บทที่ 1
ภาพรวมของคอมพิวเตอร์และการเขียนโปรแกรม
(Overview of Computers and Programming)

1.1 คอมพิวเตอร์ฮาร์ดแวร์เบื้องต้น
1.2 การแก้ปัญหาและการเขียนโปรแกรม
1.3 ภาพรวมของภาษาเขียนโปรแกรม
1.4 การประมวลผลโปรแกรมภาษาระดับสูง
1.5 การใช้สิ่งแวดล้อมแบบรวมของ Turbo Pascal

นับตั้งแต่ช่วงปี ค.ศ. 1940s เป็นต้นมา คอมพิวเตอร์เปลี่ยนแปลงอย่างรวดเร็วต่อวิธีที่เรามีชีวิตและเราทำธุรกิจกันอย่างไร ทุกวันนี้คอมพิวเตอร์เป็นสิ่งที่ใช้เพื่อประกอบการสอนในโรงเรียน พิมพ์ใบรายงานผลการเรียน (transcripts) เตรียมใบเสร็จรับเงินและเช็คจ่ายเงิน สำรองที่นั่งของสายการบิน ตั๋วงานคอนเสิร์ต ทำให้เราไปธนาคาร และซื้อสินค้าได้อย่างสะดวก ช่วยเหลือเราเขียนรายงาน และแม้กระทั่งเขียนหนังสือ ถึงแม้ว่าบ่อยครั้งที่เราเชื่อในคอมพิวเตอร์ กรณีอื่นๆ คอมพิวเตอร์ไม่สามารถให้เหตุผลได้เช่นที่เราทำโดยพื้นฐาน คอมพิวเตอร์หมายถึงอุปกรณ์สำหรับกระทำการคำนวณด้วยความเร็วแทบไม่น่าเชื่อ (กระทำการดำเนินการได้มากกว่าหนึ่งล้านครั้งต่อวินาที) และด้วยความถูกต้องสูงมาก อย่างไรก็ตาม เพื่อให้ประสบผลสำเร็จในสิ่งที่เป็นประโยชน์ คอมพิวเตอร์ต้องทำงานด้วย**โปรแกรม** ซึ่งหมายถึงรายการคำสั่งเพื่อให้กระทำ (A program is a list of instructions to perform.)

โปรแกรมปกติเขียนด้วยภาษาเขียนโปรแกรมคอมพิวเตอร์โดยเฉพาะ เช่น Pascal ในบทนี้จะแนะนำคอมพิวเตอร์และส่วนประกอบของคอมพิวเตอร์ จากนั้นนำเสนอภาพรวมของภาษาเขียนโปรแกรม สุดท้ายอธิบายว่าจะใช้ Turbo Pascal อย่างไร

คอมพิวเตอร์มีส่วนประกอบที่สำคัญสองอย่างคือ ฮาร์ดแวร์ และซอฟต์แวร์

ฮาร์ดแวร์ หมายถึง อุปกรณ์ที่ใช้กระทำการคำนวณที่จำเป็น ได้แก่ หน่วยประมวลผลกลาง (ซีพียู) จอมอนิเตอร์, คีย์บอร์ด และเครื่องพิมพ์ (Hardware is the equipment

used to perform the necessary computations and includes the central processing unit (CPU), monitor, keyboard, and printer.)

ซอฟต์แวร์ หมายถึง โปรแกรมซึ่งทำให้เราสามารถแก้ปัญหาด้วยคอมพิวเตอร์ โดยจัดทำรายการคำสั่งให้มันกระทำงาน (Software is the programs that enable us to solve problems with a computer by providing it with list of instructions to perform.)

การโปรแกรมด้วยคอมพิวเตอร์มีการเปลี่ยนแปลงอย่างมีนัยสำคัญผ่านเวลานับเป็นปีๆ เริ่มต้นจากเป็นงานที่ยากมาก ต้องให้โปรแกรมเมอร์เขียนคำสั่งโปรแกรมเป็นเลขฐานสองที่มีขนาดยาว (ลำดับของ 0s และ 1s) ภาษาโปรแกรมในทุกวันนี้ เช่น Pascal ทำให้การเขียนโปรแกรมง่ายมาก

ข้อสังเกต

ฮาร์ดแวร์ หมายถึง อุปกรณ์คอมพิวเตอร์จริง (Hardware is the actual computer equipment.)

ซอฟต์แวร์ หมายถึง เซตของโปรแกรมเกี่ยวข้องกับคอมพิวเตอร์ (Software is the set of programs associated with a computer.)

โปรแกรม หมายถึง รายการคำสั่งซึ่งทำให้คอมพิวเตอร์ใช้กระทำงานเฉพาะ (Program is a list of instructions that enable a computer to perform a specific task.)

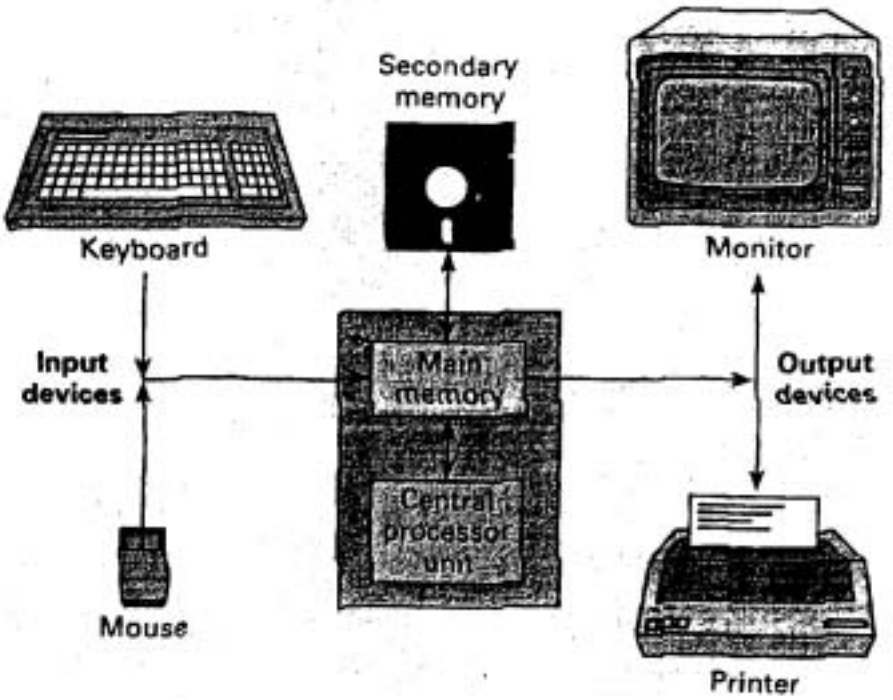
เลขฐานสอง หมายถึง เลขโดด ได้แก่ เลข 0 และ 1 (A binary number is a number whose digits are 0 and 1.)

1.1 คอมพิวเตอร์ฮาร์ดแวร์เบื้องต้น (Introduction to computer Hardware)

แม้จะมีความหลากหลาย ที่สำคัญในเรื่องราคา (cost) ขนาด (size) และความสามารถ (capabilities) คอมพิวเตอร์สมัยใหม่มีสิ่งที่คล้ายซึ่งกันและกันในพื้นฐานมากมาย โดยเฉพาะคอมพิวเตอร์ส่วนใหญ่มีส่วนประกอบดังต่อไปนี้

(1) หน่วยความจำหลัก
(2) หน่วยเก็บรอง ได้แก่ อุปกรณ์หน่วยเก็บ เช่น ฮาร์ดดิสก์ และฟล็อปปีดิสก์
(3) หน่วยประมวลผลกลาง
(4) อุปกรณ์อินพุต เช่น คีย์บอร์ด และเมาส์
(5) อุปกรณ์เอาต์พุต เช่น จอมอนิเตอร์ และเครื่องพิมพ์

รูป 1.1 แสดงให้เห็นส่วนประกอบเหล่านี้เชิงโต้ตอบกัน ในคอมพิวเตอร์ มีลูกศรชี้ทิศทางการไหลของสารสนเทศ ขั้นแรกโปรแกรมต้องถูกถ่ายโอนจากหน่วยความจำรองไปยังหน่วยความจำหลักก่อนการกระทำการ ปกติบุคคลซึ่งใช้โปรแกรม (program user) ต้องใส่ข้อมูลเพื่อให้ประมวลผล ข้อมูลเหล่านี้ส่งเข้าไปทางอุปกรณ์อินพุต (input device) และเก็บในหน่วยความจำหลักของคอมพิวเตอร์ ซึ่งมันจะถูกเข้าถึงและจัดดำเนินการโดยหน่วยประมวลผลกลาง จากนั้นผลลัพธ์ของการจัดดำเนินการจะเก็บกลับคืนในหน่วยความจำหลัก สุดท้ายสารสนเทศในหน่วยความจำหลักจะแสดงผลผ่านทางอุปกรณ์เอาต์พุต (output device)

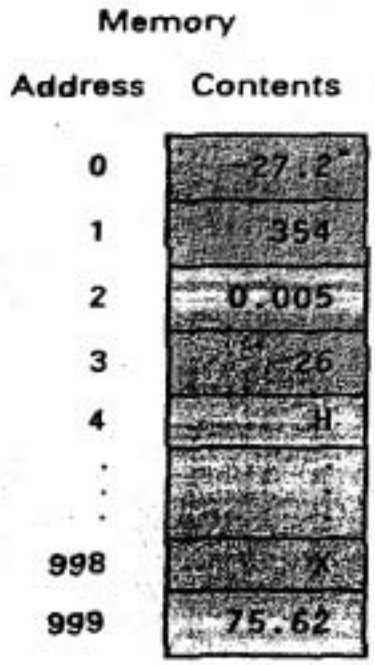


รูป 1.1 ส่วนประกอบของคอมพิวเตอร์

หน่วยความจำ (Memory)

หน่วยความจำ หมายถึง ส่วนประกอบที่สำคัญในคอมพิวเตอร์ทุกเครื่อง ก่อนที่จะอธิบายชนิดของหน่วยความจำ - หน่วยความจำหลัก และหน่วยความจำรอง เราจะดูว่าหน่วยความจำประกอบด้วยอะไร และคอมพิวเตอร์ทำงานกับหน่วยความจำอย่างไร

กายวิภาคของหน่วยความจำ (Anatomy of Memory)



รูป 1.2 เซลล์หน่วยความจำ 100 เซลล์ ในหน่วยความจำหลัก

เซลล์หน่วยความจำ หมายถึง ตำแหน่งหน่วยเก็บแต่ละตำแหน่งในหน่วยความจำ (Memory cell is an individual storage location in memory.)

เลขที่อยู่ของเซลล์หน่วยความจำ หมายถึง ตำแหน่งสัมพัทธ์ของเซลล์หน่วยความจำ ในหน่วยความจำหลักของคอมพิวเตอร์ (Address of a memory cell is the relative position of a memory cell in the computer's main memory.)

contents ของเซลล์หน่วยความจำ หมายถึง สารสนเทศซึ่งเก็บในเซลล์หน่วยความจำ อาจจะเป็นคำสั่งโปรแกรม หรือข้อมูล (Contents of a memory cell is the information stored in a memory cell, either a program instruction or data.)

