກ ອຸປະກອນທີ່ສໍາຄັນບັນແຜງງຈຮລັກ ໄດ້ແກ່ ທ່າງປະລາດຄວາມປັບປຸງຂອງອິນເທັດ ແລະ ທ່າງປະລາດ ເພີ້ມ ເຊິ່ງມາຕ່ອຳພ່ານພອർຕໍ່ຫຼືອ່ານທາງການເຊື່ອມຕ່ອງວ່າ ອຸປະກອນຕ່ອຳພ່າງຫຼືອ່ານອຸປະກອນຮົບຂ້າງ (peripheral device)

#### 1.4.1 ທ່າງປະລາດຄວາມປັບປຸງ (Central Processing Unit - CPU)

ທ່າງປະລາດຄວາມປັບປຸງ (ຮູບທີ 1.20) ເປັນຫ້າໃຈໃນການທຳການຂອງຄອມພິວເຕອຣ໌ ມີຫ້າທີ່ຄິດຄຳນວນຕາມຄຳສົ່ງແລະ ຄວບຄຸມການທຳການປະສານກັນຂອງທ່າງປະລາງຕ່າງໆ ໃນຮະບບຄອມພິວເຕອຣ໌ ທ່າງປະລາດຄວາມປັບປຸງໃນ

<sup>17</sup>ດັດແປລັງຈາກ Jonathan Zander [GFDL, CC-BY-SA-3.0 ຫຼື CC-BY-SA-2.5], ຜ່ານ Wikimedia Commons

<sup>18</sup>ໂດຍ Eric Gaba [CC-BY-SA-3.0], ຜ່ານ Wikimedia Commons

<sup>19</sup>ໂດຍ Henriok - Own work, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=22310746>

ปัจจุบันใช้เทคโนโลยีรวมขนาดใหญ่ ในชิปหนึ่งตัวประกอบด้วยทรานซิสเตอร์จำนวนมากซึ่งทำงานร่วมกัน

หน่วยประมวลผลกลางประกอบด้วยหน่วยการทำงานสองหน่วยอยู่ ได้แก่

1. หน่วยควบคุม (Control Unit) มีหน้าที่ควบคุมการทำงานประสานกันของหน่วยต่างๆ ในเครื่อง
2. หน่วยคำนวณและตรรกะ (Arithmetic Logic Unit - ALU) มีหน้าที่คิดคำนวณต่างๆ เช่น การบวก ลบ คูณ หาร เปรียบเทียบจำนวนมากกว่า น้อยกว่า เท่ากับ ไม่เท่ากับ เป็นต้น

สำหรับเครื่องคอมพิวเตอร์อย่างง่ายอาจจะมีหน่วยประมวลผลกลางเพียงหน่วยเดียว แต่ในปัจจุบันที่การคำนวณต่างๆ ซับซ้อนมากขึ้นเรื่อยๆ ความต้องการในการประมวลผลมากขึ้น ในคอมพิวเตอร์หนึ่งเครื่องอาจมีหน่วยประมวลผลกลางได้มากกว่าหนึ่งหน่วย ในแต่ละหน่วยอาจมีแกน (core) ในการประมวลผลมากกว่าหนึ่งแกน

สถาปัตยกรรมคอมพิวเตอร์ หมายถึง รายละเอียดองค์ประกอบในระบบคอมพิวเตอร์ และลักษณะการทำงานร่วมกันของอุปกรณ์เหล่านั้น สถาปัตยกรรมคอมพิวเตอร์ของหน่วยประมวลผลกลางที่ใช้กันมากในปัจจุบัน เช่น

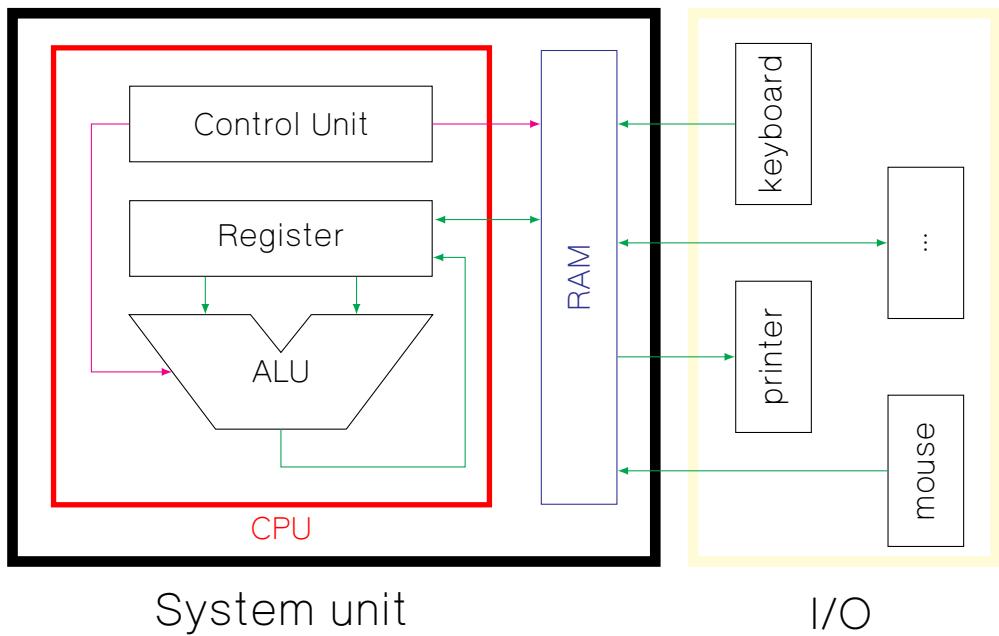
- สถาปัตยกรรม x86 ออกแบบโดย Intel ผลิตโดยผู้ผลิตเช่น Intel และ AMD ใช้กันทั่วไปในคอมพิวเตอร์ส่วนบุคคล
- สถาปัตยกรรม ARM ออกแบบโดย ARM ผลิตและนำไปตัดแปลงก่อนผลิตโดยผู้ผลิตหลากหลาย เช่น Samsung Qualcomm Apple มากใช้ในอุปกรณ์พกพา สมาร์ตโฟน

สถาปัตยกรรมที่ต่างกันนี้มีวิธีการทำงานที่ต่างกัน ซอฟต์แวร์ซึ่งทำงานบนเครื่องซึ่งมีสถาปัตยกรรมที่ต่างกันนี้จะต่างกันตามไปด้วย

สถาปัตยกรรมคอมพิวเตอร์ในปัจจุบันส่วนใหญ่เป็นสถาปัตยกรรมฟอนนอยมันน์ (Von Neumann architecture, รูปที่ 1.21) ซึ่งใช้หน่วยความจำหลักเก็บทั้งข้อมูลและคำสั่งที่ต้องการให้คอมพิวเตอร์ประมวลผล

### เครื่อง 32 บิตและ 64 บิตคืออะไร

สำหรับคอมพิวเตอร์ในปัจจุบัน การทำงานของหน่วยประมวลผลกลางคือการอ่านคำสั่งรวมถึงค่าต่างๆ ที่ถูกเก็บไว้ในหน่วยความจำหลัก และทำงานตามคำสั่งไปเรื่อยๆ การเข้าถึงหน่วยความจำหลักทำได้โดยการอ้างอิงถึงเลขที่อยู่ (address) ของข้อมูลนั้นๆ ในหน่วยความจำ เราจึงสามารถจำแนกหน่วยประมวลผลกลางตามขนาดของเลขที่อยู่ของหน่วยความจำหลักของเครื่องนั้นๆ ได้ด้วย เช่น เครื่อง 32 บิต หมายถึง เครื่องนี้ใช้เลขที่อยู่ของหน่วยความจำหลักขนาด 32 บิต (มีพื้นที่ในหน่วยความจำ  $2^{32}$  ตำแหน่ง) เป็นต้น หากเลขที่อยู่ของหน่วยความจำมีขนาดใหญ่ขึ้น แสดงว่าหน่วยประมวลผลกลางสามารถเข้าถึงหน่วยความจำหลักโดยตรงได้มากขึ้น หากเป็นการประมวลผลข้อมูลขนาดใหญ่ เครื่องที่เข้าถึงที่อยู่จำนวนมากได้จะประมวลผลได้เร็วกว่าเครื่องที่ใช้ที่อยู่ขนาดเล็กกว่า เนื่องจากหากที่อยู่มี



