บทที่ 2

ทฤษฎีและแนวคิดเกี่ยวกับโปรแกรมบรรณาธิกร

เครื่องมือในการพัฒนาโปรแกรม

เครื่องมือที่ใช้ในการพัฒนาโปรแกรมทางซอฟต์แวร์ มีหลายชนิดด้วยกัน เช่น

1. โปรแกรมบรรณาธิกร (11) (Editing program) เป็นชุดคำสั่งหรือโปรแกรม ที่ออกแบบมาเพื่อช่วยให้ผู้ใช้สามารถทำการแก้ไข ปรับปรุง หรือเรียบเรียง โปรแกรมอื่นที่ผู้ใช้ต้องการได้โดยสะดวกและมีปฏิภาคกับผู้ใช้โดยตรง ตัวอย่างเช่น โปรแกรมอีดิตของบริษัทไมโครซอฟต์ โปรแกรมคิวอีดิต ของษริษัทเซ็มแวร์ เป็นต้น

2. โปรแกรมแก้ไขจุดบกพร่อง (12) (Debugging aids program) เป็นชุดคำสั่งที่เขียนขึ้นเพื่อเตรียมไว้ให้คอมพิวเตอร์ซี้หาจุดบกพร่อง หรือที่ผิดในโปรแกรมอื่น ตัวอย่างเช่นโปรแกรมเทอร์โบดีบักเก้อ (Turbo Debugger) ของบริษัทบอร์แลนด์

3. โปรแกรมติดตาม (13) (Trace program หรือ Codeview) เป็นชุดคำสั่งที่พิมพ์รายการแสดงในโปรแกรมต่างๆ (Listing) เพื่อจะได้นำมาแก้ไขหากมีสิ่งที่ผิดพลาด ตัวอย่างเช่น โปรแกรมโค๊ดวิว (Codeview) ของบริษัทไมโครซอฟต์

4. โปรแกรมอรรถประโยชน์ (11) (Utility program) เป็นชุดคำสั่งระบบ (System software) ที่มีผู้เขียนและเก็บไว้ในตัวเครื่องคอมพิวเตอร์ เพื่ออำนวยความสะดวกแก่ผู้ใช้ เนื่องจากเป็นโปรแกรมที่ผู้ใช้ส่วนมากจะใช้อยู่บ่อยๆ เช่น โปรแกรมเรียงลำดับ โปรแกรมย้ายข้อมูล เป็นต้น

โปรแกรมที่ทำการพัฒนาในการวิจัยครั้งนี้ ได้เลือกเฉพาะโปรแกรมบรรณาธิกรเท่านั้น ทั้งนี้เพราะเครื่องมือที่ใช้ในการพัฒนาโปรแกรมทั้งหมด โปรแกรมบรรณาธิกรเป็นโปรแกรมที่มีความจำเป็นและมีความสำคัญสำหรับเครื่องคอมพิวเตอร์มากที่สุด และเป็นโปรแกรมพื้นฐานในการพัฒนาโปรแกรมอื่นๆต่อไปได้

ประเภทของโปรแกรมบรรณาธิกร

โปรแกรมบรรณาธิกรที่พบเห็นกันโดยทั่วๆไป สามารถแบ่งออกได้เป็น 2 ประเภทด้วยกันคือ (14)

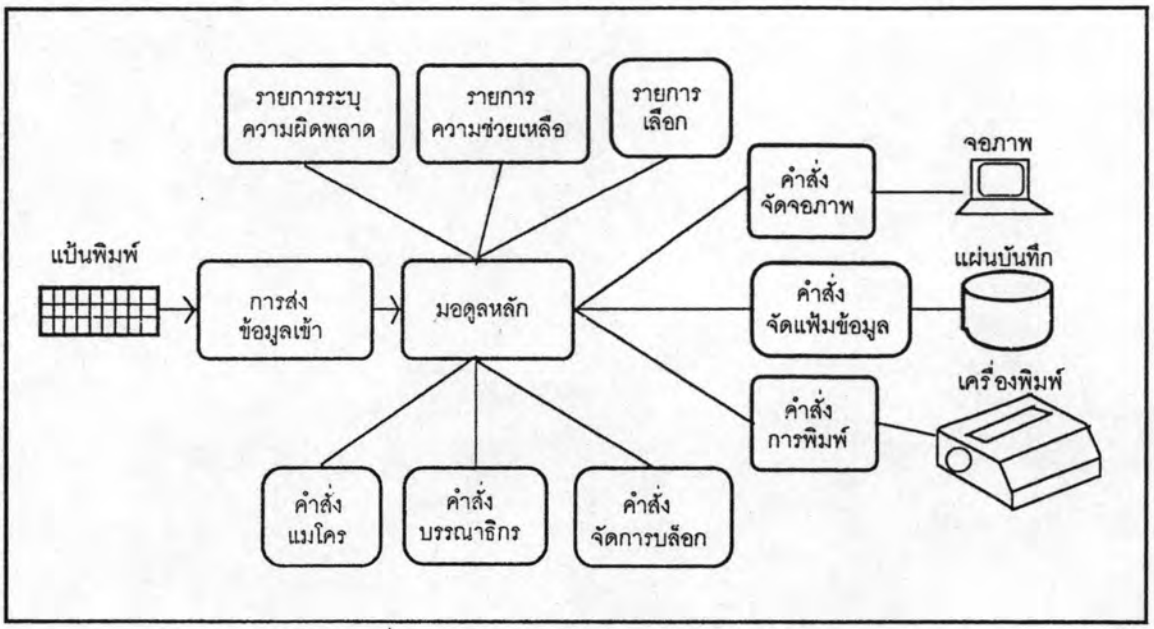
1. โปรแกรมบรรณาธิกรบรรทัด (11) (Line editor) เป็นชุดคำสั่งบรรณาธิกรที่ผู้ใช้สามารถทำการแก้ไข ปรับปรุง เปลี่ยนแปลงหรือเรียบเรียงข้อความ ได้ทีละบรรทัดและเมื่อทำการแก้ไขปรับปรุงเสร็จก็สามารถเลื่อนไปยังบรรทัดอื่นในโปรแกรมได้ โดยการใช้คำสั่งควบคุมที่ถูกกำหนดไว้ในโปรแกรมบรรณาธิกรนั้น ส่วนใหญ่แล้วคำสั่งควบคุมที่มีให้ใช้อยู่ จะมีคำสั่งให้เลือกใช้ได้ไม่กี่คำสั่ง โดยมีเฉพาะคำสั่งควบคุมที่จำเป็นต้องใช้จริงๆเท่านั้น ดังนั้นจึงทำให้การใช้งานทำได้ไม่สะดวก ขาดความคล่องตัว และยังประสบกับปัญหาในการเลื่อนบรรทัดเพื่อจะแก้ไขข้อความที่กำลังทำงานอยู่ เพราะสามารถทำการบรรณาธิกรข้อความได้ครั้งละบรรทัดเท่านั้น ที่เป็นเช่นนี้เพราะว่าโปรแกรม บรรณาธิกรบรรทัดมักจะถูกเขียนขึ้นใช้ในยุคแรกๆ ของการใช้เครื่องคอมพิวเตอร์ ซึ่งในยุคนั้นจำเป็นต้องใช้ทรัพยากรต่างๆอย่างคุ้มค่าที่สุด เนื่องจากอุปกรณ์คอมพิวเตอร์มีราคาแพงหาได้ยากและมีอยู่น้อยดังนั้นผู้เขียนโปรแกรมบรรณาธิกร จึงต้องเขียนโปรแกรมอย่างง่ายๆให้พอใช้งานได้ โดยใช้หน่วยความจำให้น้อยที่สุด เพื่อให้มีหน่วยความจำเหลือพอที่จะทำงานอย่างอื่นต่อไปได้จึงใช้งานได้ไม่สะดวก มาจนถึงในยุคปัจจุบันโปรแกรมบรรณาธิกรบรรทัด ก็ยังคงมีใช้กันอยู่ โดยได้พัฒนาให้มีความสามารถมากขึ้น ตัวอย่างเช่น บนระบบปฏิบัติการ ชื่อว่า ยูนิกซ์ (15) (UNIX) จะมีโปรแกรมชื่อว่าอีดี (ed) หรือในระบบปฏิบัติการเอ็มเอสดอส (MSDOS) ก็จะมีโปรแกรมบรรณาธิกรชื่อว่า เอ็ดลิน (EDLIN.EXE) โปรแกรมเหล่านี้แม้จะมีอยู่ในปัจจุบันแต่ ก็มีผู้นิยมใช้น้อย ซึ่งส่วนใหญ่จะใช้ก็ต่อเมื่อจำเป็นหรือเมื่อไม่มีโปรแกรมบรรณาธิกรชนิดอื่นให้ใช้เท่านั้น