แนวคิดการเก็บโปรแกรม หมายถึง การเก็บคำสั่งโปรแกรมในหน่วยความจำหลักก่อนการกระทำการ (Stored program concept is storing program instructions in main memory prior to execution.)

ไบต์ หมายถึง ปริมาณของหน่วยเก็บซึ่งใช้เก็บอักขระหนึ่งตัว (A byte is the amount of storage required to store a single character.)

บิต ย่อมาจาก **b**inary digit หมายถึงเลข 0 หรือ 1 (Bit is a binary digit; a 0 or a 1.)

หน่วยเก็บข้อมูล หมายถึง การตั้งค่าแต่ละบิตของเซลล์หน่วยความจำให้เป็น 0 หรือ 1 และทำลาย contents ก่อนหน้าของมัน (Data storage is setting the individual bits of a memory cell to 0 or 1, destroying its previous contents.)

การค้นคืนข้อมูล หมายถึง การสำเนา contents ของเซลล์หน่วยความจำเฉพาะไปยังพื้นที่หน่วยเก็บอีกที่หนึ่ง (Data retrieval is copying the contents of a particular memory cell to another storage area.)

หน่วยความจำหลัก (Main Memory)

หน่วยความจำหลักเก็บโปรแกรม ข้อมูล และผลลัพธ์ คอมพิวเตอร์ส่วนใหญ่มีหน่วยความจำหลักสองชนิด ได้แก่

(1) หน่วยความจำเข้าถึงโดยสุ่มหรือเรียกว่า แรม (random access memory (RAM)) ซึ่งเป็นหน่วยความจำเก็บโปรแกรมและข้อมูลชั่วคราว

(2) หน่วยความจำอ่านอย่างเดียวหรือเรียกว่า รอม (read-only memory (ROM)) ซึ่งเก็บโปรแกรมหรือข้อมูลถาวร

RAM เก็บโปรแกรมชั่วคราวขณะที่มันกำลังกระทำการโดยคอมพิวเตอร์ และ RAM ยังเก็บข้อมูลเช่น เลข ชื่อ รูปภาพ ขณะที่โปรแกรมกำลังจัดดำเนินการกับสิ่งเหล่านี้ RAM **เป็นหน่วยความจำลบเลือนได้** (volatile memory) หมายความว่า ทุกสิ่งใน RAM จะหายไปเมื่อเราปิดสวิทช์คอมพิวเตอร์

ในทางตรงกันข้าม ROM เก็บสารสนเทศอย่างถาวรภายในคอมพิวเตอร์ คอมพิวเตอร์สามารถค้นคืน (หรืออ่าน) แต่ไม่สามารถเก็บ (หรือเขียน) สารสนเทศใน ROM เหมือนกับชื่อของมัน คือ อ่านอย่างเดียว (read only)

เนื่องจาก ROM เป็น**หน่วยความจำชนิดไม่ลบเลือน** (nonvovatile) ข้อมูลซึ่งเก็บไว้ในหน่วยจำนี้ จึงไม่หายไป เมื่อเราปิดสวิทช์คอมพิวเตอร์

ROM เก็บคำสั่งที่จำเป็นเพื่อให้คอมพิวเตอร์ดำเนินงานเมื่อเราเปิดสวิทช์ครั้งแรก ความจุ (capacity) ของหน่วยความจำ RAM มีมากกว่าหน่วยความจำ ROM มาก และบ่อย

ครั้งที่เราสามารถเพิ่มหน่วยความจำ RAM ได้ (จนถึงความจุมากที่สุดซึ่งกำหนดไว้) ในขณะที่ปริมาณหน่วยความจำ ROM คงที่เสมอ เมื่อเราอ้างถึง หน่วยความจำหลักในหนังสือเล่มนี้ เราหมายถึง RAM เพราะว่าเป็นส่วนของหน่วยความจำหลักซึ่งปกติเข้าถึงได้โดยโปรแกรมเมอร์ Pascal

หน่วยความจำรองและอุปกรณ์หน่วยเก็บรอง (Secondary Memory and Secondary Storage Devices)

หน่วยความจำรอง จนถึงอุปกรณ์หน่วยเก็บรอง จัดความสามารถของการเก็บข้อมูล**กึ่งถาวร** (semipermanent data storage capbility)

หน่วยขับดิสก์ หมายถึง อุปกรณ์ร่วมหน่วยเก็บรอง ซึ่งใช้ในการเก็บและค้นคืนข้อมูลและโปรแกรมบนสื่อหน่วยเก็บที่เรียกว่า **ดิสก์** (Disk drive is a common secondary storage device which stores and retrieves data and program on a storage medium called disk.)

ดิสก์ (Disk) ถือว่าเป็นสื่อบันทึกชนิดกึ่งถาวร ไม่ใช่ถาวร เพราะว่า contents ของมันเปลี่ยนแปลงได้ คล้ายกับตลับเทปซึ่งบันทึกเพลงที่เล่นแล้วเล่นอีก และลบได้ บันทึกใหม่ทับได้

ดิสก์มีสองชนิด ได้แก่ ฮาร์ดดิสก์หรือ fixed disk และอีกชนิดหนึ่งคือ ฟล็อปปีดิสก์ คอมพิวเตอร์ส่วนใหญ่มีฮาร์ดดิสก์หนึ่งตัว ซึ่งไม่สามารถเอาออกจากหน่วยขับดิสก์ของมัน และมีพื้นที่หน่วยเก็บซึ่งใช้ร่วมกันของผู้ใช้ทั้งหมดของคอมพิวเตอร์ ปกติโปรแกรมซึ่งจำเป็นต้องใช้คำเนินงานของระบบคอมพิวเตอร์เก็บในฮาร์ดดิสก์ของมัน ผู้ใช้คอมพิวเตอร์แต่ละคนมีฟล็อปปีดิสก์หนึ่งแผ่นหรือมากกว่าหนึ่งแผ่น ซึ่งใส่เข้าไปในหน่วยขับฟล็อปปีดิสก์ของคอมพิวเตอร์

ฟล็อปปีดิสก์ เป็นแผ่นพลาสติกบางๆ ขนาดเส้นผ่าศูนย์กลาง 3.5 นิ้ว หรือ 5.25 นิ้ว เก็บในซองจัตุรัสบาง ฟล็อปปีดิสก์เหล่านี้เก็บโปรแกรมของผู้ใช้แต่ละคนและข้อมูลส่วนตัว

ฮาร์ดดิสก์เก็บข้อมูลได้มากกว่ามาก และสามารถถูกเข้าถึงโดย CPU ได้อย่างรวดเร็วมากกว่าแผ่นฟล็อปปีดิสก์

อย่างไรก็ตาม ผู้ใช้แต่ละคนมีแผ่นฟล็อปปีดิสก์ได้จำนวนไม่จำกัด ซึ่งไม่เหมือนฮาร์ดดิสก์

ฟล็อปปีดิสก์เคลื่อนย้ายได้ง่าย (portable) หมายความว่ามันสามารถใช้กับคอมพิวเตอร์ที่แตกต่างกันได้จำนวนมาก ตราบใดที่ทั้งหมดนั้นใช้แทนกันได้ (all compatible)

สารสนเทศซึ่งเก็บบนดิสก์ถูกจัดระเบียบให้เป็นกลุ่มที่แยกจากกัน เรียกว่า แฟ้ม (files) หนึ่งแฟ้มอาจเป็นโปรแกรม Pascal อีกหนึ่งแฟ้มอาจเป็นข้อมูลซึ่งจะถูกประมวลผลโดยโปรแกรมนั้น เรียกว่า แฟ้มข้อมูล (data file) แฟ้มที่สามอาจเป็นผลลัพธ์ที่สร้างโดยโปรแกรม เรียกว่า แฟ้มเอาต์พุต (output file)

แฟ้ม หมายถึง กลุ่มของสารสนเทศที่เกี่ยวข้องกันเก็บบนดิสก์ (File is a collection of related information stored on a disk.)

อุปกรณ์หน่วยเก็บรองร่วมที่เพิ่มขึ้น ได้แก่ หน่วยขับ CD-ROM หน่วยขับนี้เข้าถึงสารสนเทศซึ่งเก็บบนแผ่นดิสก์พลาสติกคล้าย CDs ที่ใช้ในเครื่องเล่น CD

หน่วยประมวลผลกลาง (Central Processor Unit)

หน่วยประมวลผลกลางหรือซีพียู (CPU) มีหน้าที่หลักสองอย่างคือ

(1) ประสานการดำเนินการทั้งหมดของคอมพิวเตอร์

(2) กระทำการดำเนินการคำนวณและตรรกะของข้อมูล

CPU ทำตามคำสั่งที่อยู่ในโปรแกรมคอมพิวเตอร์เพื่อกำหนดว่าการดำเนินการชุดใดควรจะกระทำและการเรียงอันดับเป็นอย่างไร จากนั้น CPU ส่งสัญญาณ (signal) การควบคุมการประสานงานไปยังส่วนประกอบคอมพิวเตอร์อื่นๆ ตัวอย่างเช่น ถ้าคำสั่งต้องการอ่านหน่วยข้อมูล CPU จะส่งสัญญาณควบคุมที่จำเป็นไปยังอุปกรณ์อินพุต

การประมวลผลโปรแกรมซึ่งเก็บในหน่วยความจำหลัก CPU ดันคืนแต่ละคำสั่งแบบลำดับ (เรียกว่าการรับเอา (fetching) คำสั่ง) จากหน่วยความจำหลักแปลความหมาย (interprets) ของคำสั่งว่ามันให้ทำอะไร จากนั้นดันคืนข้อมูลใดๆ ก็ตามที่จำเป็นต้องใช้กับคำสั่งนั้นต่อไป CPU กระทำการจัดดำเนินการจริง (actual manipulation) หรือการประมวลผลข้อมูลที่มันดันคืน CPU เก็บ ผลลัพธ์ในหน่วยความจำหลัก

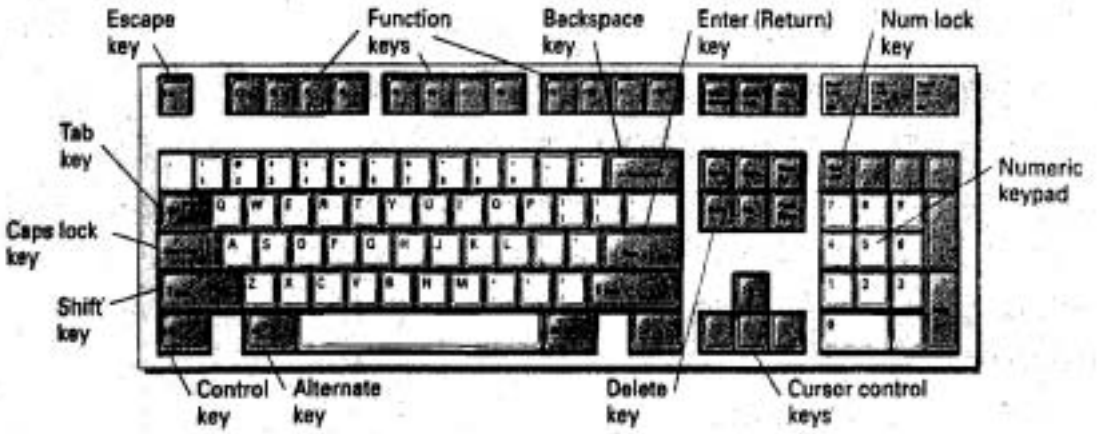
CPU กระทำการดำเนินการคำนวณ เช่น การบวก การลบ การคูณ และการหาร CPU ยังสามารถทำการเปรียบเทียบ contents ของเซลล์หน่วยความจำสองแห่งได้ด้วย (ตัวอย่างเช่น ชุดใดเก็บค่าที่ใหญ่กว่า, สองค่านั้นเท่ากันหรือไม่) และทำการตัดสินใจโดยขึ้นอยู่กับผลลัพธ์ของการเปรียบเทียบนั้น

การรับเอาคำสั่ง หมายถึง การดันคืนคำสั่งจากหน่วยความจำหลัก (Fetching an instruction is the retrieving an instruction from main memory.)