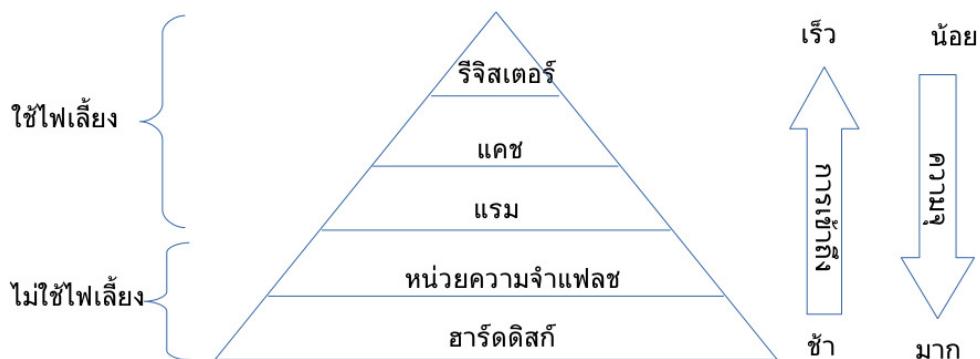
รูปที่ 1.21: สถาปัตยกรรมฟอนนอยมันน์

ขนาดเล็กแล้วจะจำเป็นต้องมีการเก็บโปรแกรมหรือข้อมูลบางส่วนในหน่วยความจำภายในของ การเข้าถึงก็จะทำได้ช้าลง

นอกจากการประมวลผลข้อมูลจำนวนมากแล้ว การออกแบบหน่วยประมวลผลกลางจะออกแบบให้ขนาดของจำนวนเต็มและทศนิยมโดยทั่วไปหนึ่งจำนวนใช้พื้นที่ในหน่วยความจำเท่ากับขนาดของเลขที่อยู่เพื่อประสิทธิภาพในการคำนวณของหน่วยประมวลผล เมื่อเปรียบเทียบเครื่อง 32 บิตกับเครื่อง 64 บิต จึงทำให้เครื่อง 64 บิตซึ่งเก็บข้อมูลทศนิยมได้ละเอียดกว่ามีนัยสำคัญหรือความแม่นยำในการคำนวณสูงกว่าเครื่อง 32 บิตด้วย

### การวัดความเร็วของคอมพิวเตอร์

ผู้ใช้มักกบอกร่วมกันว่าคอมพิวเตอร์ทำงานได้ “เร็ว” โดยวัดจากความเร็วในการตอบสนองต่อคำสั่งต่างๆ ที่ผู้ใช้สั่ง คอมพิวเตอร์อีกทรอนิกส์นั้นทำงานตามลัญญาณนาฬิกาของหน่วยประมวลผลกลาง โดยในหนึ่งรอบของลัญญาณนาฬิกาอาจจะทำได้หนึ่งคำสั่ง หรือมากน้อยกว่านั้นก็ได้ตามแต่การออกแบบ การวัดความเร็วในการทำงานของคอมพิวเตอร์อย่างคร่าวๆ อาจเทียบได้จากความเร็วของลัญญาณนาฬิกา เช่น หน่วยประมวลผลกลางมีความความถี่ลัญญาณนาฬิกา 2.8GHz หมายถึง ทำงานได้ 2.8 พันล้านรอบต่อวินาที ก็จะทำงานได้เร็วกว่าหน่วยประมวลผลกลางซึ่งมีความถี่ลัญญาณนาฬิกาน้อยกว่านี้ อย่างไรก็ตาม จำนวนคำสั่งที่หน่วยประมวลผลกลางทำงานได้ในหนึ่งรอบของลัญญาณนาฬิกานั้นอาจต่างกันไปตามการออกแบบ ในทางทฤษฎี การวัดความเร็วในการทำงานของคอมพิวเตอร์จึงวัดจากจำนวนคำสั่งที่หน่วยประมวลผลกลางทำงานในหนึ่งหน่วยเวลา หน่วยการวัดสมรรถนะนี้เรียกว่า MIPS (Million Instructions Per Second) คือการนับจำนวนคำสั่งที่หน่วยประมวลผลกลางสามารถทำงานได้ในหนึ่ง



รูปที่ 1.22: ลำดับชั้นของหน่วยความจำภายในหน่วยระบบ

วินาที ไม่ว่าจะเป็นคำสั่งการคำนวน หรือคำสั่งอ่านเขียนข้อมูล

ในคอมพิวเตอร์ขนาดใหญ่ที่มีการประมวลผลสูงมากนั้น งานส่วนใหญ่จะเป็นการคำนวนเลขทศนิยม การเปรียบเทียบสมรรถนะการคำนวนของคอมพิวเตอร์ขนาดใหญ่จึงมักจะนับเป็นจำนวนคำสั่งคำนวน เลขทศนิยมที่ทำได้ในหนึ่งวินาที เรียกว่า FLOPS (Floating-point Operations Per Second) แทน

หน่วยประมวลผลกลางของคอมพิวเตอร์ส่วนบุคคลทั่วไปในปัจจุบัน (พ.ศ. 2558) สามารถทำคำสั่งทศนิยมได้ 4 คำสั่งใน 1 รอบสัญญาณนาฬิกา หากมีแกนประมวลผล 1 แกน และความถี่สัญญาณนาฬิกา 2.5GHz คอมพิวเตอร์เครื่องนี้จะมีประสิทธิภาพในทางทฤษฎี  $4 \times 2.5 \times 10^9 = 10$  GFLOPS เป็นต้น

ซูเปอร์คอมพิวเตอร์ที่มีความเร็วสูงในปัจจุบัน (มิถุนายน 2559) คือเครื่อง Sunway TaihuLight ของประเทศจีน ประกอบด้วยแกนประมวลผลกว่าล้านแกน ทำความเร็วสูงสุดได้ถึง  $93.01 \times 10^{15}$  FLOPS ( $93.01$  PFLOPS)<sup>20</sup>

## 1.4.2 หน่วยความจำภายใน (Internal memory)

ในการประมวลผลต่างๆ จำเป็นต้องมีหน่วยความจำเป็นเสริมที่เก็บข้อมูลรวมถึงเก็บคำสั่งต่างๆ ที่จะส่งให้หน่วยประมวลผลและตระกากใช้งาน หน่วยความจำในคอมพิวเตอร์แบ่งโดยความไกล์ชิดกับหน่วยประมวลผลกลางออกเป็นลำดับชั้นดังรูปที่ 1.22

หน่วยความจำที่เข้าถึงได้โดยตรงจากหน่วยประมวลผลกลางเรียกว่า หน่วยความจำปฐมภูมิ หรือ หน่วยความจำภายใน แบ่งได้เป็นสองประเภท คือ

1. หน่วยความจำที่อยู่ภายในหน่วยประมวลผลกลาง ได้แก่ รีจิสเตอร์ (register) และ แคช (CPU cache)
2. หน่วยความจำหลัก เป็นหน่วยความจำที่อยู่นอกหน่วยประมวลผลกลาง ได้แก่ แรม (RAM)

<sup>20</sup> <https://www.top500.org/lists/2016/06/>

หน่วยความจำที่ไม่สามารถเข้าถึงได้โดยตรงจากหน่วยประมวลผลกลาง แต่ยังอยู่ในหน่วยระบบของคอมพิวเตอร์ เรียกว่า หน่วยความจำทุติยภูมิ นอกจากนี้ยังมีหน่วยความจำสำรองที่ไม่ได้อยู่ในหน่วยระบบอีกด้วย

ในตอนนี้จะกล่าวถึงเฉพาะหน่วยความจำภายใน การทำงานกับหน่วยความจำของคอมพิวเตอร์ทำเป็นลำดับชั้นเรียงตามความเร็วในการเข้าถึง โดยระดับบนสุดคือรีจิสเตอร์ ซึ่งมีความจุน้อยที่สุดแต่อ่านเขียนได้เร็วที่สุด แคชมีความจุมากกว่ารีจิสเตอร์ แต่การอ่านเขียนทำได้ช้ากว่า คอมพิวเตอร์สมัยใหม่อาจจะมีแคชได้หลายระดับ เช่น แคช L1 แคช L2 เป็นต้น โดยที่แคช L1 (ระดับ 1) จะเข้าถึงได้เร็วกว่าแคชในระดับต่ำกว่า รวมมีความจุมากที่สุดในหน่วยความจำสามรูปแบบนี้ แต่ก็เข้าถึงได้ช้าที่สุดเช่นกัน หากต้องการทำงานให้เร็ว ก็ควรใช้รีจิสเตอร์ให้มากที่สุด แต่หากมีข้อมูลมากเกินกว่าจะเก็บได้หมดในรีจิสเตอร์ ก็จำเป็นต้องใช้หน่วยความจำในลำดับรองๆ ลงมาเพิ่มเติม การทำงานที่รวดเร็วของคอมพิวเตอร์จึงขึ้นอยู่กับการใช้หน่วยความจำทุกระดับร่วมกันให้ได้ประสิทธิภาพสูงที่สุด