2. โปรแกรมบรรณาธิกรจอภาพ (16) (Screen editor) เป็นชุดคำสั่งบรรณาธิกรที่ช่วยให้ผู้ใช้สามารถทำการแก้ไข ปรับปรุง เปลี่ยนแปลง หรือเรียบเรียงข้อความ โดยสามารถแสดงข้อความได้เต็มจอภาพซึ่งผู้ใช้สามารถเคลื่อนย้ายตัวชี้ตำแหน่ง (Cursor) ไปกระทำการบรรณาธิกรหรือแก้ไข ณ ตำแหน่งใดๆในจอภาพได้ ดังนั้น ในการใช้งานโดยทั่วๆไป โปรแกรมบรรณาธิกรจอภาพจึงมักมีผู้นิยมใช้มากกว่าโปรแกรมบรรณาธิกรบรรทัด เนื่องจากใช้งานได้สะดวก มีความคล่องตัวในการใช้งาน มีคำสั่งต่างๆให้เลือกใช้ได้มากกว่า ในปัจจุบันมีผู้ทำการพัฒนาโปรแกรมบรรณาธิกรจอภาพกันมาก ทำให้โปรแกรมบรรณาธิกรจอภาพมีคุณภาพที่ดีขึ้น ตัวอย่าง โปรแกรมบรรณาธิกรจอภาพ เช่น บนระบบปฏิบัติการยูนิกซ์มี (15) โปรแกรมบรรณาธิกรชื่อ วีไอ (vi) หรือ อีแมค (17) (EMAC) บนระบบปฏิบัติการเอ็มเอสดอสมีบรรณาธิกรชื่อ อีดิต (EDIT.EXE) และยังมีผลิตภัณฑ์จากอีกหลายบริษัท

นอกจากโปรแกรมบรรณาธิกรทั้งสองแบบแล้ว ก็ยังมีการนำเอาโปรแกรมบรรณาธิกรเหล่านี้ไปใช้ร่วมกับโปรแกรมชนิดอื่นอีก เช่น นำไปใช้งานร่วมกับโปรแกรมแปลภาษา (Compiler) ตัวอย่าง เช่น โปรแกรมเทอร์โบซี (6) (Turbo C) โปรแกรมภาษาเบสิก (BASIC) โปรแกรมเทอร์โบปาสคาล (Turbo Pascal) เป็นต้น หรือนำไปใช้ร่วมกับโปรแกรมสำเร็จรูปอื่น เช่น โปรแกรมดีเบส (dBASE) โปรแกรมโลตัส (Lotus 123) เป็นต้น และยังนำไปใช้เป็นพื้นฐานในการพัฒนาโปรแกรมประมวลผลคำ (Word processor) เช่น โปรแกรมเวิร์ดสตาร์ (WordStar) โปรแกรมเวิร์ดเพอร์เฟก (Wordperfect) เป็นต้น ในการทำวิจัยครั้งนี้ได้ทำการพัฒนาเฉพาะโปรแกรมบรรณาธิกรจอภาพเท่านั้น เพราะเป็นโปรแกรมบรรณาธิกรชนิดที่มีผู้นิยมใช้กันมากที่สุด เนื่องจากใช้งานได้สะดวกกว่าโปรแกรมบรรณาธิกรบรรทัด หรือโปรแกรมบรรณาธิกรชนิดอื่นๆ

ระบบของโปรแกรมบรรณาธิกร (16)

ระบบของโปรแกรมบรรณาธิกร ประกอบไปด้วยมอดูลย่อยๆหลายมอดูลด้วยกัน เพื่อให้สะดวกต่อการทำการพัฒนาโปรแกรม โดยมอดูลเหล่านี้สามารถแบ่งเป็นกลุ่มๆใหญ่ตามหน้าที่การทำงานได้ ดังรูปที่ 2.1



รูปที่ 2.1 ระบบของโปรแกรมบรรณาธิกร

จากรูป เป็นการแสดงถึงระบบของโปรแกรมบรรณาธิกร ซึ่งอธิบายได้ดังนี้

1. เริ่มต้นจากผู้ใช้ติดต่อกับโปรแกรม โดยการส่งข้อมูลหรือความต้องการของผู้ใช้ ผ่านทางแป้นพิมพ์ และโปรแกรมจะรับข้อมูลของผู้ใช้โดยผ่านทางมอดูลการส่งข้อมูลเข้า (Key entry) แล้วส่งต่อไปยังมอดูลหลัก

2. มอดูลหลัก (Main module) เป็นมอดูลที่จัดการกับคำสั่งที่รับเข้ามา แล้วเลือกทำงานตามประเภทของคำสั่ง โดยส่งข้อมูลต่อไปยังมอดูลที่ถูกต้องตรงกับหน้าที่ในการทำงานที่ผู้ใช้ต้องการต่อไป