อุปกรณ์อินพุต/เอาต์พุต (Input/Output Devices)

เราใช้อุปกรณ์อินพุต/เอาต์พุต (I/O) เพื่อสื่อสารกับคอมพิวเตอร์โดยเฉพาะ การใส่ข้อมูลสำหรับการคำนวณและการสังเกตุผลลัพธ์ของการคำนวณนั้น

เราจะใช้คีย์บอร์ด (keyboard) ดูรูป 1.3 เป็นอุปกรณ์อินพุต และจอมอนิเตอร์เป็นอุปกรณ์เอาต์พุต เมื่อเรากดปุ่มตัวอักษรหรือปุ่มตัวเลขบนคีย์บอร์ด อักขระตัวนั้นจะถูกส่งไปยังหน่วยความจำหลัก และแสดงบนจอมอนิเตอร์ ณ ตำแหน่งของเคอร์เซอร์ (cursor) ซึ่งเป็นตัวทำเครื่องหมายการเคลื่อนที่ (ปกติใช้สัญลักษณ์ตัวขีดเส้นใต้กระพริบ) (Cursor is a moving place marker that appears on the monitor.)



รูป 1.3 คีย์บอร์ด

คีย์บอร์ดของคอมพิวเตอร์คล้ายคีย์บอร์ดของเครื่องพิมพ์ดีด ยกเว้นปุ่มพิเศษบางปุ่มสำหรับกระทำฟังก์ชันเฉพาะ ตัวอย่างเช่น บนคีย์บอร์ดของคอมพิวเตอร์ซึ่งแสดงในรูป 1.3 มีปุ่มพิเศษ 12 ปุ่ม ตอนแถวบนสุดระบุเป็นปุ่ม F1 ถึงปุ่ม F12 เรียกว่าฟังก์ชันคีย์ (function keys) กิจกรรมที่กระทำเมื่อเรากดฟังก์ชันคีย์ ขึ้นอยู่กับโปรแกรมปัจจุบันซึ่งกำลังกระทำการ นั่นคือ การกดปุ่ม F1 ในหนึ่งโปรแกรมปกติจะไม่ให้ผลลัพธ์เหมือนกับการกดปุ่ม F1 ในอีกหนึ่งโปรแกรม ปุ่มพิเศษอื่นๆ ทำให้เราสามารถลบทิ้งอักขระ เคลื่อนย้ายเคอร์เซอร์ และใส่ (enter) บรรทัดของข้อมูลที่เราพิมพ์ที่คีย์บอร์ด (ดูส่วนที่แรเงา ในรูป 1.3)

อุปกรณ์อินพุตร่วมอีกตัวหนึ่งคือ เมาส์ (mouse)

เมาส์ คืออุปกรณ์ขนาดมือถือ (hand-held device) ซึ่งใช้ในการเลือกการดำเนินการ (A mouse is an input device that moves its cursor on the computer screen to select an operation.)

เมื่อเราเคลื่อนที่เมาส์ไปรอบๆ บนโต๊ะทำงาน ลูกบอลยางซึ่งติดกับเมาส์จะหมุน และในเวลาเดียวกันเคลื่อนที่เคอร์เซอร์ (ปกติเป็นรูปสี่เหลี่ยมขนาดเล็ก หรือเป็นลูกศร) แสดงบนจอของมอนิเตอร์ เราเลือกการดำเนินการโดยเคลื่อนที่เมาส์ไปยังคำหรือรูปภาพ ซึ่งแทนการดำเนินการคอมพิวเตอร์ที่เราต้องการให้กระทำ และจากนั้นกดปุ่มบนเมาส์ หรือเรียกว่าคลิกปุ่มบนเมาส์ให้การดำเนินการที่เลือกไว้กระทำงาน

จอมอนิเตอร์จัดแสดงชั่วคราวของสารสนเทศซึ่งปรากฏบนจอภาพ เมื่อภาพไม่ปรากฏจากมอนิเตอร์ มันคือการสูญหายถ้าเราต้องการ hard copy (สำเนาถาวร/เวอร์ชันที่พิมพ์) ของสารสนเทศ เราต้องส่งสารสนเทศนั้นไปยังอุปกรณ์เอาต์พุต เรียกว่า เครื่องพิมพ์

เครื่องพิมพ์ หมายถึง อุปกรณ์เอาต์พุต ซึ่งให้สำเนาถาวรของสารสนเทศที่ส่งไปยังมัน (Printer is an output device that produces a hard copy of information sent to it.)

เครือข่ายคอมพิวเตอร์ (computer Network)

เราได้พูดเกี่ยวกับส่วนประกอบของคอมพิวเตอร์แต่ละตัวไปแล้ว บ่อยครั้งที่คอมพิวเตอร์หลายเครื่องมาเชื่อมโยงกัน (interconnected) เป็นเครือข่ายคอมพิวเตอร์ ดังนั้นผู้ใช้คอมพิวเตอร์แต่ละคนสามารถเข้าถึงฮาร์ดดิสก์ขนาดใหญ่ และเครื่องพิมพ์คุณภาพสูงได้ คอมพิวเตอร์ซึ่งควบคุมการเข้าถึงดิสก์ที่ใช้ร่วมกันเรียกว่า เครื่องบริการแฟ้ม (file server)

คอมพิวเตอร์แต่ละตัวในเครือข่ายจะมีคีย์บอร์ด จอมอนิเตอร์ และหน่วยขับดิสก์ของมันเอง ห้องปฏิบัติการคอมพิวเตอร์จำนวนมากจัดการฮาร์ดแวร์คอมพิวเตอร์ของเขาในเครือข่าย

เครือข่ายคอมพิวเตอร์ หมายถึง กลุ่มของคอมพิวเตอร์ซึ่งเชื่อมโยงกัน (Computer network is a group of interconnected computers.)

เครื่องบริการแฟ้ม หมายถึง คอมพิวเตอร์ในเครือข่ายซึ่งควบคุมการเข้าถึงดิสก์ใช้ร่วมกัน (File server is the computer in a network that controls access to the shared disk.)

แบบฝึกหัด 1.1 self-shcek

1. ถ้าคอมพิวเตอร์ถูกสั่งให้บวก content ของเซลล์หน่วยความจำ 2 และ 999 ในรูป 1.2 และเก็บผลลัพธ์ในเซลล์ 0 จงบอก content ของเซลล์ 0, 2 และ 999

2. หนึ่งบิตมีสองค่า ได้แก่ 0 หรือ 1 การจัดหมู่ (combination) ของสองบิตมี 4 ค่า ได้แก่ 00, 01, 10 และ 11 จงแสดงรายการของค่าต่างๆ ซึ่งเกิดจากการจัดหมู่ของ 3 บิต และทำเช่นเดียวกันกับ 4 บิต

3. จงแสดงรายการเรียงอันดับจากค่าที่เล็กที่สุด ไปยังค่าใหญ่ที่สุดของสิ่งต่อไปนี้ : byte, bit, main memory, memory cell, secondary memory

1.2 การแก้ปัญหาและการเขียนโปรแกรม (Problem Solving and Programming)

เนื่องจากคอมพิวเตอร์คิดเองไม่ได้ คอมพิวเตอร์ต้องใช้โปรแกรมเพื่อทำงานที่เป็นประโยชน์ใดๆ ก็ตาม โปรแกรมคอมพิวเตอร์เกี่ยวข้องมากกว่าการเขียนรายการคำสั่งอย่างง่าย การแก้ปัญหา คือ ส่วนประกอบที่สำคัญของการเขียนโปรแกรมและต้องการการเตรียมแผนงานล่วงหน้าที่ดีก่อน การเขียนโปรแกรมเพื่อแก้ปัญหาเฉพาะ เราต้องพิจารณาอย่างรอบคอบทุกด้านของปัญหา จากนั้นจึงพัฒนาและจัดการผลเฉลยของมัน

เหมือนกับนักศึกษาเขียนโปรแกรมส่วนมาก คือเราอาจจะใช้เวลาส่วนใหญ่ใส่โปรแกรมไปยังคอมพิวเตอร์ ต่อมาภายหลังเราจะใช้เวลาที่มากกว่าเพื่อลบทิ้งข้อผิดพลาดซึ่งจะหนีไม่พ้นที่มีอยู่ในโปรแกรม ดังนั้นให้มีความด้านทานความต้องการที่จะเร่งรีบ เริ่มต้นเขียนโปรแกรมทันทีที่เรามีความคิดว่าจะเขียนโปรแกรมอย่างไร แต่ให้คิดอย่างรอบคอบเกี่ยวกับปัญหาและผลเฉลยของมันก่อนการเขียนคำสั่งโปรแกรมใดๆ

เมื่อเรามีผลเฉลยที่มีศักยภาพอยู่ในใจแล้ว วางแผนก่อนการเขียนโดยใช้กระดาษและดินสอหรือโปรแกรมประมวลผลคำ และดัดแปรผลเฉลย ถ้ามีความจำเป็นก่อนเขียนโปรแกรมด้วยภาษาเขียนโปรแกรม