รูปที่ 1.23: RAM<sup>21</sup> ความจำเข้าถึงแบบสุ่ม แรมแบ่งออกเป็น 2 ชนิดตามเทคโนโลยีที่ใช้ใน การผลิตคือ SRAM (Static RAM) และ DRAM (Dynamic RAM) แรมที่ใช้ในคอมพิวเตอร์ส่วนบุคคลทั่วไปในปัจจุบันคือ DDR SDRAM (Double Data Rate Synchronous DRAM) ซึ่งเป็น DRAM ชนิดหนึ่ง หน่วยวัดความเร็วในการเข้าถึงแรมมักจะวัดเป็นจำนวนครั้งการถ่ายโอนข้อมูลในหนึ่งหน่วยเวลา เช่น DDR3-800E เป็น DDR แรมรุ่นที่ 3 มีอัตราการถ่ายโอนข้อมูล 800 ล้านครั้งในหนึ่งวินาที เป็นต้น

### 1.4.3 หน่วยความจำภายนอก (External memory)

หน่วยความจำภายนอกหรือหน่วยความจำทุติยภูมินั้นไม่สามารถถูกเรียกใช้โดยตรงจากหน่วยประมวลผลกลาง แต่ถูกเรียกใช้ผ่านอุปกรณ์อ่าน/เขียนข้อมูลอีกต่อหนึ่ง หน่วยความจำภายนอกเป็นหน่วยความจำชนิดไม่ลบเลื่อน (Non-volatile memory) ซึ่งไม่จำเป็นต้องมีกระแสไฟเลี้ยงเพื่อเก็บข้อมูล มักมีความจุข้อมูลสูงกว่าหน่วยความจำภายในมาก แต่ก็เรียกใช้ได้ช้ากว่ามาก เช่นกัน หน่วยความจำภายนอกจึงมักใช้เป็นแหล่งสำรองข้อมูล แต่ไม่ใช่ในการประมวลผลหากไม่จำเป็น ชนิดของหน่วยความจำภายนอกที่น่าสนใจ เช่น

- หน่วยเก็บแบบจานแม่เหล็ก เช่น ฮาร์ดดิสก์ (Hard Disk Drive - HDD, รูปที่ 1.24 (ซ้าย)) อาศัยทิศทางข้ามแม่เหล็กในการเก็บข้อมูล ในการอ่านข้อมูลจะมีหัวอ่านขยับเลื่อนไป ณ ตำแหน่งที่

<sup>21</sup>Public Domain, <https://commons.wikimedia.org/w/index.php?curid=647267>



รูปที่ 1.24: HDD (ซ้าย) และ SSD (ขวา)<sup>22</sup>



รูปที่ 1.25: แผ่นซีดี<sup>23</sup>



รูปที่ 1.26: ยูเอสบีแฟลชไดรฟ์<sup>24</sup>

เก็บข้อมูลและอ่านข้อมูลแม่เหล็กที่เก็บไว้ ณ ตำแหน่งนั้น หัวอ่านเป็นจุดประบางของฮาร์ดดิสก์ หากถูกกระแทกอาจเสียหายได้ง่าย

- จานแสง (optical disc) เช่น แผ่นซีดี รูปที่ 1.25 แผ่นดีวีดี แผ่นบลูเรย์ อาศัยหลักการสะท้อนแสงในการเก็บข้อมูล การบันทึกข้อมูลลงไปในแผ่นคือการปรับการสะท้อนแสงของพื้นผิว ณ บริเวณนั้น เพื่อให้แทนค่าบิตที่ต้องการ
- รอม (Read Only Memory - ROM) มักมีลักษณะเป็นชิปที่ถูกบัดกรีติดกับแผงวงจร เป็นหน่วยความจำซึ่งไม่สามารถแก้ไขค่าที่เก็บไว้ได้ หรือในบางกรณีอาจทำได้แต่ไม่ง่ายนัก รอมจึงมักใช้เป็นที่เก็บคำสั่งที่ใช้ในการเริ่มต้นทำงานเมื่อเปิดเครื่อง เป็นที่เก็บเฟิร์มแวร์ซึ่งเป็นคำสั่งในการจัดการฮาร์ดแวร์ของอุปกรณ์นั้นๆ เป็นต้น รอมในยุคแรกนั้นต้องกำหนดค่าข้อมูลในรอมขณะผลิต และแก้ไขค่าไม่ได้ ซึ่งไม่สะดวกในการใช้งาน จึงมีการพัฒนาให้ผู้ใช้สามารถบรรจุข้อมูลลงในรอมได้ครั้งเดียว เรียกว่า (Programmable ROM - PROM) จากนั้นพัฒนาต่อให้สามารถลบค่าที่เก็บไว้ได้โดยใช้อุปกรณ์พิเศษ เพื่ออำนวยความสะดวกในการแก้ไขค่าโดยผู้ใช้ (Erasable PROM - EPROM) และทำให้ลบค่าที่เก็บไว้ได้โดยใช้สัญญาณไฟฟ้า (Electrically Erasable PROM - EEPROM) ตามลำดับ สำหรับ EEPROM ถึงแม้ว่าจะแก้ไขค่าได้หลายครั้ง เรายังคงเรียก EEPROM ว่าเป็นหน่วยความจำแบบไม่สามารถแก้ไขค่าได้ เนื่องจากวิธีการแก้ไขค่านั้นค่อนข้างซับซ้อน
- หน่วยความจำแบบแฟลช (Flash memory) เป็นหน่วยความจำที่ลบและบันทึกใหม่ได้เช่นเดียว กับ EEPROM แต่ใช้เทคโนโลยีที่ต่างกันไป รูปแบบของหน่วยความจำแบบแฟลช เช่น
  - ยูเอสบีแฟลชไดรฟ์ (USB flash drive) ดังรูปที่ 1.26 ซึ่งมาทดแทนfloppies (floppy disk) ที่เป็นหน่วยเก็บข้อมูลแบบงานแม่เหล็ก เพราะมีความจุสูงกว่า และอ่านเขียนได้เร็วกว่า
  - เอสดีการ์ด (SD card) ใช้มากในการเก็บข้อมูลสำหรับอุปกรณ์พกพา กล้องดิจิทัล
  - โซลิดสเตตไดรฟ์ (Solid State Drive - SSD) เป็นดิสก์แบบแฟลช ใช้เป็นดิสก์ทดแทนฮาร์ดดิสก์แบบงานแม่เหล็ก จุดเด่นของดิสก์แบบแฟลชที่เหนือกว่าดิสก์แบบงานแม่

<sup>23</sup>โดย husin.sani [CC-BY-SA-2.0], ผ่าน Flickr

<sup>24</sup>โดย Ubern00b [GFDL หรือ CC-BY-SA-3.0], ผ่าน Wikimedia Commons

<sup>25</sup>โดย Ravenperch - camera photograph, CCO, <https://commons.wikimedia.org/w/index.php?curid=19319839>

เหล็กคือ การอ่านข้อมูลทำได้เร็วกว่ามาก เพราะไม่มีส่วนเชิงกลแบบหัวอ่านของฮาร์ดดิสก์ (รูปที่ 1.24) นอกจานนี้ยังลักษณะแก่การพกพา เนื่องจากไม่มีส่วนประบงแบบหัวอ่านของฮาร์ดดิสก์อยู่ แต่ข้อเสียคืออย่างทำความสะอาดจุข้อมูลได้ไม่มากเท่าดิสก์แบบงานแม่เหล็ก และมีราคาสูง

#### 1.4.4 ส่วนรับเข้า/ส่วนส่งออก (Input/Output Unit - I/O Unit)

ข้อมูลเข้าและออกมีได้หลากหลายรูปแบบ เช่น ภาพ เสียง ตัวอักษร ส่วนรับเข้าและส่งออกเป็นตัวกลาง การเชื่อมต่อกับผู้ใช้หรือระบบอื่นๆ ตัวอย่างของอุปกรณ์รับเข้า เช่น คีย์บอร์ด เม้าส์ เมาส์ปากกา เป็นต้น ตัวอย่างของอุปกรณ์ส่งออก เช่น จอภาพ เครื่องพิมพ์ ลำโพง เป็นต้น การควบคุมอุปกรณ์รับเข้าและส่งออกทำผ่านระบบปฏิบัติการและโปรแกรมขับอุปกรณ์ซึ่งเป็นซอฟต์แวร์