3. คำสั่งแมโคร (Macro command) เป็นกลุ่มมอดูลที่เก็บคำสั่งในการทำงาน ไว้หลายๆ คำสั่งเพื่อให้เรียกใช้ชุดคำสั่งที่ใช้อยู่บ่อยๆได้สะดวกขึ้น โดยการกำหนดแป้นพิมพ์ที่ไม่ใช่อักขระปกติ ยกตัวอย่างเช่น แป้นอักขระ F9 หรือแป้นอักขระ F4 เป็นต้น เพื่อใช้เป็นแป้นอักขระที่จะเรียกใช้ชุดคำสั่งนั้นๆ ชุดคำสั่งแมโครที่ผู้ใช้ได้กำหนดขึ้นนี้สามารถทำการบันทึกลงแฟ้มได้ และเรียกกลับมาใช้ใหม่ได้อีกเมื่อผู้ใช้ต้องการ

4. คำสั่งบรรณาธิกร (Editor command) เป็นกลุ่มคำสั่งที่ใช้ในการจัดการประมวลผลข้อมูลที่เก็บไว้ในหน่วยความจำหลักภายในเครื่องคอมพิวเตอร์ อันได้แก่การแก้ไข เปลี่ยนแปลง

เคลื่อนย้าย ปรับปรุง การเพิ่มเติม หรือการลบข้อมูลที่เก็บอยู่

5. คำสั่งจัดการบล็อก (Block command) เป็นกลุ่มคำสั่งที่ใช้ในการประมวลผลข้อมูลที่ถูกกำหนดบล็อกไว้แล้ว (Mark block) โดยการทำงานจะทำเฉพาะในส่วนที่เป็นบล็อกเท่านั้น ในส่วนอื่นที่ไม่ได้เลือกไว้ก็จะไม่ถูกเปลี่ยนแปลง ในกลุ่มคำสั่งนี้ได้รวมกลุ่มคำสั่งที่เกี่ยวกับการจัดการพื้นที่เก็บข้อมูลชั่วคราว (Clipboard) และกลุ่มคำสั่งที่เกี่ยวกับการจัดการหน่วยความจำชั่วคราว (Scrap buffer) ไว้ด้วย

6. คำสั่งการพิมพ์ (Print command) เป็นคำสั่งที่สั่งให้เครื่องคอมพิวเตอร์นำข้อมูลที่เก็บอยู่ในหน่วยความจำหลักมาแสดงออกทางเครื่องพิมพ์ (Printer) เพื่อเป็นสำเนาถาวร (Hard copy)

7. คำสั่งจัดแฟ้มข้อมูล (File command) เป็นกลุ่มคำสั่งที่จัดการกับแฟ้มข้อมูล เพื่อทำการเปิดและปิดแฟ้ม หรือสร้างแฟ้มใหม่ หรือบันทึกแฟ้มที่กำลังใช้งานอยู่ โดยจะทำการติดต่อกับหน่วยบันทึกจานแม่เหล็ก (Disk drive)

8. คำสั่งจัดจอภาพ (Screen command) เป็นกลุ่มคำสั่งที่มีผลแต่เฉพาะ การแสดงออกทางจอภาพเท่านั้น เพื่อให้ผู้ใช้ได้รับความสะดวกในการบรรณาธิกร ในส่วนนี้จะเป็นส่วนที่ทำการควบคุมจอภาพโดยตรงจากตัวโปรแกรม

9. รายการเลือก (Menu) เป็นรายการแสดงกำหนดหน้าที่ต่างๆ ของโปรแกรมเพื่อให้ผู้ใช้สามารถเลือกการทำงานตามที่ผู้ใช้ต้องการได้โดยสะดวกขึ้น

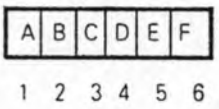
10. รายการความช่วยเหลือ (Help menu) เป็นรายการแสดงข้อความช่วยเหลือแก่ผู้ใช้เพื่อให้ผู้ใช้สามารถใช้โปรแกรมได้อย่างเต็มประสิทธิภาพมากขึ้น และรวดเร็วขึ้น โดยจะมีข้อความเกี่ยวกับการใช้แป้นอักขระต่างๆ ที่ถูกกำหนดไว้ในโปรแกรมแสดงออกมาให้ผู้ใช้ได้เห็นทางจอภาพ

11. รายการระบุความผิดพลาด (Error message) เป็นรายการที่จะเตือนผู้ใช้เมื่อมีการทำงานที่ผิดพลาดเกิดขึ้นในโปรแกรม เพื่อผู้ใช้จะได้ทำการแก้ไขได้ทันก่อนที่จะมีความเสียหายเกิดขึ้น

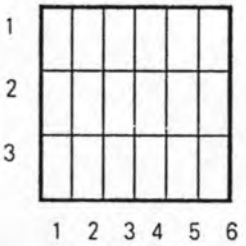
<u>โครงสร้างข้อมูลของโปรแกรมบรรณาธิกร</u> (Data structure of editing program) (18)

ในการทำงานของโปรแกรมบรรณาธิกร จำเป็นต้องใช้หน่วยความจำหลักที่มีอยู่ในเครื่องคอมพิวเตอร์ให้คุ้มค่ามากที่สุด ดังนั้นจึงจำเป็นต้องมีโครงสร้างของข้อมูลที่ดีเพื่อให้การบริหารการใช้หน่วยความจำทำได้อย่างมีประสิทธิภาพ โดยในการทำการวิจัยครั้งนี้ได้เลือกใช้โครงสร้างข้อมูลดังต่อไปนี้ คือ

1. โครงสร้างข้อมูลแบบแถวลำดับ (Array) (18) คือ การจัดข้อมูลเป็นแถวต่อเนื่องกันไปตามลำดับของข้อมูล ซึ่งอาจมี 1 มิติ 2 มิติ หรือหลายมิติ ดังรูปที่ 2.2 ในการทำวิจัยได้เลือกใช้โครงสร้างข้อมูลแบบแถวลำดับชนิด 1 มิติ และแบบ 2 มิติเท่านั้น โดยโครงสร้างข้อมูลแบบแถวลำดับชนิด 1 มิติ แสดงดังรูปที่ 2.2 ซึ่งเป็นโครงสร้างที่มีแถวลำดับอยู่จำนวน 6 ลำดับ และมีข้อมูลที่เก็บอยู่คือ "ABCDEF" ส่วนโครงสร้างข้อมูลแบบแถวลำดับชนิด 2 มิติ แสดงดังรูปที่ 2.3 ซึ่งเป็นโครงสร้างที่มี 2 มิติ โดยมิติแรกตามแนวแกนตั้งมีแถวลำดับอยู่ 3 ลำดับ และมีแถวลำดับตามแนวแกนนอนอยู่ 6 ลำดับ หรือเรียกว่าตารางชนิด 3x6 มิติ