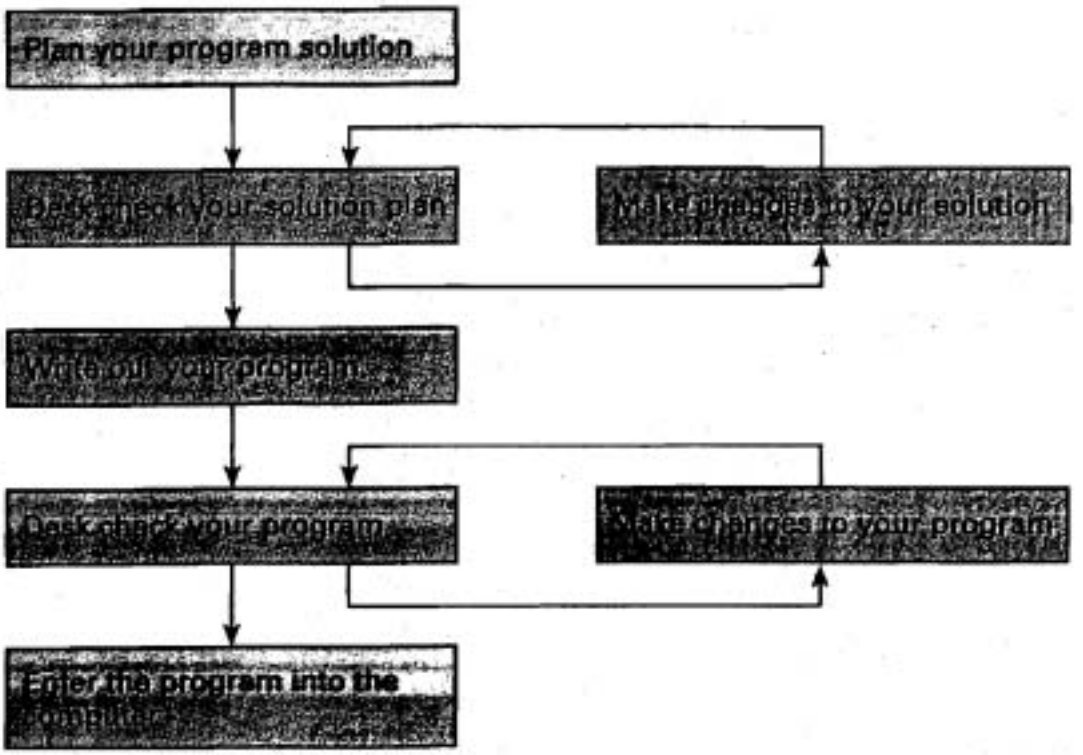
เราควรตรวจสอบแผนการแก้ปัญหาบนโต๊ะทำงาน โดยกระทำอย่างรอบคอบแต่ละขั้นตอนมากเท่ากับคอมพิวเตอร์ทำ การตรวจสอบบนโต๊ะ การวางแผนแก้ปัญหา การจำลองแบบผลลัพธ์ของแต่ละขั้นตอน โดยใช้ข้อมูลอย่างถูกต้อง ซึ่งง่ายต่อการจัดดำเนินการ (ตัวอย่างเช่น จำนวนเลขขนาดเล็ก)

จากนั้น เปรียบเทียบผลลัพธ์ของเรากับผลลัพธ์ที่คาดคิดไว้แล้ว และทำการแก้ไขที่ผิดพลาดที่จำเป็นใดๆ ให้กับแผนของการแก้ปัญหา

ณ จุดนี้ เราสามารถเขียนแผนการแก้ปัญหาของเราให้เป็นโปรแกรม เราควรตรวจสอบโปรแกรมของเราก่อนใส่โปรแกรมไปในคอมพิวเตอร์ การใช้เวลาเพียงไม่กี่นาทีที่เพิ่มขึ้นเพื่อการประเมินผลแผนการแก้ปัญหาและโปรแกรม โดยใช้กระบวนการ (process) ที่สรุปในรูป 1.4 บ่อยครั้งทำให้ประหยัดเวลานับชั่วโมงที่จะเกิดขึ้นภายหลัง

ในตำราเล่มนี้ เราเน้นระเบียบวิธี (methodology) ของการแก้ปัญหา ซึ่งได้พิสูจน์แล้วว่าเป็นประโยชน์ ช่วยนักศึกษาในการเรียนเขียนโปรแกรม เทคนิคนี้จะอธิบายและแสดงให้เห็นว่าจะประยุกต์ใช้อย่างไรในหัวข้อ 2.1

การตรวจสอบบนโต๊ะ หมายถึง การทำทีละขั้นตอน จำลองแบบการวางแผนแก้ปัญหา (Desk check is doing a step-by-step simulation of a solution plan.)



รูป 1.4 ยุทธวิธีของการเขียนโปรแกรม

แบบฝึกหัด 1.2 self-check

1. ถ้าคอมพิวเตอร์สามารถกระทำการโปรแกรมได้อย่างรวดเร็วและมีประสิทธิภาพ อะไรคือวัตถุประสงค์ของการตรวจสอบบนโต๊ะของโปรแกรม

1.3 ภาพรวมของภาษาเขียนโปรแกรม (Overview of Programming Languages)

เราใช้ภาษาเขียนโปรแกรมเพื่อเขียนโปรแกรมคอมพิวเตอร์ ภาษาเขียนโปรแกรมมีจำนวนมากที่แตกต่างกัน แต่ภาษาร่วมส่วนมากที่ใช้กันทุกวันนี้เรียกว่า ภาษาระดับสูง (high-level language) เพื่อให้มีความเข้าใจข้อดีของภาษาระดับสูง ขั้นแรกเราต้องมีความเข้าใจว่าคอมพิวเตอร์สื่อสารอย่างไร

ภาษาเครื่อง (Machine Language)

ภาษาโดยกำเนิดของคอมพิวเตอร์คือภาษาเครื่อง คำสั่งภาษาเครื่องทุกคำสั่งเป็นสายบิตของเลข 0s และ 1s ซึ่งหมายถึงการดำเนินการและระบุเซลล์หน่วยความจำที่เกี่ยวข้องกับการดำเนินการนั้น

ตัวอย่างเช่น ถ้าเราต้องการแทนสูตรพีชคณิต

cost = price + tax

เป็นโปรแกรมภาษาเครื่อง เราต้องมีลำดับของคำสั่ง เช่น

```
0010  0000  0000  0100
0100  0000  0000  0101
0011  0000  0000  0110
```

ในคำสั่งภาษาเครื่องแต่ละคำสั่ง การดำเนินการที่จะให้กระทำและเลขที่อยู่ของข้อมูลที่จะถูกจัดดำเนินการเขียนเป็นเลขฐานสอง ถึงแม้ว่าคอมพิวเตอร์จะเข้าใจคำสั่งภาษาเครื่องสามคำสั่งข้างต้นได้ไม่ยาก แต่สำหรับผู้คนส่วนใหญ่ไม่สามารถเข้าใจได้

ภาษาเครื่อง หมายถึง ภาษาดั้งเดิมของคอมพิวเตอร์ที่มีคำสั่งเป็นเลขฐานสอง (Machine Language is a computer's native language with instructions that are binary numbers.)

ภาษาระดับสูง (High-Level Languages)

เมื่อเราเขียนโปรแกรมในภาษาระดับสูง เราใช้คำสั่งซึ่งคล้ายกับภาษาที่ใช้ในทุกวันนี้ ใน Pascal เราใช้ข้อความสั่ง

Cost := Price + Tax

ซึ่งคล้ายกับสูตรดังเดิม ข้อความสั่งนี้ หมายความว่า "บวกค่าของ Price กับค่าของ Tax และเก็บผลลัพธ์ใน Cost" เมื่อเขียนโปรแกรมในภาษาระดับสูง เราสามารถอ้างถึงข้อมูลซึ่งเก็บในหน่วยความจำโดยใช้ชื่อเชิงพรรณนา (descriptive names) ตัวอย่างเช่น Price, Cost, Tax ไม่ใช่เลขที่อยู่เซลล์หน่วยความจำเชิงตัวเลข เราสามารถใช้สัญลักษณ์ที่คุ้นเคย (เช่น เครื่องหมาย +) เพื่ออธิบายการดำเนินการที่เราต้องการให้กระทำ

ภาษาระดับสูง หมายถึง ภาษาโปรแกรมซึ่งคำสั่งของมันคล้ายกับภาษาที่เราใช้ทุกวันนี้ (High-level language is a programming language whose instructions resemble everyday language.)

ภาษาระดับสูง มีมาตรฐานภาษาซึ่งอธิบายรูปแบบไวยากรณ์ (วากยสัมพันธ์) ของภาษา คำสั่งภาษาระดับสูงทุกคำสั่งต้องตรงกัน (conform) กับกฎวากยสัมพันธ์ที่กำหนดในมาตรฐานภาษากฎเหล่านี้จะเที่ยงตรงมาก (very precise) หมายถึง มันจะไม่ยอมให้แม้แต่คำสั่งซึ่งเกือบจะถูกต้อง (almost correct)

โปรแกรมซึ่งตรงกันกับกฎเหล่านี้จะเคลื่อนย้ายง่าย (portable) ซึ่งหมายความว่ามันสามารถใช้กับคอมพิวเตอร์หลายชนิด (หลายระบบ) โดยไม่ต้องมีการดัดแปรโปรแกรมแต่อย่างใด

ในทางตรงกันข้าม โปรแกรมภาษาเครื่องใช้ได้เฉพาะบนคอมพิวเตอร์เพียงหนึ่งระบบเท่านั้น

มาตรฐานภาษา หมายถึง คำอธิบายวากยสัมพันธ์และความหมายของคำสั่งภาษาระดับสูงแต่ละคำสั่ง (Language standard is a description of the syntax and meaning of each high level language instruction.)

ภาษาระดับสูงร่วม ได้แก่ Pascal, Fortran, Basic, COBOL, C และ C++ ถึงแม้ว่าภาษาเหล่านี้แต่ละภาษาถูกออกแบบสำหรับวัตถุประสงค์ใช้งานเฉพาะด้าน (ดูตาราง 1.1)

ภาษาทั้งหมดนี้ใช้เขียนซอฟต์แวร์ประยุกต์หลากหลาย - ซอฟต์แวร์ซึ่งกระทำงานสำหรับผู้ใช้คอมพิวเตอร์ ภาษา C และ C++ บ่อยครั้งใช้เขียนซอฟต์แวร์ระบบ - ซอฟต์แวร์ซึ่งกระทำงานที่ต้องใช้กับการดำเนินการของระบบคอมพิวเตอร์

ซอฟต์แวร์ประยุกต์ หมายถึง โปรแกรมซึ่งเขียนสำหรับผู้ใช้คอมพิวเตอร์ (Application software is programs written for a computer user.)

ซอฟต์แวร์ระบบ หมายถึง โปรแกรมซึ่งเขียนสำหรับระบบคอมพิวเตอร์ (System software is programs written for the computer system.)

ตาราง 1.1 ภาษาระดับสูงร่วม (Common High-Level Languages)

ภาษาระดับสูง	วัตถุประสงค์ดั้งเดิม
Pascal	สำหรับสอนนักศึกษาเขียนโปรแกรมอย่างรอบคอบ ในวิธีที่มีวินัย
Fortran (Formula Translation)	สำหรับงานประยุกต์ทางด้านวิศวกรรมศาสตร์และวิทยาศาสตร์
BASIC	ภาษาอย่างง่ายตั้งใจให้นักศึกษาใช้ทำงานในโรงเรียน (schoolwork)
COBOL	สำหรับกระทำการประมวลผลข้อมูลทางธุรกิจ
C	สำหรับเขียนซอฟต์แวร์ระบบ
C++	ส่วนขยายของ C ซึ่งสนับสนุนการเขียนโปรแกรมเชิงอ็อบเจกต์

ความสัมพันธ์ระหว่างภาษาระดับสูงกับภาษาเครื่อง (Relationship Between High-Level and Machine Language)

เนื่องจากคอมพิวเตอร์สามารถเข้าใจได้เฉพาะโปรแกรมซึ่งเขียนในภาษาเครื่องเท่านั้น แต่ละคำสั่งในโปรแกรมภาษาระดับสูง ขั้นแรกต้องแปลให้เป็นภาษาเครื่องก่อนที่มันจะถูกกระทำการ โปรแกรมภาษาระดับสูงดังเดิม เรียกว่า โปรแกรมต้นฉบับ เมื่อแปลเป็นภาษาเครื่องแล้วเรียกว่า โปรแกรมจุดหมาย หัวข้อถัดไปจะอภิปรายขั้นตอนต่างๆ ซึ่งต้องใช้ประมวลผลโปรแกรมภาษาระดับสูง

โปรแกรมต้นฉบับ หมายถึง โปรแกรมภาษาระดับสูงซึ่งกำลังจะถูกแปล (Source program is the high-level language program being translated.)