เนื่องจากการประมวลผลทั้งหมดของคอมพิวเตอร์จะทำในหน่วยความจำหลักเท่านั้น เมื่อต้องการรับข้อมูลจากส่วนรับเข้า หน่วยประมวลผลกลางจะส่งให้ถ่ายโอนข้อมูลจากช่องทางเชื่อมต่อไปเก็บไว้ในหน่วยความจำ ในทำนองเดียวกัน เมื่อต้องการส่งข้อมูลออกที่ส่วนส่งออก หน่วยประมวลผลกลางก็จะส่งให้นำข้อมูลจากหน่วยความจำออกไปทางช่องทางเชื่อมต่อเช่นกัน รูปแบบและวิธีการรับ/ส่ง/สื่อสารของแต่ละอุปกรณ์ถูกกำหนดโดยผู้ผลิต หรือมาตรฐานกลางต่างๆ และนำไปใช้โดยโปรแกรมขับสำหรับอุปกรณ์นั้น ๆ

#### ลองคิด

หากไม่มีหน่วยความจำภายใน ก็มีเฉพาะหน่วยประมวลผลกลาง หน่วยความจำภายใน และส่วนรับเข้า/ส่วนส่งออก คอมพิวเตอร์จะสามารถทำงานได้หรือไม่ จะให้เหตุผลประกอบ

### 1.5 ประเภทและการทำงานของซอฟต์แวร์

เนื่องจากความต้องการให้คอมพิวเตอร์ทำงานนั้นซับซ้อนขึ้นเรื่อยๆ การปรับแต่งฮาร์ดแวร์เฉพาะสำหรับการทำงานแต่ละอย่างนั้นสิ้นเปลืองมาก เมื่อมีการออกแบบคอมพิวเตอร์ให้สามารถโปรแกรมได้แล้ว การพัฒนาคอมพิวเตอร์ให้ทำงานได้ตามความต้องการก็เปลี่ยนไปจากการออกแบบฮาร์ดแวร์ใหม่ ไปเน้นที่การเขียนชุดคำสั่งให้คอมพิวเตอร์ทำงานตามที่ต้องการแทน

คำว่า โปรแกรม นอกจากจะหมายถึงการปรับแต่งโดยชุดคำสั่งให้คอมพิวเตอร์ทำงานตามต้องการแล้ว โปรแกรมยังหมายถึงตัวชุดคำสั่งที่สั่งให้เครื่องทำงานได้อีกด้วย ในความหมายหลังนี้ โปรแกรมกับซอฟต์แวร์คือสิ่งเดียวกัน

ซอฟต์แวร์แบ่งออกเป็นสองระดับ คือ ซอฟต์แวร์ระบบ และซอฟต์แวร์ทั่วไป โดยที่ซอฟต์แวร์ระบบคือซอฟต์แวร์ที่ใช้จัดการการทำงานของคอมพิวเตอร์ ทำงานใกล้ชิดกับฮาร์ดแวร์ และซอฟต์แวร์ทั่วไปคือซอฟต์แวร์ที่ออกแบบมาเพื่อตอบสนองความต้องการใช้งานของผู้ใช้โดยผ่านซอฟต์แวร์ระบบอีกต่อหนึ่ง ตัวอย่างของซอฟต์แวร์ที่นำเสนอในมีดังนี้

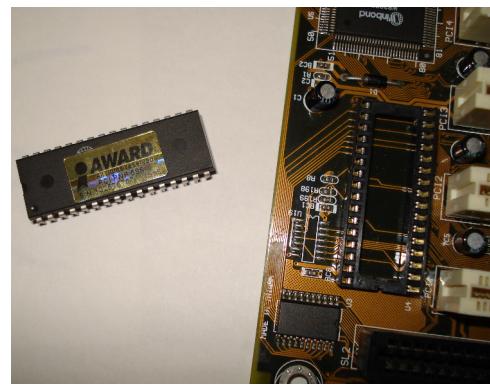
### 1.5.1 ไบอส

เมื่อเปิดเครื่องหรือกดสวิตช์รีเซ็ต เรียกว่าการทำโคลด์บูต (cold boot) ซอฟต์แวร์ตัวแรกที่เริ่มทำงานคือ ไบอส (BIOS) ซึ่งจะเริ่มจากการทดสอบหน่วยความจำ ส่วนต่อประสาน อุปกรณ์ต่างๆ ของเครื่องเมื่อ เปิดเครื่องใหม่ เรียกว่า POST (Power-On-Self-Test - POST)

หลังจากทดสอบอุปกรณ์ต่างๆ เรียบร้อย ไบอสจะหาอุปกรณ์ที่ติดตั้งระบบปฏิบัติการ และ สั่งให้เครื่องถ่ายข้อมูลจากระบบปฏิบัติการเข้าไป ในหน่วยความจำ จากนั้นคอมพิวเตอร์จะเริ่มทำงานจากระบบปฏิบัติการ ค่าต่างๆ ของไบอส ถูกเก็บอยู่ในหน่วยความจำแบบไม่ลบเลื่อนบน แผงวงจรหลัก อุปกรณ์ที่มักใช้เก็บไบอสในสมัย ก่อนนั้นใช้เทคโนโลยี CMOS จึงมักมีการเรียก หน่วยความจำที่ใช้เก็บไบอสนี้ว่า CMOS ไป ด้วย

สำหรับการสั่งเริ่มต้นทำงานใหม่ (restart)

ภายในระบบปฏิบัติการ เรียกว่าการทำウォร์มบูต (warm boot) เมื่อเครื่องเริ่มต้นทำงานใหม่ ไบอส จะไม่มีการทำ POST อีก ซึ่งจะประหยัดเวลาในการเริ่มต้นทำงานใหม่จากการทำโคลด์บูตลงมาได้พอ สมควร



รูปที่ 1.27: ชิปไบอส<sup>26</sup>

### 1.5.2 ระบบปฏิบัติการ

ระบบปฏิบัติการมีหน้าที่ติดต่อกันหน่วยประมวลผลกลางโดยตรงเพื่อให้คอมพิวเตอร์ทำงานตามต้องการ ระบบปฏิบัติการแต่ละตัวจึงออกแบบมาโดยยึดกับสถาปัตยกรรมของหน่วยประมวลผลกลาง ด้วยอย่าง ของระบบปฏิบัติการที่เป็นที่นิยม เช่น

-  **Windows** จาก Microsoft มีทั้งรุ่นสำหรับเครื่องบริการ เครื่องคอมพิวเตอร์ส่วนบุคคล และ อุปกรณ์พกพา
-  **OSX** สำหรับเครื่องคอมพิวเตอร์ส่วนบุคคลและ iOS สำหรับอุปกรณ์พกพาของ Apple
-  **Linux** มีหลากหลายดิสทริบูชัน เช่น ubuntu®, fedora®, redhat, debian
-  เป็นระบบปฏิบัติการสำหรับอุปกรณ์พกพาของ Google ซึ่งมีฐานมาจาก Linux

หน้าที่หลักของระบบปฏิบัติการคือการจัดสรรทรัพยากรสำหรับโปรแกรม หน้าที่เสริมอื่นๆ ของระบบปฏิบัติการ เช่น การจัดการการเชื่อมต่อเครือข่าย การรักษาความปลอดภัย การอำนวยความสะดวกในการต่อประสานผู้ใช้ เป็นต้น