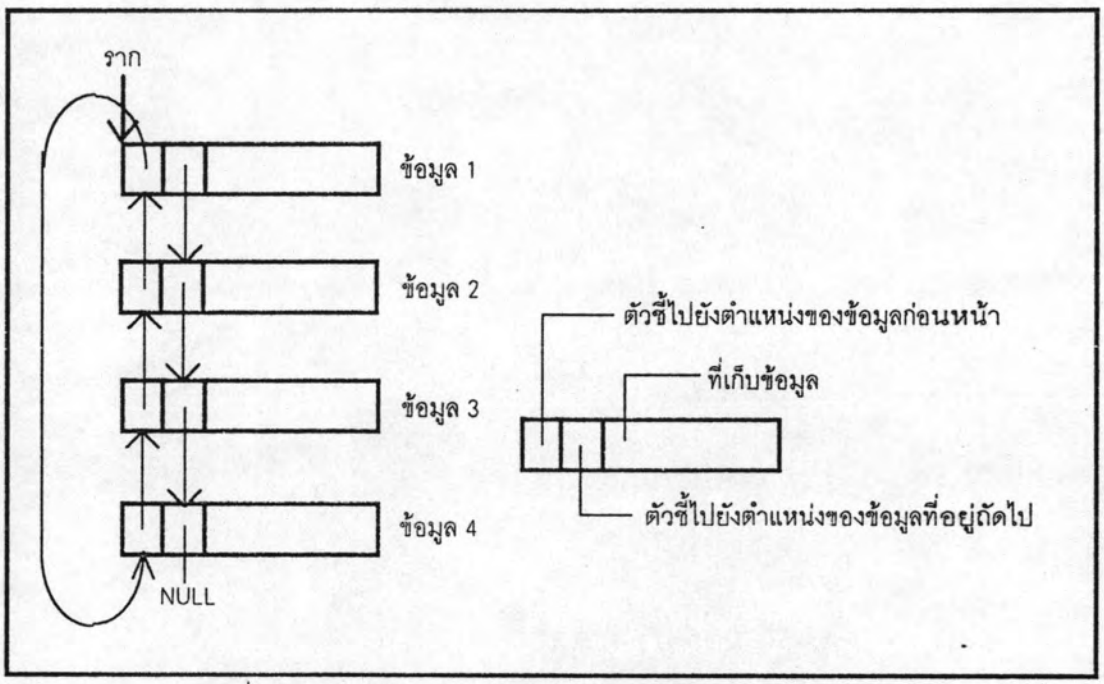


รูปที่ 2.2 แสดงโครงสร้างข้อมูลแบบแถวลำดับชนิด 1 มิติ



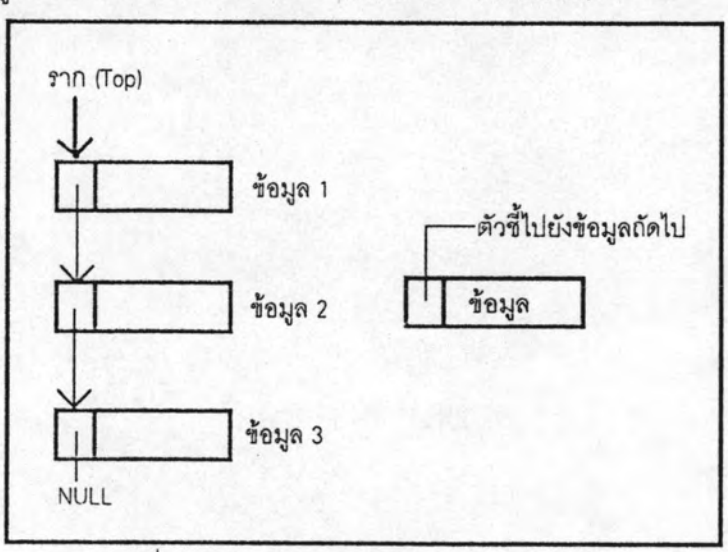
รูปที่ 2.3 แสดงโครงสร้างข้อมูลแบบแถวลำดับชนิด 2 มิติ

2. โครงสร้างข้อมูลแบบดับบลีลิงค์เซอร์คูล่าลิสต์ (Doubly link circular list) (19) เป็นการจัดข้อมูลที่มีความต่อเนื่องกัน โดยใช้ตัวชี้เป็นสื่อในการเชื่อมโยงข้อมูลเข้าด้วยกัน ตัวชี้นี้จะมีอยู่สองชุดด้วยกันคือ ชุดที่หนึ่งจะชี้ไปยังตำแหน่งข้อมูลที่อยู่ก่อนหน้า และชุดที่สองจะชี้ไปยังตำแหน่งข้อมูลที่อยู่ถัดไป ตัวชี้ของข้อมูลแรกและข้อมูลชุดสุดท้ายจะมีตัวชี้อย่างละหนึ่งชุด ที่ทำการชี้ไปยังส่วนหัวของข้อมูล โดยที่ส่วนหัวของข้อมูลก็จะมีตัวชี้สองทิศทางเช่นกัน โดยชุดที่หนึ่งชี้ไปยังตำแหน่งที่เก็บข้อมูลชุดแรก และชุดที่สองชี้ไปยังตำแหน่งที่เก็บข้อมูลชุดสุดท้าย ข้อมูลชุดสุดท้ายจะมีตัวชี้สองทิศทาง โดยตัวชี้ชุดที่หนึ่งชี้ไปยังข้อมูลที่อยู่ก่อนหน้า และตัวชี้ชุดที่สองจะว่าง (null) ดังนั้นจึงดูเหมือนกับว่า ข้อมูลทั้งหมดถูกเชื่อมกันเป็นวง รูปที่ 2.4 โครงสร้างข้อมูลชนิดนี้ได้นำมาใช้ในการเก็บข้อมูลของแต่ละบรรทัด



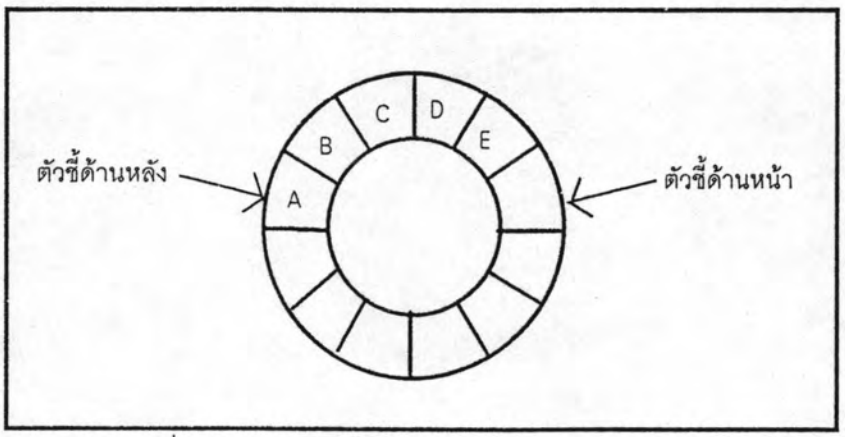
รูปที่ 2.4 แสดงโครงสร้างข้อมูลแบบดับบลีลิงค์เซอร์คูล่าลิสต์