โปรแกรมจุดหมาย หมายถึง ส่วนที่แปลเป็นภาษาเครื่องแล้วของโปรแกรมต้นฉบับ (Object program is the machine-language translation of a source program.)

แบบฝึกหัด 1.3 Self-Check

1. ข้อความสั่งภาษาระดับสูง สี่คำสั่งต่อไปนี้ มีความหมายว่าอะไร

   X := A + B + C;

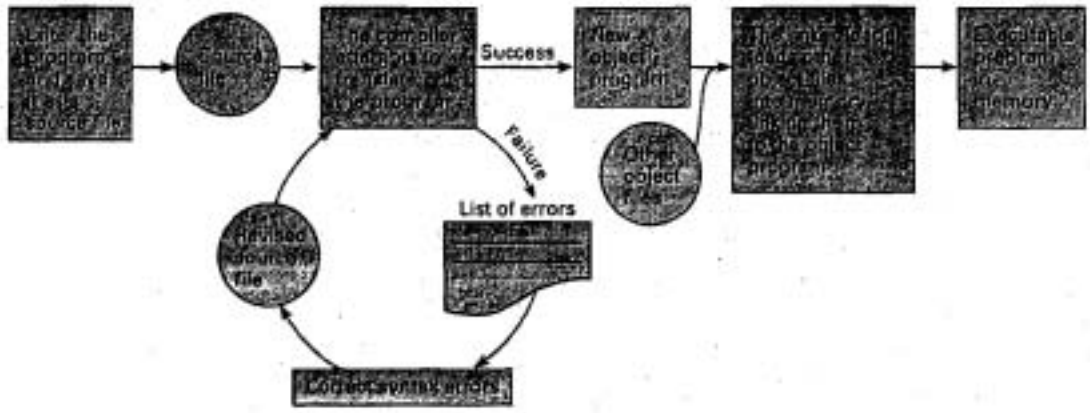
   X := Y/Z;

```
D := C - B + A;
Z := Z + 1
```

2. จงบอกเหตุผลสองข้อว่าทำไมนักศึกษาจึงชอบเขียนโปรแกรมในภาษา Pascal มากกว่าเขียนโปรแกรมในภาษาเครื่อง

1.4 การประมวลผลโปรแกรมภาษาระดับสูง (Processing a High-Level Language Program)

ก่อนที่คอมพิวเตอร์จะสามารถกระทำการโปรแกรมภาษาระดับสูงได้ โปรแกรมเมอร์ต้องใส่โปรแกรมต้นฉบับเข้าไปในคอมพิวเตอร์ และคอมพิวเตอร์ต้องเก็บโปรแกรมในรูปแบบกระทำการได้ในหน่วยความจำ โปรแกรมระบบหลายโปรแกรมช่วยเหลือการทำงานนี้ เราจะอธิบายความสำคัญของโปรแกรมเหล่านี้ และทำข้อสรุปกระบวนการในรูป 1.5



รูป 1.5 การเตรียมโปรแกรมสำหรับการกระทำการ

ขั้นตอนการเตรียมโปรแกรมสำหรับการกระทำการ (Steps for Preparing a Program for Execution)

1. ใช้โปรแกรมบรรณาธิกร เพื่อใส่แต่ละบรรทัดของโปรแกรมต้นฉบับไว้ในหน่วยความจำ และเก็บโปรแกรมบนดิสก์ให้เป็นแฟ้มต้นฉบับ

2. ใช้โปรแกรมคอมไพเลอร์ (ตัวแปลโปรแกรม) เพื่อแปลโปรแกรมต้นฉบับให้เป็นภาษาเครื่อง ถ้ามีข้อผิดพลาดวากยสัมพันธ์ใด ๆ (ข้อผิดพลาดในไวยากรณ์) คอมไพเลอร์

จะแสดงข้อผิดพลาดเหล่านี้บนจอมอนิเตอร์ ให้ใช้โปรแกรมบรรณาธิกร แก้ไขข้อผิดพลาดโดยการปรับแต่ง (editing) และเก็บใหม่ (resaving) โปรแกรมต้นฉบับ

3. เมื่อโปรแกรมต้นฉบับไม่มีข้อผิดพลาด คอมไพเลอร์เก็บ (save) ส่วนที่แปลเป็นภาษาเครื่องแล้วเป็นโปรแกรมจุดหมาย

4. โปรแกรมเชื่อมโยง/โปรแกรมบรรจุ รวม (combine) โปรแกรมจุดหมายของเรากับแฟ้มจุดหมายเพิ่มเติม ซึ่งอาจจำเป็นสำหรับการกระทำการโปรแกรมของเรา (ตัวอย่าง เช่น โปรแกรมสำหรับอินพุต และเอาต์พุต) และเก็บโปรแกรมภาษาเครื่องสุดท้ายในหน่วยความจำ พร้อมสำหรับการกระทำการ ตัวเชื่อมโยง/ตัวบรรจุจะ save โปรแกรมภาษาเครื่องสุดท้ายเป็นแฟ้มกระทำการได้บนดิสก์ด้วย บ่อยครั้งที่โปรแกรมเชื่อมโยง/โปรแกรมบรรจุเขียนเป็นโปรแกรมระบบแยกจากกันเป็นสองชุด

โปรแกรมบรรณาธิกร หมายถึง โปรแกรมซึ่งใช้ใส่โปรแกรมต้นฉบับและเก็บเป็นแฟ้มต้นฉบับ (Editor is a program used to enter source programs and save source files.)

คอมไพเลอร์ หมายถึง โปรแกรมซึ่งแปลโปรแกรมภาษาระดับสูงให้เป็นภาษาเครื่อง (Compiler is a program that translates a high-level language program into machine language.)

ข้อผิดพลาดวากยสัมพันธ์ หมายถึง ข้อผิดพลาดในรูปแบบเชิงไวยากรณ์ของบรรทัดในโปรแกรมภาษาระดับสูง (Syntax error is an error in the grammatical form of a line in a high-level language program.)

โปรแกรมเชื่อมโยง/โปรแกรมบรรจุ หมายถึง โปรแกรมซึ่งรวม (โยง) โปรแกรมจุดหมายกับแฟ้มจุดหมายอื่นๆ และเก็บ (บรรจุ) โปรแกรมภาษาเครื่องสุดท้ายในหน่วยความจำ (Linker/Loader is a program that combines (links) an object program with other object files and stores (loads) the final machine language program in memory.)

แฟ้มกระทำการได้ หมายถึง แฟ้มดิสก์ซึ่งประกอบด้วยโปรแกรมภาษาเครื่องที่พร้อมกระทำการ (Executable file is a disk file containing a ready-to-execute machine language program.)

การกระทำการโปรแกรม (Executing a Program)

การกระทำการโปรแกรมภาษาเครื่อง, CPU ต้องตรวจสอบคำสั่งโปรแกรมแต่ละคำสั่งในหน่วยความจำ และส่งออกสัญญาณคำสั่งงานที่ต้องใช้ปฏิบัติการคำสั่งให้บรรลุผล

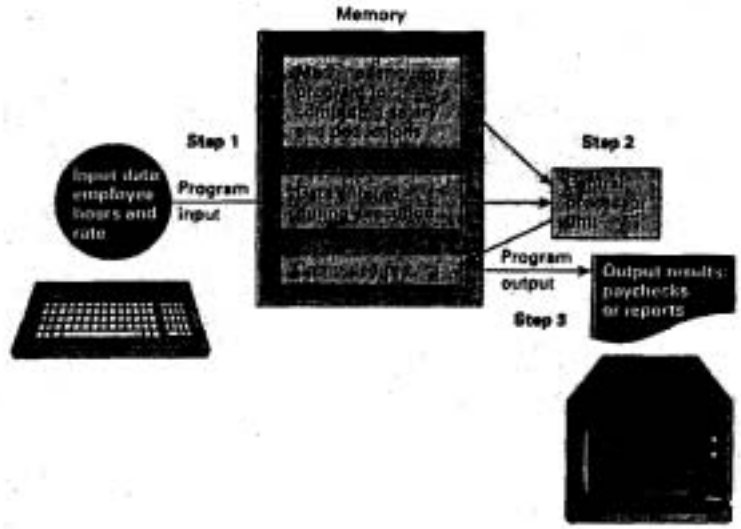
สำเร็จ ถึงแม้ว่าโดยปกติคำสั่งถูกกระทำการแบบลำดับ ซึ่งเราจะอภิปรายภายหลัง เป็นไปได้ที่ CPU จะข้ามบางคำสั่งหรือกระทำการบางคำสั่งมากกว่าหนึ่งครั้ง

ระหว่างการกระทำการ ข้อมูลสามารถส่งไปยังหน่วยความจำและถูกจัดดำเนินการในวิธีซึ่งกำหนดไว้ คำสั่งโปรแกรมพิเศษนำมาใช้สำหรับการใส่หรือการอ่านข้อมูลของโปรแกรม (เรียกว่าข้อมูลอินพุต) ไว้ในหน่วยความจำ หลังจากข้อมูลอินพุตประมวลผลเสร็จแล้ว คำสั่งสำหรับแสดงผลหรือการพิมพ์ค่าต่างๆ ในหน่วยความจำถูกกระทำการเพื่อแสดงผลลัพธ์โปรแกรมบรรทัดซึ่งแสดงผลโดยโปรแกรมเรียกว่าเอาต์พุตโปรแกรม

ข้อมูลอินพุต หมายถึง ค่าข้อมูลซึ่งถูกอ่านโดยโปรแกรม (Input data are the data values that are read by a program.)

เอาต์พุตโปรแกรม หมายถึง บรรทัดซึ่งแสดงผลโดยโปรแกรม (Program output are the lines displayed by a program.)

