สรุปย่อเนื้อหา SU101

โดยนายชานนทร์ เหลืองประเสริฐ

Introduction

1. **คอมพิวเตอร์ทำงานอย่างไร**

คอมพิวเตอร์ประกอบไปด้วย CPU หรือ central processing unit หรือเรียกว่าเป็นหน่วยประมวผลกลางหรือตัวคิด (ไม่ว่าจะเป็นการคิดแบบคำนวณทางคณิตศาสตร์หรือคิดแบบตัดสินใจจากการใส่เงื่อนไขเข้าไปและการคิดนี้จะนำไปสู่การคิดที่ซับซ้อนขึ้น) นั่นเอง ซึ่งเมื่อ CPU ต้องการจะคุยกับส่วนประกอบอื่นก็จะทำผ่าน BUS ทั้งไปและกลับ

ส่วนประกอบต่อไปจะเรียกว่า RAM หรือ Random access memory (แปลว่า memory ที่เข้าถึงได้แบบ random หรือเท่าเทียมกัน) มีหน้าที่จำ โดย RAM เป็นช่องจำนวนมากสำหรับเก็บข้อมูลเช่น RAM 4 GB ก็จะมี 4 พันล้านช่อง ไม่ว่าจะ access เข้าในช่องไหนก็จะใช้เวลาเท่า ๆ กัน (ไม่เหมือน CD-ROM ที่ต้องหมุนก่อนเพื่อให้ access เข้าได้ตามช่องที่กำหนด) ซึ่ง RAM จะเชื่อมต่อกับ BUS เหมือนกัน

ส่วนอื่น ๆ ที่เหลือนอกจาก CPU กับ RAM จะถูกเรียกรวม ๆ กันว่า I/O ย่อมาจาก input output unit ถ้าพิจารณาจาก keyboard , mouse , หน้าจอที่มาต่อรวมกันเป็นคอมพิวเตอร์แล้ว keyboard กับ mouse ก็จะเป็น input คือนำเข้าข้อมูลจากมนุษย์ ส่วนหน้าจอ (หรือลำโพง เครื่อง printer แขนกล) จะเป็น output ส่งกลับออกไป

ข้อมูลจะไหลจาก keyboard เข้าไปใน RAM เพื่อจำหรือรับตัวเลขจาก keyboard เข้าไปใน CPU เพื่อคิดหาผลลัพธ์ก่อนและค่อยไหลเข้าไปใน RAM

1. **ทำไมต้องมีคอมพิวเตอร์**

การทำงานของสมองมนุษย์มีข้อจำกัดจากการจำ (เช่นการจำตัวเลข 3 5 7 9 5 4 5 6 8 5 3 2 ซึ่งบางคนก็จำได้แต่เมื่อเวลาผ่านไปนาน ๆ ก็จะเริ่มลืมแล้ว) และการคำนวณ (เช่นการหาผลลัพธ์ของ 235643 \* 8723999983ภายใน 1 วินาที) แต่คอมพิวเตอร์สามารถจำได้นานเป็นปีและสามารถคำนวณได้รวดเร็วมาก มนุษย์จึงต้องให้คอมพิวเตอร์ช่วยจำ (กรณีนี้โทรศัพท์มือถือก็เป็นคอมพิวเตอร์) และคิดแทนมนุษย์ (เช่นการเดิน การวิ่ง การยิงปืนในเกม)

1. **การเขียนโปรแกรมคืออะไร**

ในการที่คอมพิวเตอร์จะทำงานได้ต้องมีสิ่งที่เรียกว่า hardware กับ software

Hardware คือสิ่งที่เป็นตัวเครื่อง ส่วน software คือโปรแกรมคือสิ่งที่อยู่ข้างในคอมพิวเตอร์ ถ้าเทียบกับมนุษย์ แขน ขา มือ สมอง ของมนุษย์คือ hardware ส่วนการคิดของมนุษย์คือ software การคิดที่ต่างกันทำให้มนุษย์ต่างกัน เครื่องใช้ไฟฟ้าเช่นโทรศัพท์มือถือ,ตู้เย็น,microwave,remote ก็มี software แฝงอยู่

การเขียนโปรแกรมจึงเป็นการเอา hardware มาทำให้สามารถทำงานได้โดยการเขียนชุดคำสั่งลงไป