3. โครงสร้างข้อมูลแบบกองซ้อน (Stack) (18) เป็นโครงสร้างของข้อมูลชนิดเรียงซ้อนกัน โดยข้อมูลแต่ละชุดจะมีตัวชี้อยู่ชุดละหนึ่งตัว ตัวชี้นี้จะชี้ไปยังข้อมูลที่อยู่ก่อนหน้า และตัวชี้ของข้อมูลชุดสุดท้ายจะว่าง ข้อมูลมีทางเข้าและทางออกอยู่ทางเดียวกัน โดยข้อมูลที่เข้ามาก่อนจะถูกดึงออกมาหลังสุด และข้อมูลที่เข้ามาหลังสุดจะถูกดึงออกมาก่อน (LIFO, Last In First Out) ดังรูปที่ 2.5 โครงสร้างข้อมูลชนิดนี้ นำมาใช้ในการทำการเรียกซ้ำ (recursion)



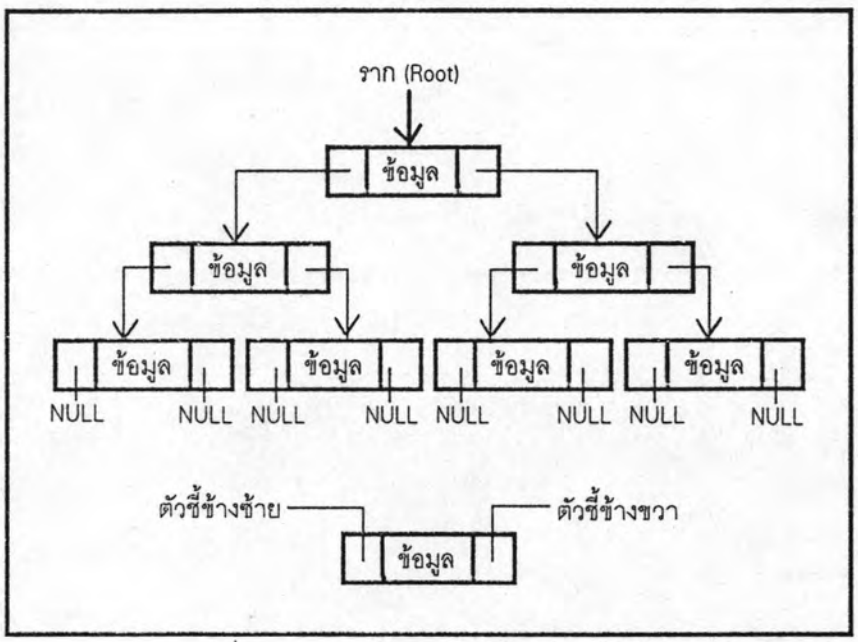
รูปที่ 2.5 แสดงโครงสร้างข้อมูลแบบกองซ้อน

4. โครงสร้างข้อมูลแถวคอยแบบวงกลม (Circular Queue) (19) เป็นโครงสร้างข้อมูลที่ตรงข้ามกับโครงสร้างแบบกองซ้อน คือเป็นแบบข้อมูลที่เข้าไปก่อนจะออกมาก่อน (FIFO, First In First Out) และโครงสร้างจะวนเป็นวงแหวน โดยจะมีตัวชี้อยู่ 2 ชุดด้วยกัน ตัวชี้ตัวแรกจะชี้ที่ส่วนหน้าของแถวคอยและตัวชี้ตัวที่สองจะชี้ที่ด้านหลังแถวคอย ดังรูปที่ 2.6 โครงสร้างชนิดนี้นำมาใช้ในการเก็บข้อมูลที่ถูกลบ โดยจะมีการแทนที่กันถ้าตัวชี้ทั้งสองตัวมาชี้ที่ตำแหน่งเดียวกัน



รูปที่ 2.6 แสดงโครงสร้างข้อมูลแถวคอยแบบวงกลม

5. โครงสร้างข้อมูลแบบต้นไม้ (Tree) (18) เป็นระบบโครงสร้าง ซึ่งแสดงความสัมพันธ์แบบลำดับชั้นของข้อมูล โดยข้อมูลแต่ละรายการจะมีการเชื่อมโยงไปยังข้อมูลอื่นในลำดับต่ำลงไปคล้ายๆ กิ่งก้านของต้นไม้ที่แตกออกจากลำต้น ดังรูปที่ 2.7



รูปที่ 2.7 แสดงโครงสร้างข้อมูลแบบต้นไม้

ขั้นตอนวิธีที่ใช้ในการพัฒนาโปรแกรม (Algorithm) (20)

1. การเรียงลำดับข้อมูล (Sorting)(21) หมายถึง การจัดเรียงลำดับข้อมูลตามตรรกใดตรรกหนึ่ง ตัวอย่างเช่น ในการทำวิจัยนี้ได้ทำการจัดเรียงข้อมูลตามลำดับตัวอักษรเป็นต้น สาเหตุที่ต้องเรียงข้อมูล ก็เพื่อให้ง่ายต่อการค้นหาและตีความ ซึ่งสามารถจะนำข้อมูลที่เรียงแล้วนั้นไปใช้ประโยชน์อย่างอื่นๆอีกได้ การเรียงข้อมูลสามารถแบ่งออกได้เป็น 2 ลักษณะใหญ่ๆ คือ

1.1 การเรียงลำดับข้อมูลภายใน (Internal sorting) จะใช้กับข้อมูลที่มีขนาดไม่ใหญ่กว่าพื้นที่หน่วยความจำหลักที่เหลืออยู่ โดยผู้เขียนโปรแกรมสามารถคำนวณ และจัดเรียงข้อมูลภายในหน่วยความจำหลักของเครื่องคอมพิวเตอร์ได้ โดยไม่ต้องใช้หน่วยความจำสำรองช่วย เช่น เทปหรือจานแม่เหล็ก เป็นต้น การจัดเรียงลำดับข้อมูลภายในโดยทั่วไปมีหลายวิธีด้วยกัน วิธีที่สำคัญมีดังนี้

1.1.1 แบบการเปรียบเทียบ (Sorting by comparison) แบ่งได้เป็น 2 ชนิดคือ

1.1.1.1 แบบการเปรียบเทียบปกติ วิธีการทำคือ การนำเอาค่าของข้อมูลแต่ละระเบียนมาทำการเปรียบเทียบกันทุกตัว ข้อมูลชุดใดที่มีค่าน้อยกว่าจะถูกสลับที่