ให้ใช้สถานการณ์ ซึ่งอธิบายในรูป 1.6 - การกระทำการโปรแกรมบัญชีเงินเดือน ซึ่งเก็บในหน่วยความจำเป็นตัวอย่าง ขั้นที่หนึ่งของโปรแกรมใส่ข้อมูลซึ่งอธิบายจำนวนชั่วโมงทำงานของพนักงานและอัตราค่าจ้างไว้ในหน่วยความจำ ขั้นที่สองโปรแกรมจัดดำเนินการข้อมูลพนักงานและเก็บผลลัพธ์ของการคำนวณในหน่วยความจำ ขั้นสุดท้ายผลลัพธ์จากการคำนวณ แสดงผลเป็นรายงานบัญชีเงินเดือนหรือเช็คบัญชีเงินเดือนของพนักงาน



รูป 1.6 การไหลของสารสนเทศระหว่างกระทำการโปรแกรม

แบบฝึกหัด 1.4 Self-Check

1. ข้อผิดพลาดวากยสัมพันธ์จะถูกพบในโปรแกรมต้นฉบับหรือพบในโปรแกรมจุดหมาย โปรแกรมระบบชื่ออะไรซึ่งจะพบข้อผิดพลาดวากยสัมพันธ์ถ้ามันมีอยู่ในโปรแกรมที่เราเขียน และโปรแกรมระบบชื่ออะไรซึ่งเราใช้แก้ไขข้อผิดพลาดนั้นให้ถูกต้อง

2. จงอธิบายความแตกต่างระหว่างโปรแกรมต้นฉบับ, โปรแกรมจุดหมาย และโปรแกรมกระทำการได้ ชุดใดเราเป็นคนสร้าง ชุดใดคอมไพเลอร์สร้าง และชุดใดโปรแกรมเชื่อมโยง/โปรแกรมบรรจุสร้าง

1.5 การใช้สิ่งแวดล้อมแบบรวมของ Turbo Pascal (Using the Turbo Pascal Integrated Environment)

กลไกของการใส่โปรแกรมให้เป็นแฟ้มต้นฉบับ การแปลแฟ้มต้นฉบับให้เป็นภาษาเครื่อง และการกระทำการโปรแกรมภาษาเครื่องแตกต่างกันบนระบบคอมพิวเตอร์แต่ละระบบ ในตำราเล่มนี้เราใช้ IBM-PC (คอมพิวเตอร์ส่วนบุคคล) หรือคอมพิวเตอร์ใช้แทนกันได้ compatible computer) และ Turbo Pascal เวอร์ชัน 7.0 เวอร์ชันของ Turbu Pascal ก่อนหน้านี้อาจกระทำแตกต่างกัน Turbo Pascal พัฒนาโดยบริษัท Borland International สำหรับ IBM PCs

Turbo Pascal จัดหาสิ่งแวดล้อมแบบรวม ซึ่งหมายความว่า เราสามารถสร้าง (create), ตรวจแก้ (edit), คอมไพล์, เชื่อมโยง และบรรจุ โปรแกรม Pascal ของเราจากใน Turbo Pascal สิ่งแวดล้อมแบบรวม หมายถึง ระบบร่วมกันสำหรับการบรรณาธิกร การแปลโปรแกรม การเชื่อมโยง และการบรรจุโปรแกรม (Integrated environment is a coordinated system for editing, compiling, linking and loading a program.)

การใช้คอมพิวเตอร์ เราต้องโต้ตอบกับโปรแกรมกำกับดูแล (supervisory program) เรียกว่าระบบปฏิบัติจริง (operating system) ระบบปฏิบัติการร่วมสองชุด สำหรับ IBM PC ได้แก่ MS-DOS (Micorsoft Disk Operating System) และ Microsoft Windows (หรือ Windows)

ระบบปฏิบัติการจัดหาบริการที่สำคัญ (essential services) หลายชนิดให้กับผู้ใช้คอมพิวเตอร์ ได้แก่

(1) การบรรจุและการวิ่งโปรแกรมประยุกต์ (Loading and runnig application programs)

(2) การจัดสรรหน่วยความจำและเวลาของตัวประมวลผล (Allocating memory and processor time)

(3) จัดหาสิ่งอำนวยความสะดวกอินพุตและเอาต์พุต (Providing input and output facilities)

(4) การจัดการแฟ้มสารสนเทศ (Managing files of information)

ระบบปฏิบัติการ หมายถึง โปรแกรมกับผู้ใช้โปรแกรมโต้ตอบกันเพื่อกำหนดว่าโปรแกรมประยุกต์ชุดใด และ/หรือการดำเนินการของระบบชุดใดที่คอมพิวเตอร์ควรจะกระทำ (Operating system is the program with which the program user interacts in order to specify which application programs and/or system operations the computer should perform.)

การปลุกเครื่องคอมพิวเตอร์และการใส่ Turbo Pascal (Booting the Computer and Entering Turbo Pascal)

เมื่อเราเปิดสวิตช์ (switch on) หรือปลุกเครื่อง (boot up) คอมพิวเตอร์ส่วนบุคคลระบบปฏิบัติการจากฮาร์ดดิสก์ปกติดั้งเป็น drive C จะถูกบรรจุเข้าไปในหน่วยความจำหลักและเริ่มต้นกระทำการ ถ้าเราใช้ MS-DOS ระบบปฏิบัติการจะให้ตัวพร้อมดังนี้

C>

เพื่อแจ้งว่ามันพร้อมแล้วสำหรับให้เราพิมพ์การดำเนินการถัดไป

ใส่คำสั่งงานต่อไปนี้

C>CD\BP\BIN

C>Turbo

เพื่อใส่สิ่งแวดล้อมแบบรวมของ Turbo Pascal คำสั่งงานชุดแรก (CD, สำหรับเปลี่ยน directory)

ทำสารบบย่อย \BP\BIN ซึ่งประกอบด้วย Turbo Pascal ให้เป็นสารบบดิสก์ใช้งาน (active disk directory) คำสั่งงานนี้คือขึ้นอยู่กับการติดตั้ง ดังนั้นเราอาจจำเป็นต้องใส่สารบบย่อยที่แตกต่าง (ตัวอย่างเช่น \TP\BIN) การพิมพ์ Turbo ทำให้เราอยู่ในสิ่งแวดล้อมแบบล้อมของ Turbo Pascal ถ้าเราใช้ระบบปฏิบัติการ Windows พิมพ์คำสั่งงานดังนี้

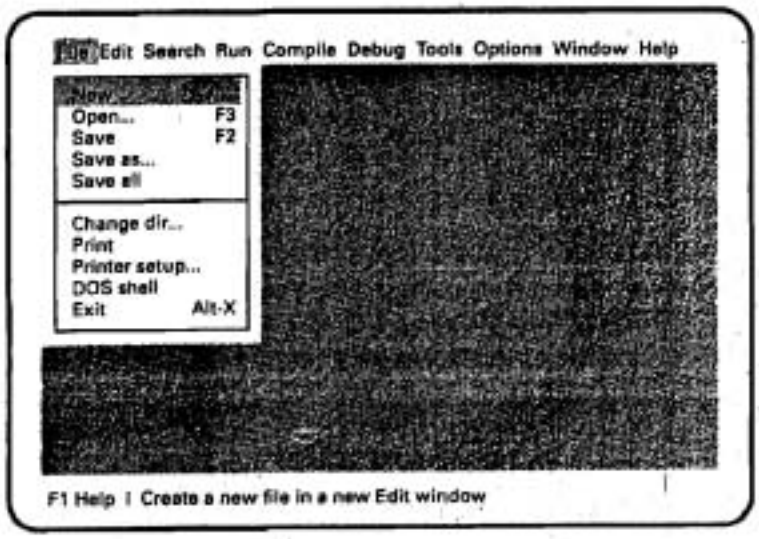
C>Win

เราจะเห็นจอภาพแสดงสัญรูปหรือไอคอน (icons) ซึ่งเป็นรูปภาพเล็กๆ สำหรับโปรแกรมประยุกต์หลายรูป การใส่สิ่งแวดล้อมแบบรวมของ Turbo Pascal ให้ย้ายเมาส์เคอร์เซอร์ไปยังสัญรูป Turbo Pascal และคลิกที่ปุ่มซ้ายของเมาส์สองครั้งอย่างเร็ว (ดับเบิลคลิก)

การปลุกเครื่อง หมายถึง กระบวนการเปิดสวิตช์ของคอมพิวเตอร์ และบรรจุระบบปฏิบัติการไว้ในหน่วยความจำ (Bootin up is the process of switching a computer on and looding the operating system into memory.)

รายการเลือกของ Turbo Pascal (Turbo Pascal Menus)

เมื่ออยู่ในสิ่งแวดล้อมแบบรวมของ Turbo Pascal แล้ว เราสามารถสร้าง (create) โปรแกรมใหม่ ดัดแปรโปรแกรมเก่าและคอมไพล์และวิ่งโปรแกรมเหล่านี้ เราสามารถโต้ตอบกับสิ่งแวดล้อมผ่านทางรายการเลือกบนจอภาพ (menu screens) บรรทัดบนสุดในรูป 1.7 คือ รายการเลือกหลักของ Turbo Pascal (File, Edit, Search, Run เป็นต้น) ถ้าคอมพิวเตอร์ของเรามีเมาส์ เราสามารถเลือก item ของรายการเลือกหลักโดยย้าย mouse cursor ไปยังตำแหน่งของ item ตัวนั้น จากนั้นคลิกปุ่มซ้ายมือบนเมาส์ กรณีอื่นๆ เราสามารถเลือกงานของรายการเลือกหลัก โดยกดปุ่ม Alt และกดปุ่มอักษรตัวแรกของชื่องาน (ตัวอย่างเช่น Atl-F สำหรับ File)



รูป 1.7 รายการเลือกของ Turbo Pascal

เราสามารถย้ายจากงานจากรายการเลือกหลัก ไปยังอีกที่หนึ่งโดยการกดปุ่มลูกศรทางซ้ายมือและลูกศรทางขวามือ

จอภาพรายการเลือก หมายถึง จอภาพซึ่งแสดงส่วนละเว้นต่างๆ ซึ่งเราทำการเลือกโดยใช้เมาส์หรือกดฟังก์ชันคีย์บนคีย์บอร์ด (Menu screen is a screen display showing several options from which you make a selection by using a mouse or pressing keyboard function keys.)

จอภาพในรูป 1.7 แสดงให้เห็นว่าเกิดอะไรขึ้น เมื่อเราเลือก File แสดงรายการของงาน (New, Open, Save, Save as ... เป็นต้น) งานย่อยทั้งหมดนี้เกี่ยวข้องกับ File ปรากฏบนรายการเลือกแบบดึงลง (pull-down menu)

รายการเลือกแบบดึงลงที่แตกต่างปรากฏสำหรับงานรายการเลือกหลัก (main menu task) แต่ละงานที่เราเลือก เราสามารถใช้เมาส์เลือกงานย่อย (subtask) จากรายการเลือกแบบดึงลง หรือเราสามารถย้ายแถบที่เน้นให้เด่น (highlight bar) บนงานเฉพาะ (โดยใช้ปุ่มลูกศรชี้ขึ้นหรือชี้ลง) จากนั้นกดปุ่ม Enter เราสามารถเลือกงานจากรายการเลือกแบบดึงลงโดยพิมพ์อักษรตัวพิมพ์ใหญ่ (capitalized letter) สำหรับงานนั้นได้ด้วย (ตัวอย่างเช่น N สำหรับ New) หรือการกดฟังก์ชันคีย์ที่แสดงบนขวามือของงาน (ตัวอย่างเช่น F3 สำหรับ Open)

รายการเลือกแบบดึงลง หมายถึง จอภาพรายการเลือกซึ่งปรากฏขึ้นเมื่อเราเลือกงานรายการเลือกเฉพาะ (Pull-down menu is a menu screen that appears when you select a particular menu task.)