<sup>26</sup>โดย Tremaster [CC-BY-SA-3.0], ผ่าน Wikimedia Commons

```

dittaya@rye:~$ top
top - 15:16:30 up 7:10, 2 users, load average: 0.57, 0.45, 0.37
Tasks: 293 total, 2 running, 280 sleeping, 0 stopped, 1 zombie
Cpu(s): 2.0%us, 0.9%sy, 0.0%ni, 96.9%id, 0.2%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 8077808k total, 4045236k used, 4032572k free, 304504k buffers
Swap: 8287228k total, 0k used, 8287228k free, 1748988k cached

PID USER      PR  NI  VIRT   RES   SHR   S %CPU %MEM TIME+ COMMAND
2232 dittaya  20   0 1020m 86m 20m S    6  1.1 21:11.09 chrome
1753 dittaya  9 -11  429m 7416 5208 S    4  0.1 17:35.42 pulseaudio
1974 dittaya  20   0 1098m 221m 70m S    2  2.8 15:34.32 chrome
2343 dittaya  20   0 1056m 165m 27m S    2  2.1 6:51.54 chrome
12972 dittaya 20   0 995m 128m 22m S    1  1.6 0:02.31 chrome
1030 root     20   0 366m 53m 29m S    0  0.7 7:03.22 Xorg
3187 dittaya  20   0 1784m 170m 85m S    0  2.2 1:00.88 soffice.bin
10784 root     20   0     0     0     0     S    0  0.0 0:01.15 kworker/0:0
12390 dittaya 20   0 972m 68m 28m S    0  0.9 0:01.87 chrome
1 root      20   0 24560 2500 1352 S    0  0.0 0:00.51 init
2 root      20   0     0     0     0     S    0  0.0 0:00.00 kthreadd
3 root      20   0     0     0     0     S    0  0.0 0:02.12 ksoftirqd/0
5 root      0 -20    0     0     0     S    0  0.0 0:00.00 kworker/0:0H
7 root      0 -20    0     0     0     S    0  0.0 0:00.00 kworker/u:0H
8 root      RT  0     0     0     0     S    0  0.0 0:00.00 migration/0
9 root      20   0     0     0     0     S    0  0.0 0:00.00 rCU_bh
10 root     20   0     0     0     0     S    0  0.0 0:09.01 rCU_sched

```

รูปที่ 1.28: การจัดสรรทรัพยากรและจัดลำดับการทำงานโดย Ubuntu

### การจัดสรรทรัพยากรสำหรับโปรแกรม

ทรัพยากรที่สำคัญในการประมวลผลของคอมพิวเตอร์คือหน่วยประมวลผลกลาง และหน่วยความจำระบบปฏิบัติการบางรูปแบบอาจยอมให้โปรแกรมหลาย ๆ โปรแกรมทำงานพร้อมกันได้ (multitasking) เช่น การเปิดเพลงพังในขณะที่พิมพ์งานด้วยโปรแกรมประมวลคำ ระบบปฏิบัติการจะแบ่งส่วนการใช้งานหน่วยประมวลผลกลางและหน่วยความจำให้โปรแกรมต่าง ๆ ที่กำลังทำงานอยู่ โปรแกรมที่กำลังตอบสนองต่อผู้ใช้จะเรียกว่าโปรแกรมแอ็คทีฟ (active program) หรือโปรแกรมพื้นหน้า (foreground process) ในขณะที่โปรแกรมซึ่งกำลังทำงานแต่ไม่ได้ตอบสนองโดยตรงต่อผู้ใช้ เช่นโปรแกรมเล่นเพลงในตัวอย่างข้างต้น เรียกว่าโปรแกรมพื้นหลัง (background process) นอกจากนี้ การจัดลำดับความสำคัญของงานที่ต้องทำเพื่อให้โปรแกรมต่าง ๆ ทำงานได้ลุล่วง ก็เป็นหน้าที่ของระบบปฏิบัติการ เช่นกัน

เนื่องจากคอมพิวเตอร์เครื่องหนึ่งอาจมีหน่วยประมวลผลได้หลายหน่วย (multiprocessor) ระบบปฏิบัติการจึงมีหน้าที่แบ่งงานที่ต้องทำให้กับหน่วยประมวลผลแต่ละหน่วย หรือใช้หน่วยประมวลผลหลาย ๆ หน่วยทำงานร่วมกัน นอกจากนี้ ระบบปฏิบัติการที่มีความสามารถสูงยังสามารถจัดการกับหน่วยประมวลผลกลางชนิดที่มีหลายแกนประมวลผลได้อย่างมีประสิทธิภาพ ในระบบคอมพิวเตอร์ที่รองรับผู้ใช้มากกว่าหนึ่งคนในเวลาเดียวกัน (multiuser) ระบบปฏิบัติการก็มีหน้าที่แบ่งทรัพยากรรวมถึงจัดลำดับความสำคัญของงานให้กับผู้ใช้ทุกคนด้วย

การใช้งานหน่วยความจำหลักให้มีประสิทธิภาพก็ขึ้นกับความสามารถของระบบปฏิบัติการ เช่น เดียวกัน การทำงานของคอมพิวเตอร์ในสถาปัตยกรรมฟอนโนymann ต้องมีการถ่ายโอนข้อมูลที่ต้องการใช้งานรวมถึงโปรแกรมเข้าไปอยู่ในหน่วยความจำหลักก่อนจะให้หน่วยประมวลผลกลางทำงาน กรณีที่โปรแกรมและข้อมูลมีขนาดเล็กกว่าขนาดของหน่วยความจำหลักหรือมีหน่วยความจำหลักเหลืออีกนั่นนักจะไม่มีปัญหา แต่หากโปรแกรมและข้อมูลมีขนาดใหญ่เกินกว่าที่หน่วยความจำหลักทั้งหมดจะเก็บได้ ระบบปฏิบัติการจะมีหน้าที่จัดสรรหน่วยความจำเสมือน (virtual memory) เพิ่มเติมให้ เพื่อให้โปรแกรมยังสามารถทำงานได้ หน่วยความจำเสมือนคือการใช้หน่วยความจำภายในอก เช่น ดิสก์ เป็นเสมือนหน่วยความจำหลัก แล้วใช้การลับค่า (swap) ในหน่วยความจำเลื่อนกับหน่วยความจำหลักจริงแทน เพื่อให้หน่วยประมวลผลกลางสามารถประมวลผลได้ตามปกติ แต่เนื่องจากหน่วยความจำภายในอกนั้นทำงาน

ซึ่งกว่าหน่วยความจำภายในมาก และการสับค่าก็ต้องใช้เวลา หากจำเป็นต้องใช้หน่วยความจำเสมือนคอมพิวเตอร์จะทำงานได้ช้าลงอย่างเห็นได้ชัด

นอกจากการใช้งานหน่วยประมวลผลกลางและหน่วยความจำแล้ว ระบบปฏิบัติการยังจัดลำดับการใช้งานอุปกรณ์ต่อพ่วงต่างๆ ไม่ว่าจะเป็นอุปกรณ์อ่านเขียนข้อมูล อุปกรณ์เครือข่าย หรืออุปกรณ์อื่นๆ รวมถึงช่วยอำนวยความสะดวกให้กับโปรแกรมต่างๆ ที่เรียกใช้งานอุปกรณ์นั้นๆ ด้วย ตัวอย่างการใช้งานอุปกรณ์ต่อพ่วงอย่างมีประสิทธิภาพโดยระบบปฏิบัติการ เช่น การทำงานกับเครื่องพิมพ์ โดยปกติแล้วความเร็วในการประมวลผลของหน่วยประมวลผลกลางนั้นสูงกว่าความเร็วในการพิมพ์มาก หากต้องรอนานพิมพ์เสร็จจะทำงานอื่นต่อได้ก็จะใช้หน่วยประมวลผลกลางได้ไม่เต็มประสิทธิภาพ ระบบปฏิบัติการจึงอาจกันพื้นที่ในหน่วยความจำเพื่อเป็นที่พักข้อมูลไว้ เมื่อสั่งให้พิมพ์เอกสาร ข้อมูลที่ต้องการพิมพ์จะถูกนำไปเก็บไว้ที่ที่พักข้อมูล เครื่องพิมพ์จะสามารถอ่านข้อมูลจากที่พักข้อมูลนี้ ในขณะเดียวกันหน่วยประมวลผลกลางก็สามารถนำไปทำงานอย่างอื่นต่อได้

### การเชื่อมต่อกับระบบเครือข่าย

การเชื่อมต่อกับคอมพิวเตอร์เครื่องอื่นๆ ในระบบเครือข่ายจำเป็นต้องมีพอร์ตโคลชีนเป็นข้อตกลงกลางในการสื่อสารระหว่างกัน พอร์ตโคลชีนนี้ได้รับการสนับสนุนจากระบบปฏิบัติการ ไม่ว่าเครื่องในระบบเครือข่ายจะใช้ระบบปฏิบัติการใด หากใช้พอร์ตโคลชีนเดียวกัน เครื่องเหล่านั้นจะสามารถสื่อสารกันได้

### ดูแลรักษาความปลอดภัย

เนื่องจากระบบปฏิบัติการเป็นซอฟต์แวร์ที่บริหารจัดการทรัพยากรต่างๆ ของเครื่องโดยตรง ระบบปฏิบัติการจึงอาจมีความสามารถในการอนุญาต (authorize) การเข้าถึงทรัพยากรต่างๆ เพื่อรักษาความปลอดภัยโดยรวมของระบบอีกด้วย การจัดการนี้มักจะผูกกับการตรวจสอบและยืนยันตัวตนของผู้ใช้งาน (identification and authentication) ว่ามีสิทธิ (privilege) ในการเข้าถึงทรัพยากรหรือไม่