1.1.1.2 แบบการเปรียบเทียบโดยการนับ เป็นการเรียงลำดับแบบการเปรียบเทียบผสมกับการนับ การทำคือจะหาว่าข้อมูลแต่ละชุด อยู่ในลำดับที่เท่าไหร่ของข้อมูลทั้งหมด ดังนั้นจึงไม่มีการสลับตำแหน่งของข้อมูลเกิดขึ้น

1.1.2 แบบการแทรก (Sorting by insertion) การเรียงลำดับแบบนี้จะถือเสมือนว่าข้อมูลแต่ละชุดมีการเรียงลำดับอยู่ก่อนแล้ว ดังนั้นการแทรกชุดของข้อมูล จะกระทำโดยการนำข้อมูลชุดที่จะแทรก ทำการแทรกเข้าไปในตำแหน่งที่ข้อมูลนั้นควรจะอยู่ โดยเพิ่มเข้าไปทีละชุดจนครบ

1.1.3 แบบเชลล์ (Shell's sort method) การเรียงลำดับแบบเชลล์นี้เป็นการปรับปรุงวิธีการเรียงลำดับแบบการแทรก แต่ใช้วิธีการเปรียบเทียบข้ามตำแหน่งโดยการแบ่งข้อมูลออกเป็นกลุ่มแล้วทำการเรียงลำดับข้อมูลในกลุ่มเหล่านี้ก่อน

1.1.4 แบบฟอง (Bubble sort method) เป็นวิธีการเรียงลำดับโดยการเปลี่ยนที่ระหว่างข้อมูลที่มีค่าคีย์น้อยกับข้อมูลที่มีค่าคีย์มากโดยการเปรียบเทียบทีละคู่ของข้อมูล ดังนั้นข้อมูลที่มีค่าคีย์มากสุดจะค่อยๆลอยขึ้นมาอยู่ข้างบน

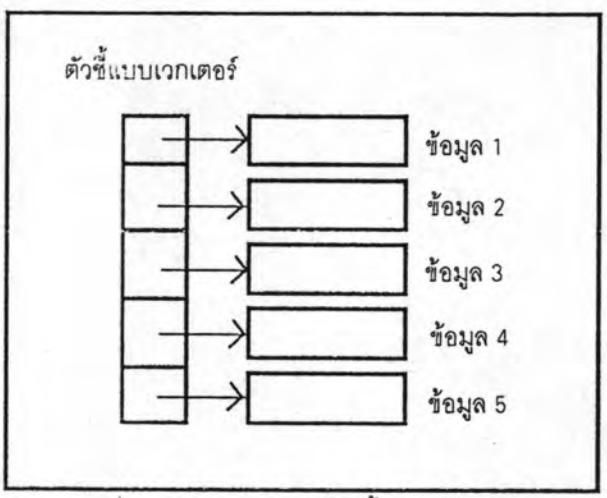
1.1.5 แบบควิกซอร์ต (Quick sort method) ปรับปรุงมาจากแบบฟอง แต่ใช้ผล จากการเปรียบเทียบในครั้งก่อน เพื่อตัดสินใจว่าจะเปรียบเทียบคีย์ใดต่อไป

1.1.6 แบบเรดิกซ์-เอกซ์เชนจ์ (Radix-exchange sort method) ทำโดยการเปลี่ยนค่าคีย์ทุกตัวให้เก็บอยู่ในรูปของเลขฐานสอง การเปรียบเทียบจะทำทีละบิต ว่าเป็น 0 หรือ 1 ซึ่งมีหลักการทำงานคล้ายกับแบบควิกซอร์ต

ในการทำวิจัยได้เลือกใช้วิธีการเรียงข้อมูลแบบฟอง เพราะวิธีการทำงานของการเรียงข้อมูลแบบนี้เหมาะสมกับโครงสร้างข้อมูลที่ใช้ในการพัฒนาโปรแกรม

1.2 การเรียงข้อมูลภายนอก (External sorting) จะใช้กับข้อมูลที่มีขนาดใหญ่เกินกว่าที่จะเก็บลงในหน่วยความจำได้หมดภายในครั้งเดียว และจะใช้หน่วยความจำภายนอกช่วย เช่น จานแม่เหล็กหรือเทปสำหรับเก็บข้อมูลชั่วคราว แล้วจึงค่อยจัดการเรียงข้อมูลภายในต่อไป

ในการทำวิจัยได้เลือกใช้การเรียงข้อมูลภายใน เพราะใช้กับข้อมูลที่มีขนาดไม่ใหญ่มาก ในการเรียงข้อมูลจำเป็นต้องทำตัวชี้ขึ้นมาใหม่ เป็นตัวชี้แบบเวกเตอร์ (Poiter vector) เพื่อจะได้ไม่ต้องทำการย้ายข้อมูลจริงในขณะเรียงข้อมูลอยู่ ตัวชี้แบบเวกเตอร์แสดงดังรูปที่ 2.8



รูปที่ 2.8 แสดงการใช้ตัวชี้แบบเวกเตอร์

2. การค้นหาข้อมูล (Searching) (18) หมายถึงการค้นหาข้อมูลที่มีลักษณะเหมือนคุณสมบัติที่กำหนดไว้ล่วงหน้า การค้นมีหลายแบบด้วยกัน เช่น

2.1 การค้นแบบทวิภาค (Binary search) วิธีการค้นหาโดยใช้ระบบแบ่งครึ่งข้อมูลทั้งหมดออกเป็นสองส่วนการค้นหาตัวที่ต้องการ จะต้องเลือกหาทีละส่วน โดยใช้วิธีเดิม คือแบ่งเป็นสองส่วนย่อย หากพบในส่วนหนึ่ง ก็ไม่ต้องเสียเวลาหาในอีกส่วนหนึ่ง

2.2 การค้นแบบครอบคลุม (Global search) เป็นการค้นหาทั่วทั้งหมดของข้อมูล ว่ามีข้อความหรือข้อมูลที่ต้องการหรือไม่

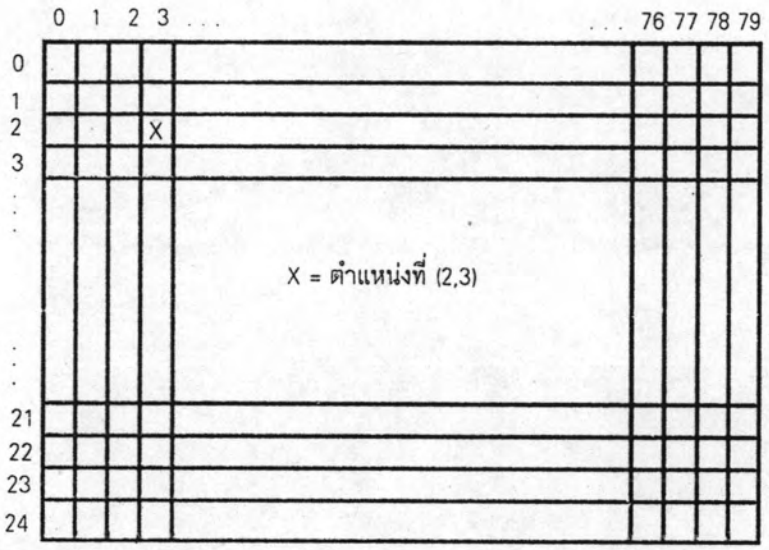
2.3 การค้นแบบลำดับ (Sequential search) เป็นการค้นหาโดยใช้วิธีการ เปรียบเทียบข้อความที่ต้องการกับข้อมูลที่เก็บอยู่ โดยเรียงไปทีละชุดของข้อมูล วิธีนี้ทำได้ง่ายที่สุด แต่จะช้าที่สุด