บรรทัดล่างสุดของจอภาพอธิบายงานซึ่งเราเน้นให้เด่น (highlight) (New-Creat a new file ...) และแสดงรายการงานอื่นๆ ซึ่งเราสามารถเลือกได้ (ตัวอย่างเช่น กด F1 สำหรับ Help) บรรทัดล่างสุดจะเปลี่ยนเมื่อเราย้ายจากส่วนหนึ่งของสิ่งแวดล้อม turbo Pascal ไปยังอีกส่วนหนึ่ง

พื้นที่แรเงาในตอนกลางของรูป 1.9 คือพื้นที่ทำงาน (desktop area) ซึ่งเราจะเปิดหลายหน้าต่างเพื่อใช้สร้างและทดสอบโปรแกรม Pascal ของเรา

คำอธิบายแบบเชื่อมตรง (On-Line Help)

ผ่านทางระบบคำอธิบายแบบเชื่อมตรงของมัน Turbo Pascal จัดหาสารสนเทศเพิ่มเติมเกี่ยวกับงานรายการเลือกใดๆ ก็ตาม ถ้าเราต้องการสารสนเทศเกี่ยวกับตัวเลือก

ของรายการเลือกเฉพาะ (a particular menu option) เน้นให้เด่น ตัวเลือกของรายการเลือกนั้น (โดยใช้เมาส์หรือปุ่มลูกศร) จากนั้นกดปุ่ม F1 หน้าต่างคำอธิบายจะผุดขึ้น (pop up) บนจอภาพแสดงสารสนเทศเชิงพรรณนาเกี่ยวกับการดำเนินการนี้

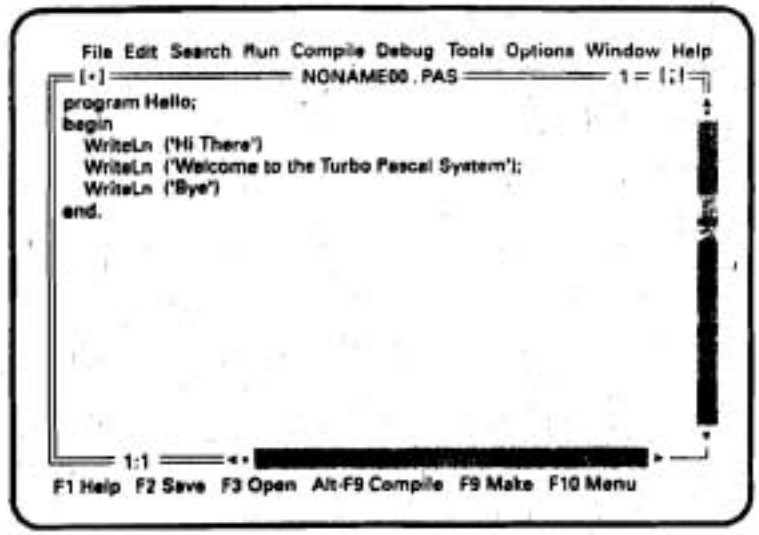
การออกจาก (exit) หน้าต่าง Help และกลับคืนยังจอภาพปัจจุบันของเรา ให้เลือก Cancel หรือกดปุ่ม Esc (Escape) ตำแหน่งบนสุดขวามือของคีย์บอร์ด

การสร้างโปรแกรมใหม่ (Creating a New Program)

การสร้างโปรแกรม Pascal ชุดใหม่ เราต้องเริ่มต้นด้วยหน้าต่าง Edit ว่างบน Desktop โดยการเปิดหน้าต่างใหม่ เลือกตัวเลือก New จากรายการเลือก File เช่นที่แสดงในรูป 1.7 Turbo Pascal 7.0 จะให้หน้าต่าง edit ชื่อ NONAME00.PAS

เมื่อ Turbo Pascal ให้หน้าต่าง edit ว่างกับเราแล้ว เริ่มต้นเราใส่โปรแกรมครั้งละหนึ่งบรรทัด กดปุ่ม Enter หลังจากพิมพ์บรรทัดโปรแกรมแต่ละบรรทัด และใช้ปุ่มลูกศรบนคีย์บอร์ด เพื่อกำหนดตำแหน่งเคอร์เซอร์ทุกแห่งบนจอภาพ

แก้ไขข้อผิดพลาดเนื่องพิมพ์ผิด โดยการกดปุ่ม backspace เพื่อลบตัวอักขระทั้งหมดจากตำแหน่งเคอร์เซอร์ปัจจุบันไปจนถึงตัวอักขระที่ผิด จากนั้นใส่ตัวอักษรที่ถูกต้องในตำแหน่งของมัน รูป 1.8 แสดงให้เห็นโปรแกรมที่เสร็จสมบูรณ์ (แต่ยังมีที่ผิด) ในหน้าต่าง edit ชื่อ NONAME00.PAS



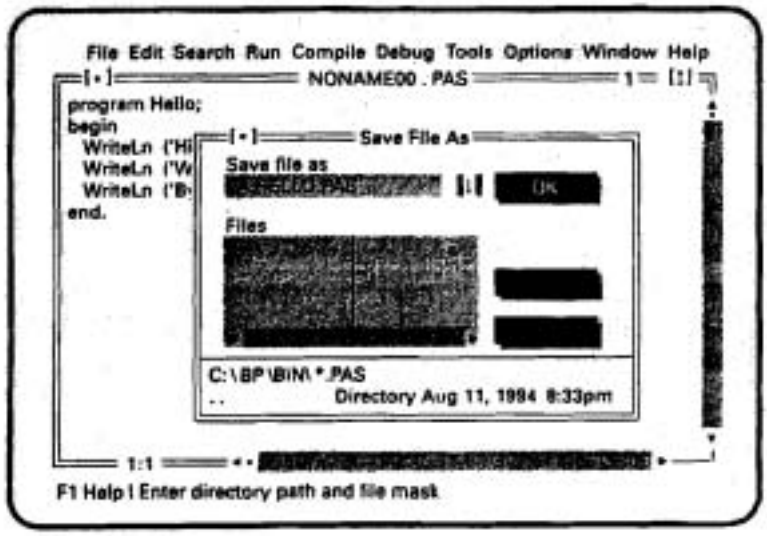
รูป 1.8 โปรแกรมสมบูรณ์แต่ยังมีที่ผิด

หลังจากโปรแกรมของเราเสร็จแล้ว ใส่แผ่นฟล็อปปีดิสก์ซึ่ง formatted แล้ว (ดูภาคผนวก A สำหรับคำสั่งการ format) ในหน่วยขับฟล็อปปีดิสก์ และ save โปรแกรมของเราบนดิสก์ โดยการเลือก save หรือโดยการกดปุ่น F2 ณ จุดนี้โปรแกรมของเรากำหนดชื่อเป็น NONAME00.PAS โดย Turbo Pascal ก่อนการ save โปรแกรมของเราด้วยชื่อนี้ turbo Pascal ให้โอกาสเราใช้ชื่อที่มีความหมายเหมาะสมมากกว่า ตัวอย่างเช่น ถ้าเราต้องการ save โปรแกรมของเราเป็นแฟ้มชื่อ HELLO บนหน่วยขับดิสก์ A, พิมพ์ A : HELLO.PAS เมื่อ Turbo Pascal ให้ตัวพร้อม เช่นที่แสดงในรูป 1.9

A : หมายถึง กำหนดหน่วยขับดิสก์ของคอมพิวเตอร์

HELLO หมายถึง ชื่อแฟ้ม และ .PAS คือส่วนขยายระบุว่าแฟ้มนี้ประกอบด้วยรหัสต้นฉบับภาษา Pascal

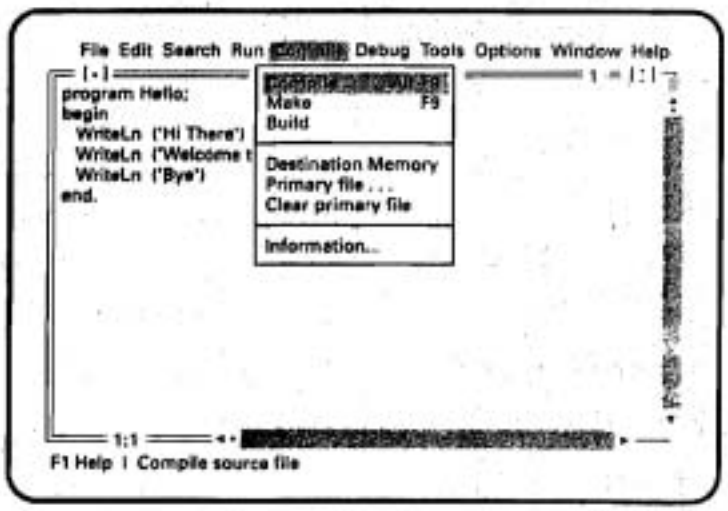
ชื่อแฟ้มสำหรับ MS-DOS (ตัวอย่างเช่น HELLO) เป็นกลุ่มของตัวอักษร เลขโดด หรืออักขระพิเศษ (ไม่ใช่ period หรือ space) ซึ่งมีความยาวไม่เกินแปดตัวอักษร ถ้าเราพิมพ์ส่วนขยายของแฟ้มผิดพลาด Turbo Pascal จะใส่ส่วนขยายแฟ้มเป็น .PAS ให้อัตโนมัติ ขณะนี้เราพร้อมแล้วที่จะ save แฟ้มโปรแกรมและกลับคืน (return) ไปเมนูหลัก ซึ่งทำได้โดยเลือก OK (คลิกปุ่ม OK)



รูป 1.9 การ save โปรแกรม

การคอมไพล์และการวิ่งโปรแกรม (Compiling and Running a Program)

การคอมไพล์โปรแกรม ให้เลือกเมนู Compile จากแถบรายการเลือกหลัก รูป 1.10 แสดงตัวเลือกต่างๆ ในเมนู Compile เนื่องจากเราต้องการคอมไพล์โปรแกรมของเรา เลือกตัวเลือก Compile จากนั้น Turbo Pascal จะเริ่มต้นคอมไพล์โปรแกรม ซึ่งแสดงอยู่ในหน้าต่าง Edit



รูป 1.10 การคอมไพล์โปรแกรม

ถ้าเราพยายามให้คอมไพล์โปรแกรมซึ่งแสดงในรูป 1.8 Turbo Pascal จะแสดงข้อความระบุข้อผิดพลาดวากยสัมพันธ์ดังนี้