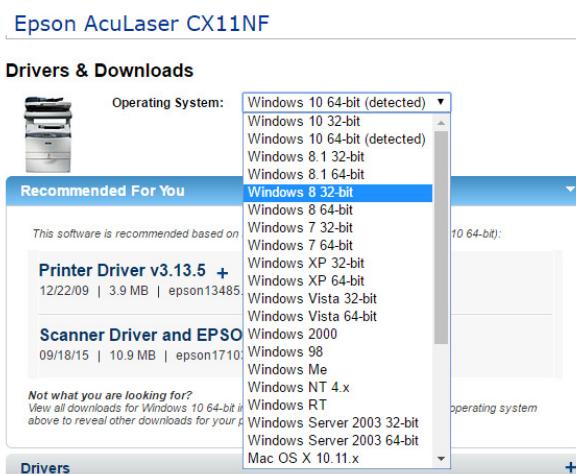
### สนับสนุนส่วนต่อประสานผู้ใช้

ส่วนต่อประสานผู้ใช้ทำหน้าที่รับคำสั่งต่างๆ จากผู้ใช้มาส่งให้ส่วนประมวลผล และแสดงผลลัพธ์คืนให้ผู้ใช้ ส่วนต่อประสานผู้ใช้แบ่งได้เป็น 2 รูปแบบ ได้แก่

- ส่วนต่อประสานแบบกราฟิก (Graphic User Interface - GUI) ผู้ใช้สั่งการผ่านภาพปุ่มต่างๆ ข้อดีคือใช้งานได้ง่าย เหมาะสำหรับผู้ใช้ทั่วไป แต่คอมพิวเตอร์ต้องแบ่งทรัพยากรมาใช้ในการสร้างรูปภาพต่างๆ ด้วย แทนที่จะใช้เพื่อการคำนวณค่าเพียงอย่างเดียว
- ส่วนต่อประสานแบบคำสั่ง (Command Line Interface - CLI) ดังตัวอย่างในรูปที่ 1.29 ผู้ใช้สั่งการผ่านแป้นพิมพ์ ข้อดีคือลั่งการได้รวดเร็ว ปรับแต่งได้ละเอียด แต่คำสั่งมักมีรายละเอียดเฉพาะตัว ทำให้ต้องใช้เวลาในการเริ่มต้นเรียนรู้มาก

```
dittaya@rye: ~
[kb]
Get:16 http://archive.ubuntu.com precise-updates/restricted amd64 Packages [13.7 kB]
Get:17 http://archive.ubuntu.com precise-updates/universe i386 Packages [248 kB]
Get:18 http://archive.ubuntu.com precise-updates/main i386 Packages [933 kB]
Get:19 http://archive.ubuntu.com precise-updates/multiverse i386 Packages [15.5 kB]
Get:20 http://archive.ubuntu.com precise-updates/restricted i386 Packages [13.7 kB]
Hit http://archive.ubuntu.com precise-updates/main TranslationIndex
Hit http://archive.ubuntu.com precise-updates/multiverse TranslationIndex
Hit http://archive.ubuntu.com precise-updates/restricted TranslationIndex
Hit http://archive.ubuntu.com precise-updates/universe TranslationIndex
Hit http://archive.ubuntu.com precise/main Translation-en
Hit http://archive.ubuntu.com precise/multiverse Translation-en
Hit http://archive.ubuntu.com precise/restricted Translation-en
Hit http://archive.ubuntu.com precise/universe Translation-en
Hit http://archive.ubuntu.com precise-updates/main Translation-en
Hit http://archive.ubuntu.com precise-updates/multiverse Translation-en
Hit http://archive.ubuntu.com precise-updates/restricted Translation-en
Hit http://archive.ubuntu.com precise-updates/universe Translation-en
Fetched 3,328 kB in 10s (315 kB/s)
Reading package lists... Done
dittaya@rye: ~
```

รูปที่ 1.29: ส่วนต่อประสานแบบคำสั่ง



รูปที่ 1.30: ตัวอย่างรุ่นของโปรแกรมขับอุปกรณ์

### 1.5.3 โปรแกรมขับอุปกรณ์ (Device driver)

การทำงานกับอุปกรณ์รอบข้างต่างๆ เช่น เครื่องพิมพ์ นั้น ระบบปฏิบัติการจะอาศัยโปรแกรมขับอุปกรณ์ ซึ่งมักจะเรียกทับศัพท์ล้วนๆ ว่า ไ/dr์ฟเวอร์ เป็นตัวกลางในการสื่อสารกับฮาร์ดแวร์ของอุปกรณ์รอบข้าง เหล่านั้น โปรแกรมขับอุปกรณ์ก็เป็นซอฟต์แวร์ระบบชนิดนึง จะเห็นได้ว่าโปรแกรมขับอุปกรณ์ขึ้นอยู่กับตัว อุปกรณ์ และยังขึ้นอยู่กับระบบปฏิบัติการอีกด้วย ดังนั้นอุปกรณ์ต่างรุ่นกันจึงอาจมีโปรแกรมขับที่ต่างกัน แม้จะผลิตโดยผู้ผลิตเดียวกัน และอุปกรณ์ขึ้นเดียวกันแต่ทำงานกับระบบปฏิบัติการที่ต่างกันอาจจำเป็นจะ ต้องใช้โปรแกรมขับที่ต่างกันเช่นกัน เช่นในรูปที่ 1.30 แสดงรุ่นต่างๆ ของโปรแกรมขับอุปกรณ์สำหรับ เครื่องพิมพ์ ซึ่งขึ้นอยู่กับระบบปฏิบัติการด้วย

### 1.5.4 ซอฟต์แวร์ทั่วไปหรือโปรแกรมประยุกต์

ซอฟต์แวร์ทั่วไปถูกออกแบบมาเพื่อตอบสนองรูปแบบการใช้งานของผู้ใช้ เช่น โปรแกรมประมวลคำ โปรแกรมตารางทำงาน โปรแกรมการนำเสนอ โปรแกรมเล่นเพลงและภาพยนตร์ เป็นต้น ซอฟต์แวร์เหล่านี้ล้วนดำเนินการให้คอมพิวเตอร์ทำงานโดยผ่านระบบปฏิบัติการอีกชั้นหนึ่ง การแยกซอฟต์แวร์ทั่วไปออกมายังทำงานบนระบบปฏิบัติการทำให้การออกแบบและเขียนซอฟต์แวร์ทำได้ละเอียดขึ้น ผู้เขียนซอฟต์แวร์ไม่จำเป็นต้องเข้าใจรายละเอียดการทำงานของฮาร์ดแวร์ เพียงอาศัยคำสั่งต่างๆ ที่ระบบปฏิบัติการมีให้ล้วนให้ทำงานตามวัตถุประสงค์ที่ต้องการก็เพียงพอ

จะเห็นว่าประสิทธิภาพการทำงานของซอฟต์แวร์ต่างๆ นั้น นокจากจะขึ้นกับการออกแบบของผู้สร้าง ซอฟต์แวร์แล้ว ยังขึ้นกับประสิทธิภาพของระบบปฏิบัติการ รวมถึงการทำงานของหน่วยประมวลผลกลาง อีกด้วย ในปัจจุบันมีความพยายามเพิ่มประสิทธิภาพการประมวลผลที่ใช้บ่อย เช่น งานการประมวลผลภาพ ด้วยการเพิ่มคำสั่งเฉพาะทางเหล่านี้ลงไปในหน่วยประมวลผลกลาง เพื่อให้การคำนวณทำได้เร็วขึ้น กว่าการคำนวณปกติทั่วไป

## ลองคิด

Macbook เครื่องหนึ่งลงระบบปฏิบัติการ Windows และ OS X ไว้ ในระบบปฏิบัติการทั้งสองมี Steam ซึ่งเป็นแพลตฟอร์มสำหรับเล่นเกมอยู่ นิสิตซื้อเกม Civilization ซึ่งเล่นได้ใน Windows แบบไม่มีปัญหา แต่เมื่อเปิดเกมเดียวกันใน OS X และ กลับพบปัญหาเกมปิดตัวเองอัตโนมัติ ภาพกระตุกไม่ลื่นไหล และ อื่นๆ อีกหลายประการ นิสิตคิดว่าสาเหตุของปัญหาเกิดจากอะไร