1. **ประวัติศาสตร์ของคอมพิวเตอร์**

ในช่วงสงครามโลกครั้งที่ 2 มนุษย์ต้องการอุปกรณ์คำนวณเพื่อใช้ในการทำสงคราม

คอมพิวเตอร์ในปัจจุบันถูกสร้างขึ้นตามรูปแบบของ Turing machine ซึ่ง Turing machine ถูกคิดขึ้นโดย Alan Mathison Turing (ผู้ให้กำเนิดคอมพิวเตอร์) นักคณิตศาสตร์ชาวอังกฤษ ผู้ที่สามารถแก้รหัส Enigma ซึ่งเป็นรหัสลับที่ฝ่ายเยอรมันใช้สื่อสารกันได้โดยวิวัฒนาการของฝั่ง hardware จะเป็น relay > หลอดสุญญากาศ > transiter > วงจรรวม > CPU

แต่สำหรับฝั่ง software นั้นจะเกิดขึ้นจากในช่วงสงครามโลกครั้งที่ 2 เวลาที่ต้องการยิงปืนใหญ่ก็จะต้องคำนวณว่าจะต้องตั้งปืนที่มุมเท่าใด ใส่ดินปืนเท่าใด ทิศทางลมเป็นอย่างไรให้ยิงลงไปที่ตำแหน่งของศัตรู สหรัฐอเมริกาจึงสร้างตัวคิดคำนวณขึ้นมาเป็นคอมพิวเตอร์ยุคแรกและคอมพิวเตอร์ยุคแรกนี้ต้องใช้กับบัตรเจาะรูที่เป็นรหัส 0 กับ 1 เช่นหากต้องการบวกเลขก็ต้องใช้ machine code (หรือ 0 กับ 1) โดยกำหนดว่าให้ 0011 เป็นการบวก , 0110 เป็นการลบ ซึ่งภาษา machine code เป็นภาษาที่เข้าใจได้ยาก มนุษย์จึงสร้างสิ่งที่เรียกว่าภาษา Assembly โดยเป็นการกำหนดคำว่า add ให้เป็น 0011100 ซึ่งมีความหมายให้บวก แต่ภาษา Assembly ก็ไม่สามารถแก้ไขปัญหาได้มากนักและไม่สามารถลดการเขียน code คำสั่งได้มากนัก จึงมีผู้พัฒนาต่อโดยเปลี่ยนจากคำว่า add ให้เป็นเครื่องหมาย + แล้วเปลี่ยนโครงสร้างเป็น add a , b store c หมายถึงบวกค่า a กับ b แล้วเอาไปเก็บใน c

แล้วต่อมาก็มีผู้พัฒนาภาษาขึ้นมาใหม่อีกเรียกว่าภาษา Basic แล้วกำหนดโครงสร้างภาษาให้เป็น c=a+b เพื่อให้ใกล้เคียงกับภาษามนุษย์มากขึ้น (หรือเรียกว่าภาษาระดับสูง) และต่อมาก็มีผู้พัฒนาภาษา Basic แล้วเปลี่ยนชื่อให้เป็นภาษา C ซึ่งชื่อ C มาจากการเรียงต่อมาจาก Assembly ที่มีสกุลไฟล์เป็น .A และ Basic ที่มีสกุลไฟล์เป็น .B

ภาษา C จะมีสิ่งที่เรียกว่า function ซึ่งทำให้มนุษย์สามารถลดการเขียน code ลงได้มากทำให้ภาษา C ได้รับความนิยมเป็นอย่างมากและเมื่อได้รับความนิยมอย่างมากทำให้มีผู้พัฒนาภาษา C ต่อมาอีกจนต่อมากลายเป็นภาษา C++ (เพิ่มเรื่อง OOP) และจาก C++ ก็มีผู้พัฒนาต่อไปเป็นภาษา JAVA แล้วต่อมา Microsoft ก็เอาภาษา JAVA มาปรับปรุงบางส่วนจนเกิดเป็นภาษา C# และเอาภาษา Basic มาปรับปรุงบางส่วนเช่นกันจนเกิดเป็นภาษา Visual Basic (ปัจจุบันก็ถูกพัฒนาต่อมาเป็น VB.NET)

โดยสรุปแล้วภาษาถูกพัฒนาขึ้นเพื่อแก้ไขปัญหาจากภาษาเก่า ๆ และเพื่อการตลาด

1. **การเขียนโปรแกรมต้องรู้อะไรบ้าง**

ต้องรู้ 4 อย่างคือ

1. คณิตศาสตร์ (ที่ไม่ซับซ้อน) ระดับมัธยมปลาย
2. ภาษาอังกฤษ
3. เวลาเรียน
4. ทำการบ้าน