Error 85 : " ; " expected

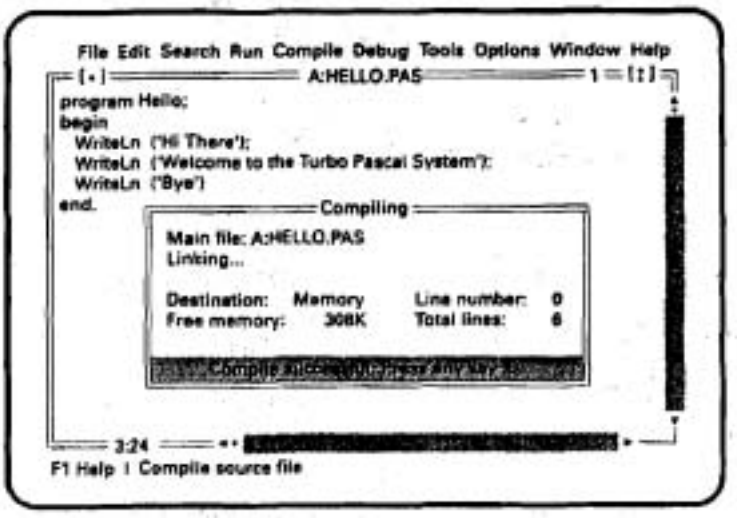
และกลับคืนไปยังหน้าต่าง Edit ตำแหน่งของเคอร์เซอร์อยู่ที่จุดในโปรแกรมซึ่งการประมวลผลการคอมไพล์หยุด (ที่ WriteLn บรรทัดที่สี่) Error 85 อธิบายว่าเราลืมใส่เครื่องหมาย semicolon ตอนจบบรรทัดที่สามของข้อความโปรแกรม (program text)

หลังจากเปลี่ยนแปลงบรรทัดที่สามเป็น

WriteLn ('Hi There');

save โปรแกรมชุดปรับแก้ไขไว้บนดิสก์โดยการเลือก save จากเมนูหลัก เลือกเมนู compile อีกครั้งหนึ่งจากนั้นเลือกตัวเลือก Compile ขณะนี้ในโปรแกรมไม่มีข้อผิดพลาดใดๆ และหน้าต่างสถานะการคอมไพล์จะมีข้อความว่า

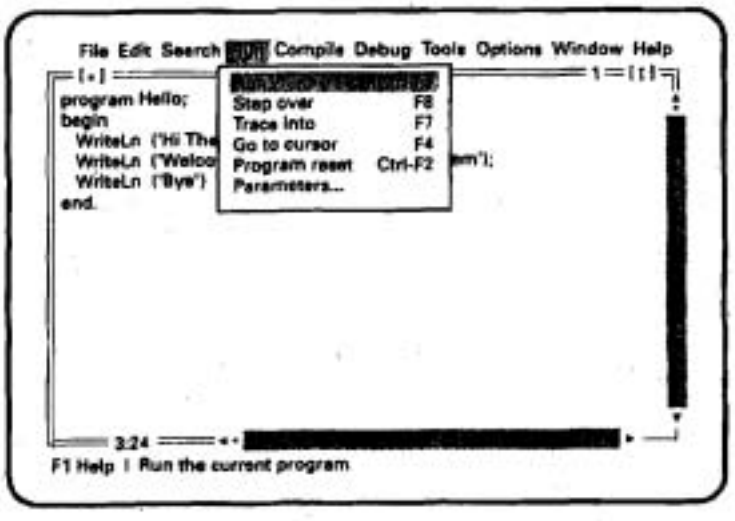
Compile successful : Press any key (รูป 1.11)



รูป 1.11

ให้กดหนึ่งปุ่มบนคีย์บอรด์

การวิ่ง (หรือกระทำการ) โปรแกรมจากเมนูหลักหลังจากคอมไพล์โปรแกรมแล้ว เลือกเมนู Run รูป 1.12 แสดงตัวเลือกต่างๆ ที่มีอยู่ในเมนู Run เนื่องจากเราต้องการวิ่งโปรแกรม จึงเลือกตัวเลือก Run



รูป 1.12

Turbo Pascal เริ่มต้นกระทำการโปรแกรม เมนูหลักจะถูกแทนที่โดยจอภาพผู้ใช้ (user screen) ซึ่งมีเอาต์พุตของโปรแกรมดังนี้

Hi There

Welcom to the Turbo Pascal System

Bye

จากนั้นจะกลับคืนมายังเมนูหลัก การดูทบทวน (rewiew) เอาต์พุตโปรแกรม ให้กดปุ่ม Alt และปุ่ม F5 พร้อมกัน (Alt-F5) จอภาพผู้ใช้จะปรากฏเอาต์พุตโปรแกรมอีกครั้งหนึ่ง และอยู่เช่นนั้นจนกระทั่งเรากดปุ่มใดๆ หนึ่งปุ่มบนคีย์บอร์ด ซึ่งจะกลับคืนไปจอภาพของเมนูหลัก

การดูเอาต์พุตโปรแกรมใน Windows (Viewing Program Output in windows)

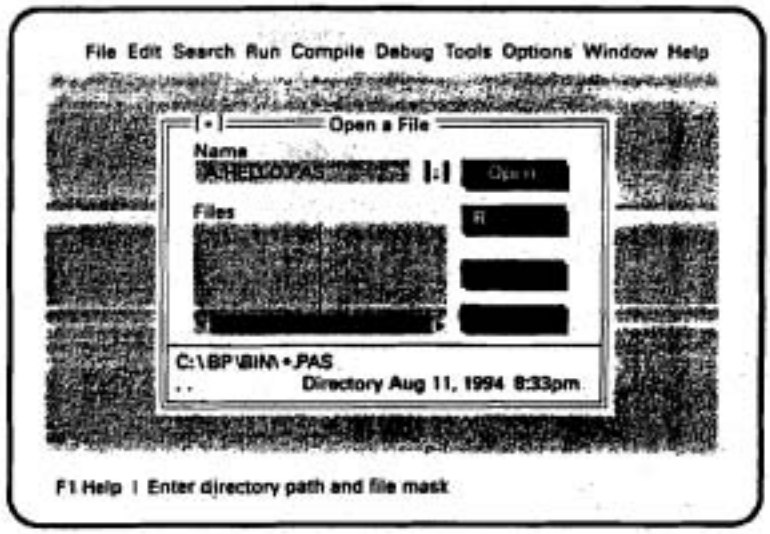
เมื่อใช้ Windows เราต้องใส่บรรทัด

uses WinCrt;

เป็นบรรทัดที่สองของทุกโปรแกรมที่เราเขียน บรรทัดนี้จะให้หน้าต่างที่มีเอาต์พุตโปรแกรมของเราปรากฏบนจอภาพ ถ้าไม่มีบรรทัดนี้ เอาต์พุตโปรแกรมจะไม่ถูกแสดงผล

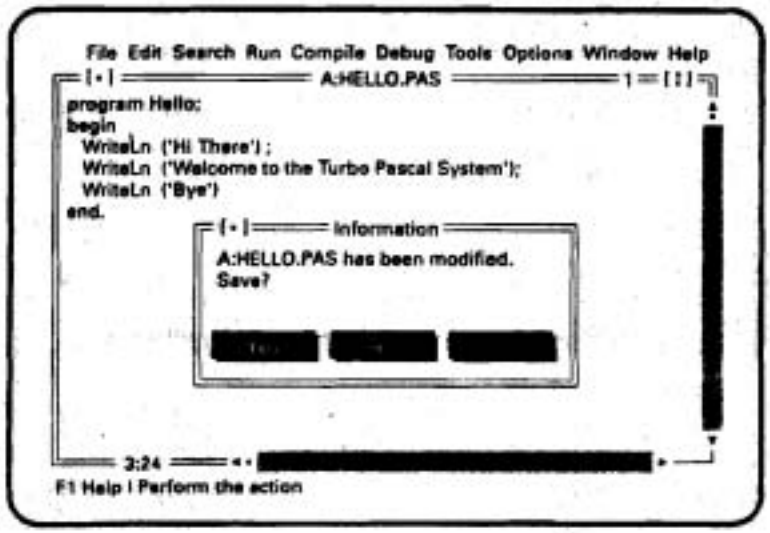
การบรรจุแฟ้ม (Loading a File)

การบรรจุแฟ้มซึ่ง save ไว้ก่อนหน้าแล้ว ไปยังหน้าต่าง Edit เราต้องให้ Turbo Pascal แสดงกล่องโต้ตอบ Open a File (รูป 1.13) การทำสิ่งนี้อาจทำผ่านเมนู File (เลือกตัวเลือก Open) หรือกดปุ่มพังก์ชันคีย์ F3 เมื่อปรากฏกล่องโต้ตอบอาจจะพิมพ์ชื่อแฟ้มลงในแถบที่มีเลเบล Name หรือเลือกชื่อแฟ้มจากรายการซึ่งแสดงใต้เลเบล Files ถ้าเราไม่มีเมาส์ให้กดปุ่ม Tab เพื่อเข้าถึงรายการของแฟ้ม จากนั้นใช้ปุ่มลูกศรชี้ขึ้นหรือชี้ลงเลือกแฟ้มที่ต้องการ



รูป 1.13

การออกจาก Turbo Pascal (Exiting Turbo Pascal) ตัวเลือก Exit จากเมนู File เป็นวิธีให้เราออกจาก Turbo Pascal และกลับคืนไปยังระบบปฏิบัติการ ถ้าเราเลือกตัวเลือก Exit และมีการเปลี่ยนแปลงโปรแกรมของเราในหน้าต่าง Edit แต่เราลืม save โปรแกรมที่มีการปรับแก้ไข Turbo Pascal ให้โอกาสเราเป็นครั้งสุดท้าย เพื่อให้ save โดยแสดงกล่องโต้ตอบคล้ายกับที่แสดงในรูป 1.14



รูป 1.14

ถ้าเรากดอักษร y บนคีย์บอร์ด โปรแกรมของเราจะถูก save ก่อนออกจาก Turbo Pascal การออกจากเมนู File และกลับคืนไปเมนูหลัก กดปุ่ม Esc (Escape) หรือ F10

การฝึกปฏิบัติที่ดีคือ ทำให้ว่าง (clear) desktop ของ Turbo Pascal ก่อนเลิก (quiting) เราทำสิ่งนี้โดยการเลือก Window จากเมนูหลัก จากนั้นเลือก Close all จากรายการเลือกแบบดึงลง เพื่อปิดหน้าต่างเปิดทั้งหมด และทำให้ desktop ว่าง

แบบฝึกหัด Quick-Check

จงเติมคำศัพท์ที่ถูกต้องในช่องว่างต่อไปนี้

1. The ..... translates a(n) ..... language program into ..... program.

2. A(n) ..... provides access to system programs for editing, compiling, and so on.

3. Specify the correct order for these operations : exceution, translation, linking/loading.

4. A high-level language program is saved on disk as a(n) ..... file or a(n) ..... file.

5. The ..... finds syntax errors in the ..... file.

6. Before linking, a machine language program is saved on disk as a(n) ..... file.

7. After linking, a machine language program is saved on disk as a(n) ..... file.

8. Computer program are ..... components of a computer system while a disk drive is .....

9. In a high-level language, you can reference data using ..... rather than memory cell addresses.

10. จงบอกว่าสิ่งต่อไปนี้ข้อใดเป็นคุณสมบัติประยุกต์ให้หน่วยความจำหลักหรือเป็นคุณสมบัติที่ประยุกต์ให้หน่วยความจำรอง

a) Faster to access

b) Volatile

c) May be extended writhout limit

d) Less expensive

e) Used to store ffiles

f) Central processor accesses it to obtion the next machine language instruction for execution

g) Provides semipermanent data storage