## 1.6 การโปรแกรมคอมพิวเตอร์เบื้องต้น

### 1.6.1 ภาษาโปรแกรม

การควบคุมคอมพิวเตอร์ในยุคแรกทำกับหน่วยประมวลผลกลางโดยตรง ยุคเริ่มต้นนั้น การป้อนคำสั่นทำโดยภาษาเครื่องเป็นรหัสเลขฐานสอง ซึ่งมีโอกาสผิดพลาดได้ง่าย รวมถึงการแก้ไขปรับปรุงโปรแกรมก็ทำได้ยาก จึงมีการใช้คำสั่งภาษาแอสเซมบลี (assembly) ซึ่งเป็นการแทนคำสั่งภาษาเครื่องต่างๆ ด้วยตัวอักษรภาษาอังกฤษที่สื่อความหมาย ทำให้การสั่งการง่ายขึ้น รูปที่ 1.31 แสดงคำสั่งภาษาเครื่องของเครื่อง MIPS32 ในรูปของเลขฐานสิบหกในการอ่านสีแดงทางซ้าย และคำสั่งภาษาแอสเซมบลีที่เทียบเท่ากันทางขวา ในการทำงานจริงนั้น เครื่องจำเป็นต้องแปลงคำสั่งแอสเซมบลีให้กลับไปเป็นเลขฐานสองเพื่อทำงานอีกครั้งอญญาติ โปรแกรมแปลงคำสั่งนี้เรียกว่า แอสเซมเบลอร์ (assembler)

ภาษาแอสเซมบลีนั้นไม่เหมาะสมสำหรับการเขียนซอฟต์แวร์ขนาดใหญ่ เนื่องจากมีคำสั่นน้อย จำเป็นต้องใช้หลายคำสั่นในการทำงานอย่างหนึ่ง จึงเกิดภาระตับสูงขึ้น โดยคำสั่งต่างๆ จะอยู่ในรูปแบบคล้ายคอลัมน์ภาษาอังกฤษ มีชุดคำสั่งต่างๆ เตรียมไว้ให้ เพื่อให้สะดวกในการพัฒนาและแก้ไขปรับปรุงโปรแกรม ชุดของคำสั่งที่เขียนเรียกว่า รหัสต้นฉบับ (source code) เมื่อเขียนเสร็จและต้องการนำไปใช้งาน ก็

ภาษาเครื่อง	ภาษาแอสเซมบลี
8fa40000	lw \$4, 0(\$29)
27a50004	addiu \$5, \$29, 4
24a60004	addiu \$6, \$5, 4
00041080	sll \$2, \$4, 2
00c23021	addu \$6, \$6, \$2
0c000000	jal 0x00000000 [main]
00000000	nop
3402000a	ori \$2, \$0, 10
0000000c	syscall

รูปที่ 1.31: ภาษาเครื่องและภาษาแอสเซมบลี

ต้องแปลงรหัสต้นฉบับเหล่านี้ไปเป็นภาษาเครื่องที่เครื่องรู้จัก การแปลงรหัสต้นฉบับอาศัยซอฟต์แวร์ระบบที่เรียกว่า ซอฟต์แวร์แปลภาษา ซึ่งก็นับเป็นซอฟต์แวร์ระบบด้วย

ภาษาโปรแกรมแบ่งออกเป็นสองระดับ ได้แก่

- ภาษาระดับต่ำ เป็นภาษาที่ติดต่อกับเครื่องได้โดยตรงหรือใกล้เคียง ได้แก่ ภาษาเครื่อง และภาษาแอสเซมบลี
- ภาษาระดับสูง เป็นภาษาที่ใช้โครงสร้างคำสั่งคล้ายภาษาอังกฤษ มีคำสั่งให้ใช้มาก สะดวกในการใช้เขียนโปรแกรมมากกว่าภาษาระดับต่ำ แต่ในการใช้งานนั้นจำเป็นต้องใช้คอมไพล์เตอร์หรืออินเทอร์เพเตอร์แปลคำสั่งเหล่านี้ให้กลายเป็นภาษาเครื่องก่อนจึงจะทำงานได้

## ภาษาโปรแกรมที่น่าสนใจ

### 1. ภาษาซีและภาษาซีพลัสพลัส (C และ C++)

ภาษาซีเป็นภาษาโปรแกรมระดับสูง พัฒนามาตั้งแต่ปี 1967 โดยเดนนิส ริชชี (Dennis Ritchie) มีลักษณะเป็นภาษาโปรแกรมเชิงโครงสร้าง ใช้กันทั่วไปตั้งแต่การเขียนซอฟต์แวร์ระบบไปจนถึงการเขียนโปรแกรมประยุกต์ คอมไพล์เตอร์สำหรับภาษาซีมีในแทบทุกสถานปัตยกรรมและระบบปฏิบัติการ ภาษาซีมีคำสั่งต่างๆ ที่ใช้ในการจัดการทรัพยากร่างกาย ของเครื่องโดยตรง จึงมักจะใช้ในการเขียนซอฟต์แวร์ระบบและซอฟต์แวร์ที่ต้องการปรับแต่งการใช้ทรัพยากร่างกาย ของเครื่องเอง

ภาษาซีพลัสพลัสพัฒนาต่อจากภาษาซี โดยออกแบบให้เป็นภาษาโปรแกรมเชิงวัตถุ เพื่อให้สะดวกต่อการพัฒนาซอฟต์แวร์มากขึ้น และยังคงมีประสิทธิภาพในการจัดการทรัพยากรของเครื่องได้สูง เช่นเดียวกับภาษาซี การใช้ภาษาซีพลัสพลัสพบมากในงานการประมวลผลภาพ และงานอื่นๆ ที่ต้องการประสิทธิภาพในการคำนวณสูง

### 2. ภาษาจาวา (Java)

ภาษาจาวาพัฒนาโดยซันไมโครซิสเต็มส์ (Sun Microsystems) ปัจจุบันอยู่ในความดูแลของออราคิล (Oracle) เป็นภาษาโปรแกรมเชิงวัตถุ ซึ่งหมายความว่าการออกแบบซอฟต์แวร์ขนาดใหญ่ ปรับปรุงแก้ไขเพิ่มเติมได้สะดวก จุดเด่นของภาษาจาวาคือเมื่อคอมไพล์แล้วจะได้ไบต์โค้ด (byte-code) แทนที่จะเป็นคำสั่งภาษาเครื่อง ในการทำงาน จะมีซอฟต์แวร์สำหรับทำงาน (runtime environment) ซึ่งจะแปลไบต์โค้ดเป็นภาษาเครื่องซึ่งจำเพาะกับแต่ละเครื่องอีกทีหนึ่ง จะเห็นว่าโปรแกรมซึ่งเขียนโดยภาษาจาวา เมื่อคอมไпал์แล้วจะได้ไบต์โค้ด ซึ่งสามารถนำไปใช้งานในเครื่องอื่นได้โดยไม่ต้องคอมไпал์ใหม่ โดยไม่จำกัดทั้งสถาปัตยกรรมและระบบปฏิบัติการ ถ้าหากมีซอฟต์แวร์สำหรับทำงานติดตั้งอยู่

### 3. ดีอตเน็ตเฟรมเวิร์ก (.NET)

ออกแบบโดยไมโครซอฟต์เพื่อการทำงานบนเครื่องหลักหลายระบบปฏิบัติการ ดีอตเน็ตเฟรมเวิร์กเป็นแพลตฟอร์มสำหรับพัฒนาซอฟต์แวร์ซึ่งสนับสนุนภาษาโปรแกรมจำนวนมาก ตัวอย่างภาษาดังต่อไปนี้ที่อุปกรณ์ที่สามารถทำงานบนเครื่องจักรได้ คือ VB.NET C# เป็นต้น ผู้เขียนซอฟต์แวร์สามารถเขียนรหัสต้นฉบับด้วยภาษาใดก็ได้ที่เฟรมเวิร์กสนับสนุน คอมไพล์จะแปลรหัสต้นฉบับเป็นภาษากลาง (Common Intermediate Language - CIL) แล้วจะมีซอฟต์แวร์สำหรับทำงานซึ่งแปลภาษาที่เขียนมาเป็นคำสั่งสำหรับเครื่องในการทำงานอีกครั้ง ดังนั้น ดีอตเน็ตเฟรมเวิร์กจะเหมือนกับภาษาในแท็บเล็ตที่สามารถคอมไпал์รหัสต้นฉบับแล้วนำไปใช้ที่อื่นได้โดยไม่ต้องคอมไпал์ซ้ำ แต่มีข้อดีกว่าที่ใช้ภาษาใดก็ได้ที่ผู้เขียนถนัดในการเขียน