ทฤษฎีที่เกี่ยวข้องกับการแสดงผล (20)

1. การแสดงผลทางจอแสดงผล (22)

ในการแสดงผลทางจอแสดงผล สิ่งที่ต้องพิจารณาคือ ต้องรู้ว่าจอแสดงผลที่ใช้อยู่ในขณะนั้นเป็นจอแสดงผลชนิดใด ซึ่งจอแสดงผลที่ใช้กันอยู่ในปัจจุบัน สามารถแบ่งใหญ่ๆได้เป็น 2 ประเภทด้วยกัน คือ

1.1 จอแสดงผลสีเดียว (Monochrome monitor) เป็นการแสดงผลซึ่งให้ภาพขาวดำ ของตัวอักษรเลข (Alphanumeric) และต้องใช้ร่วมกับตัวเชื่อมต่อสำหรับจอภาพสีเดียว (MDA, Monochrome display adapter) โดยจะทำให้สามารถแสดงผลได้ 25 บรรทัด บรรทัดละ 80 ตัวอักษร ดังนั้นจึงสามารถแสดงได้ถึง 2000 ตัวอักษรพร้อมกัน ซึ่งลักษณะการวางตำแหน่ง ของจอภาพแสดงได้ดังรูปที่ 2.9 ตัวเชื่อมต่อจะทำให้เกิดตัวอักษร โดยการแปลรหัสแอสกี (ASCII) แต่ละตัวให้เป็นตัวอักษรที่สามารถแสดงบนจอแสดงผลได้ รหัสแอสกีมีทั้งหมด 256 ตัว



รูปที่ 2.9 แสดงลักษณะการวางตำแหน่งของจอแสดงผลสีเดียว

นอกจากรหัสแอสกี ซึ่งแสดงตัวอักษรหรือรูปภาพแล้ว ตัวอักษรหรือรูปภาพทุกตำแหน่งบนจอภาพต้องมีลักษณะเฉพาะในการแสดงผลด้วย (Display attribute) ซึ่งลักษณะเฉพาะนี้จะเป็นตัวชี้บอกว่าจะแสดงตัวอักษรบนจอแสดงผลอย่างไร เช่น แสดงตัวอักษรสีขาวบนพื้นสีดำ ที่เรียกว่านอร์มอลวีดีโอ หรือแสดงตัวอักษรสีดำบนพื้นสีขาวที่เรียกว่า รีเวอร์สวีดีโอ หรือตัวอักษรอาจจะกระพริบ อย่างเช่น เคอร์เซอร์ หรือมีความเข้มสูงกว่าตัวอักษรอื่น หรืออาจมีการขีดเส้นใต้ใต้ตัวอักษรนั้น ลักษณะเฉพาะนี้มีความยาวหนึ่งไบต์ ดังรูปที่ 2.10

รูปที่ 2.10 แสดงค่าลักษณะเฉพาะในการแสดงผล

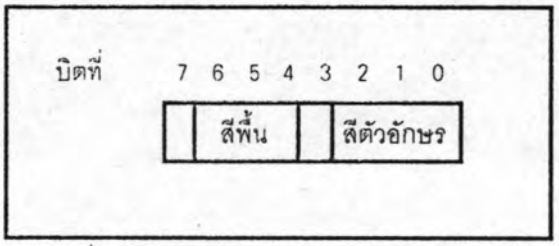
การแสดงผลบนจอแสดงผลสีเดียวทำได้ 2 วิธี คือ

1.1.1 การใส่รหัสตัวอักษรและค่าลักษณะเฉพาะในการแสดงผลลงไปบนหน่วยความจำหลักของการแสดงผลโดยตรง หน่วยความจำในส่วนนี้ มีขนาด 4 กิโลไบต์ โดยเริ่มที่ตำแหน่ง 0B0000H ถ้าใส่ค่าของเซกเมนต์รีจิสเตอร์เป็น 0B0000H ค่าออฟเซตก็ต้องมีค่าตั้งแต่ 0 ถึง 4 กิโลไบต์ ซึ่งตำแหน่งของหน่วยความจำที่เป็นเลขคู่ จะแสดงรหัสของตัวอักษร และตำแหน่งของหน่วยความจำที่เป็นเลขคี่จะแสดงค่าของลักษณะเฉพาะในการแสดงผล

1.1.2 การแสดงผลทางจอแสดงผล โดยการเรียกใช้ไบออส (BIOS) เป็นการขัดจังหวะเลขที่ 10H ซึ่งมีอยู่ด้วยกันจำนวน 16 ชุด ในการทำวิจัยไม่ได้ใช้ส่วนนี้ จึงจะไม่ขอกล่าวถึง

1.2 <u>จอสี</u> (Color / Graphic monitor adapter) การแสดงภาพบนจอสี ต้องใช้ตัวเชื่อมต่อสำหรับจอสี ซึ่งเป็นอุปกรณ์เชื่อมต่อที่สามารถแสดงตัวอักษร ที่เรียกว่าภาวะข้อความ

(Text mode) และแสดงรูปภาพ ที่เรียกว่าภาวะกราฟิก (Graphic mode) ตัวเชื่อมต่อนี้มีหน่วยความจำอยู่ 16 กิโลไบต์ โดยเริ่มที่ตำแหน่ง 0B8000H ในการใช้ภาวะตัวอักษรสามารถสร้างภาพได้สองแบบคือ แบบ 25 บรรทัด บรรทัดละ 80 ตัวอักษร ซึ่งคล้ายกับจอแสดงผลสีเดียว และแบบ 25 บรรทัด บรรทัดละ 40 ตัวอักษร ในภาวะกราฟิกจะไม่ขอกล่าวถึงเพราะไม่ได้ใช้ในการทำวิจัยครั้งนี้ การแสดงผลทั้งสองแบบ สามารถกำหนดสีของตัวอักษรแต่ละตัวได้ โดยที่แต่ละตัวอักษรจะมีลักษณะเฉพาะ มีขนาด 1 ไบต์เช่นเดียวกับของจอแสดงผลสีเดียว ลักษณะเฉพาะนี้จะเป็นตัวกำหนดสีพื้น (Background) และสีของตัวอักษร (Foreground) แต่ละตัว ดังรูปที่ 2.11