สิ่งที่จะทำให้ผู้เรียนเขียนโปรแกรมเป็นจะต้องรู้

1. เรื่องภาษา ซึ่งผู้เรียนจะใช้พลังงาน 5%
2. Algorithm ซึ่งผู้เรียนจะใช้พลังงาน 30%
3. การศึกษา library (เพราะผู้เรียนจะไม่เขียนใหม่ตั้งแต่ต้นแต่ให้ใช้ต่อของที่คนอื่นคิดมาแล้ว NASA ก็ไม่เริ่มจากการไปนั่งใต้ต้น apple) ซึ่งผู้เรียนจะใช้พลังงาน 30% (library คือ code ที่คนอื่นคิดมาแล้ว) เพราะความรู้ในโลกนี้เกิดจากกระบวนการเรียนรู้ที่สะสมต่อเนื่องกันมา (เป็นภาษาอังกฤษ) ตัวอย่างเช่นทฤษฎีการเคลื่อนที่ของดาวต่าง ๆ ก็ถูกคิดต่อเนื่องกันมาโดย NICOLAUS COPERNICUS > JOHANNES KEPLER > GALILEO GALILEI > SIR ISAAC NEWTON การสร้างโปรแกรมก็เช่นกัน
4. IDEA ซึ่งผู้เรียนจะใช้พลังงาน 35%

หากอยากเขียนโปรแกรมเก่งก็ต้อง “ฝึกเยอะ ๆ” 10,000+ ชั่วโมง

BILL GATE ฝึกเขียนโปรแกรมตั้งแต่อายุ 16 ปี ทุกวัน วันละ 8-10 ชั่วโมงจนถึงอายุ 21 ปี รวม ๆ แล้วก็ประมาณ 4,000+ ชั่วโมง ซึ่งถ้านับจนถึงช่วงเวลาทำงานกับตั้งบริษัท Microsoft ด้วยก็มากกว่านั้นอีก

TIGER WOOD ตีกอล์ฟตั้งแต่อายุ 3 ปีจนถึง 16 ปี รวม ๆ แล้ว 10,000+ ชั่วโมงก่อนที่จะเริ่มเล่นกอล์ฟเป็นอาชีพ

1. **ภาษา**

การเขียนโปรแกรมคือการสั่งให้คอมพิวเตอร์ทำงานให้แทนมนุษย์ และมนุษย์ก็ต้องสอนคอมพิวเตอร์ก่อนโดยการใช้ภาษา ภาษาจะมี 2 อย่างคือ 1 คำศัพท์ 2 ไวยากรณ์ ตัวอย่าง ภาษา C , C++ , C# , JAVA , PYTHON , VB.NET , PHP แต่ละภาษาก็มีจุดเด่นต่างกันไปเช่น C , C++ ใช้ในโรงงานที่ต้องมี performance สูง ๆ ได้ดีแต่ใช้ยาก

คนหนึ่งคนพูดภาษาไทยได้ พูดภาษาอังกฤษได้ พูดภาษาอื่น ๆ ให้คนอื่นเข้าใจได้แต่คอมพิวเตอร์ไม่เข้าใจเพราะคอมพิวเตอร์รู้จักแต่ 0 กับ 1 วิธีสื่อสารกับคอมพิวเตอร์จะมีหลายวิธีเช่น ให้คนไปเรียน machine code คือท่อง 0 1 ว่าคืออะไรแต่การทำแบบนี้คนก็ไม่ไหว คนก็เลยคิดภาษากลางระหว่าง 0 1 กับภาษาคนขึ้นมาเช่น ภาษา JAVA ที่จะถูกนำไป compile (compile คือการแปลภาษาคอมพิวเตอร์ให้เป็น machine code 0 1 ทีเดียวทั้งหมด) เพื่อให้คอมพิวเตอร์เข้าใจโดยตัว compiler จะต่างกันออกไปในแต่ละภาษาเช่น JDK , .NET FRAMEWORK (\*\* ยกเว้น PYTHON ใช้เป็น interpreter แทนคือการ compile ทีละบรรทัด ทำให้หาข้อผิดพลาดง่ายแต่ช้า)

Main