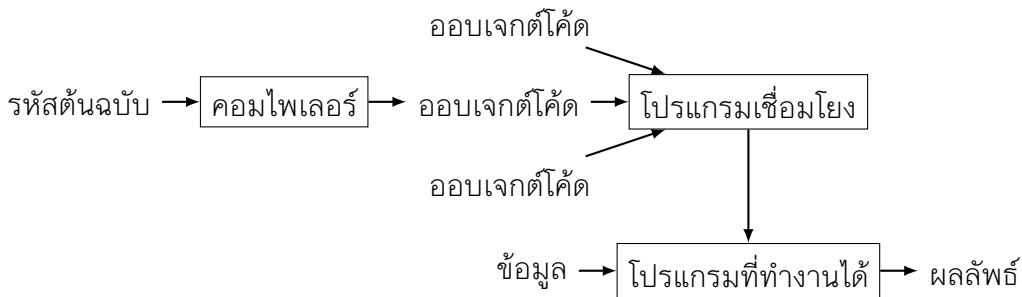
### 4. ภาษาไพธอน (Python) ภาษาไพธอนเป็นภาษาสคริปต์ ซึ่งการทำงานจะอาศัยอินเทอร์พรีเตอร์ในการแปล ใช้ได้ทั้งการเขียนโปรแกรมเชิงโครงสร้างและเชิงวัตถุ จุดเด่นของไพธอนคือมีชุดคำสั่งสำหรับงานต่างๆ หลากหลาย ใช้คำสั่งเพียงไม่กี่บรรทัดในการสั่งให้คอมพิวเตอร์ทำงานเมื่อเทียบกับคำสั่งที่ต้องใช้ในภาษาซีหรือจาวา เรียนรู้ได้ง่ายสำหรับผู้เริ่มต้น ปัจจุบันไพธอนเป็นที่นิยมในงานวิทยาศาสตร์ และงานเว็บไซต์

### 5. ภาษาอาาร์ (R) ภาษาอาาร์เป็นภาษาสคริปต์ เช่นเดียวกับไพธอน และอาศัยอินเทอร์พรีเตอร์ในการแปล เช่นเดียวกัน ภาษาอาาร์มักใช้ในงานสถิติ จุดเด่นของภาษาอาาร์คือเรียนรู้ได้ง่าย มีฟังก์ชันต่างๆ ด้านสถิติ และสร้างกราฟต่างๆ จากการคำนวณได้ง่ายและสวยงาม

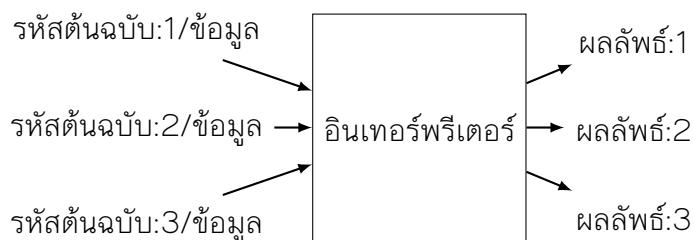
## 1.6.2 ซอฟต์แวร์แปลภาษา

การแปลรหัสต้นฉบับมีสองรูปแบบ แบบแรกคือแปลทั้งโปรแกรมให้เสร็จเป็นภาษาเครื่องที่พร้อมใช้งาน เราเรียกการแปลโปรแกรมแบบนี้ว่าการคอมไпал์ (compile) และเรียกซอฟต์แวร์การแปลรหัสต้นฉบับแบบนี้ว่า คอมไпал์เวอร์ (compiler) ตัวอย่างของภาษาโปรแกรมที่ใช้คอมไпал์เวอร์แปล เช่น ซี ปาล คัล บลีก อิกรูปแบบในการแปลรหัสต้นฉบับคือการแปลคำสั่งแล้วทำงานเลยทีละคำสั่ง ซอฟต์แวร์ที่ใช้ในการแปลภาษาแบบนี้เรียกว่า อินเทอร์พรีเตอร์ (interpreter) การแปลภาษาฐานรูปแบบนี้ใช้กับภาษาที่มีลักษณะเป็นสคริปต์ เช่น ไพธอน matlab R

ขั้นตอนการแปลรหัสต้นฉบับโดยคอมไпал์รูปได้ดังรูปที่ 1.32 โปรแกรมแปลภาษาจะแปลรหัสต้นฉบับทั้งหมดไปเป็นօբเจกต์โค้ด (object code) ซึ่งเป็นภาษาเครื่อง หากมีข้อผิดพลาดในรหัส



รูปที่ 1.32: ขั้นตอนการทำงานของคอมไพล์เตอร์



รูปที่ 1.33: ขั้นตอนการทำงานของอินเทอร์พรีเตอร์

ต้นฉบับ คอมไพล์เตอร์จะแสดงข้อผิดพลาดและให้ผู้เขียนโปรแกรมแก้ไขให้ถูกต้องก่อนจะนำกลับมาแปลงใหม่ เมื่อไม่มีข้อผิดพลาดแล้วคอมไпал์เตอร์จะแปลงรหัสต้นฉบับทั้งหมดจนได้เป็นออบเจกต์โค้ด จากนั้น โปรแกรมเชื่อมโยง (linker) จะเชื่อมโยงออบเจกต์โค้ดอื่นๆ ที่เกี่ยวข้องที่อาจแปลงไว้ล่วงหน้าแล้ว รวมกันเป็นโปรแกรมที่สามารถทำงานได้ (executable program) และจึงสามารถสั่งทำงาน (execute) ให้โปรแกรมรับข้อมูลเข้า (ถ้ามี) ประมวลผล และแสดงผลลัพธ์ (ถ้ามี) ต่อไป

การแปลงโดยอินเทอร์พรีเตอร์นั้นต่างกับการแปลงโดยคอมไ yapıl์เตอร์คือ อินเทอร์พรีเตอร์จะเป็นเสมือนล่ามแปลภาษาที่แปลทีละคำสั่งและให้คอมพิวเตอร์ทำงานทันที และทำไปเรื่อยๆ จนกว่าจะหมดรหัสต้นฉบับหรือพบข้อผิดพลาด หากมีข้อผิดพลาดที่คำสั่งใด อินเทอร์พรีเตอร์จะหยุดทำงาน ณ ตำแหน่งนั้นพร้อมกับแสดงข้อผิดพลาด หรือให้แก้ไขคำสั่ง ณ ขณะนั้นแล้วทำงานต่อ

### 1.6.3 เครื่องมือในการพัฒนาซอฟต์แวร์

ขั้นตอนการพัฒนาซอฟต์แวร์เริ่มต้นจากการออกแบบซอฟต์แวร์ก่อน ในการออกแบบจะกำหนดวัตถุหรือส่วนต่างๆ ที่จะทำงานประสานกันในซอฟต์แวร์ จากนั้นจะออกแบบการทำงานแต่ละอย่างและการทำงานประสานกันระหว่างส่วนต่างๆ ในซอฟต์แวร์นั้น ซึ่งอาจจะใช้งานรูปแบบต่างๆ ช่วยได้

หลังจากการออกแบบส่วนต่างๆ และจะเปลี่ยนการออกแบบให้เป็นรหัสต้นฉบับ ซึ่งมักจะมีรูปแบบของรหัสต้นฉบับที่สอดคล้องกับการออกแบบอยู่แล้ว การเขียนรหัสต้นฉบับอาจใช้ซอฟต์แวร์ที่เปิดได้ ที่สามารถพิมพ์ข้อความได้ แต่ในปัจจุบันมีเครื่องมือช่วยเหลือจำนวนมากที่ช่วยในการเขียนรหัสต้นฉบับ เช่น เครื่องมือช่วยในการพัฒนา (Integrated Development Environment - IDE) เช่น

Microsoft Visual Studio, Eclipse เป็นต้น นอกจาก IDE จะมีความสามารถในการพิมพ์ข้อความแล้ว ยังมีความสามารถในการแสดงความซึ่อมโยงของส่วนต่างๆ ในรหัสต้นฉบับ และยังมักจะรวมคอมไพล์อินเทอร์พรีเตอร์เข้าไปในตัว รวมถึงตัวช่วยแสดงคำแนะนำเกี่ยวกับคำสั่งต่างๆ และโปรแกรมในการตรวจสอบจุดบกพร่องอีกด้วย