รูปที่ 2.11 แสดงลักษณะเฉพาะของจอสี

ลักษณะเฉพาะของจอสี

สีพื้น มีขนาด 3 บิต กำหนดสีพื้น

สีตัวอักษร มีขนาด 3 บิต กำหนดสีตัวอักษร

สภาพสี ขนาด 1 บิต ตำแหน่งบิตที่ 3

ค่าเป็น 1 สีอ่อน

ค่าเป็น 0 สีเข้ม

การกระพริบ ขนาด 1 บิต ตำแหน่งบิตที่ 7

ค่าเป็น 1 ตัวอักษรจะกระพริบ

ค่าเป็น 0 ตัวอักษรไม่กระพริบ

ตารางที่ 2.1 แสดงสีทั้งหมดที่สามารถแสดงได้ของจอสี ในการแสดงผลก็ทำเช่นเดียวกับจอแสดงผลสีเดียว คือการอ้างถึงตำแหน่งในหน่วยความจำโดยตรง ซึ่งจะมีผลทำให้การทำงานทำได้เร็ว

I	R	G	B	สี
0	0	0	0	ดำ
0	0	0	1	น้ำเงิน
0	0	1	0	เขียว
0	0	1	1	คราม
0	1	0	0	แดง
0	1	0	1	ม่วง
0	1	1	0	น้ำตาล
0	1	1	1	เทาเข้ม
1	0	0	0	เทาอ่อน
1	0	0	1	ฟ้าอ่อน
1	0	1	0	เขียวอ่อน
1	0	1	1	ครามอ่อน
1	1	0	0	แดงอ่อน
1	1	0	1	ม่วงอ่อน
1	1	1	0	เหลือง
1	1	1	1	ขาว

ตารางที่ 2.1 แสดงรหัสสีของจอสี

2. <u>การแสดงผลทางเครื่องพิมพ์</u> (22)

การแสดงผลทางเครื่องพิมพ์สามารถทำได้โดยการใช้การขัดจังหวะที่ 17H ซึ่งมีชุดคำสั่งอยู่ 3 ชุดคำสั่งด้วยกัน คือ

2.1 ชุดคำสั่งที่ 1 ถ้า AH มีค่าเป็น 0 จะเกิดการพิมพ์ค่า โดยใส่รหัสตัวอักษรที่ต้องการพิมพ์ ในรีจิสเตอร์ AL และรีจิสเตอร์ DX มีค่า 0

2.2 ชุดคำสั่งที่ 2 ถ้า AH มีค่าเป็น 1 และ DX มีค่าเป็น 0 จะเป็นการเริ่มต้นเครื่องพิมพ์ใหม่

2.3 ชุดคำสั่งที่ 3 ถ้า AH มีค่าเป็น 2 จะเป็นการอ่านค่าสภาวะของเครื่องพิมพ์มาใส่ในรีจิสเตอร์ AH

3. การรับข้อมูลทางแป้นพิมพ์ (Keyboard) (22)

แป้นพิมพ์สำหรับคอมพิวเตอร์ ถูกจัดวางเป็น 3 กลุ่มด้วยกัน ตรงกลางจะมีลักษณะเหมือนแป้นพิมพ์ดีดธรรมดา ทางซ้ายมือจะเป็นแป้นฟังก์ชัน ซึ่งสามารถทำโปรแกรมควบคุมการทำงานได้ ทางขวามือเป็นแป้นตัวเลขและลูกศรขึ้นลงซ้ายขวา แป้นพิมพ์จะต่อกับเครื่องไมโครคอมพิวเตอร์โดยตัวแปรรหัสแอสกี 8 บิต สำหรับเครื่องไอบีเอ็มเมื่อกดแป้นจะเกิดการสร้างค่าสแกนโค๊ด (Scan code) ซึ่งไบออสจะแปลรหัสที่เรียกว่าเอ็กเทนแอสกี (Extended ASCII) ซึ่งเป็นรหัสแอสกีหนึ่งไบต์บวกกับรหัสสำหรับสัญลักษณ์ที่ไม่ใช่แอสกีและฟังก์ชันเฉพาะของแป้น หรือที่ใช้กับการขัดจังหวะไบออส สำหรับแป้นพิมพ์จะมีหน่วยความจำ ที่มีขนาดใหญ่พอสำหรับคนที่พิมพ์ได้เร็ว ในการใช้งานแป้นพิมพ์จะใช้การขัดจังหวะที่ 16H ซึ่งมีชุดคำสั่งอยู่ 3 อย่างคือ

3.1 ชุดคำสั่งที่ 1 ค่า AH มีค่า 0 จะเกิดการอ่านค่าของสแกนโค๊ดของแป้นพิมพ์ในหน่วยความจำไปใส่ในรีจิสเตอร์ AH และค่ารหัสตัวอักษรใส่ในรีจิสเตอร์ AL และเลื่อนตัวชี้ของหน่วยความจำไปชี้ที่ตัวอักษรถัดไป ถ้าในหน่วยความจำไม่มีอะไรอยู่เลย ก็จะรอจนมีการกดแป้นพิมพ์

3.2 ชุดคำสั่งที่ 2 ค่า AH มีค่า 1 จะเกิดการอ่านค่าของสภาวะของหน่วยความจำใส่ในแฟกซ์ (ZF) ถ้าหน่วยความจำไม่มีอะไรเลย จะมีค่าเป็น 0 และ ถ้ามีข้อมูลก็จะมีค่าเป็น 1

3.3 ชุดคำสั่งที่ 3 ค่า AH มีค่า 2 จะเกิดการอ่านค่าของแป้นพิมพ์ ใส่ในรีจิสเตอร์ AL โดยที่บิต 4 ถึงบิต 7 บอกว่าภาวะของแป้นเปิดอยู่ (ON) ซึ่งมีค่า 1 หรือปิดอยู่ (OFF) มีค่า เป็น 0 และบิต 0 ถึงบิต 3 บอกว่ามีการกดแป้น Alt หรือ Ctrl หรือ Shift ร่วมด้วยหรือไม่ ค่าสภาวะไบต์ของแป้นพิมพ์มีดังนี้

บิตที่ 0 เกิดการกดแป้น Shift ทางขวามือ
บิตที่ 1 เกิดการกดแป้น Shift ทางซ้ายมือ
บิตที่ 2 เกิดการกดแป้น Ctrl
บิตที่ 3 เกิดการกดแป้น Alt
บิตที่ 4 SCROLL LOCK ON
บิตที่ 5 NUM LOCK ON
บิตที่ 6 CAPS LOCK ON
บิตที่ 7 INSERT MODE ON

การพัฒนาโปรแกรมโดยใช้โอโอพี (OOP, Object-Oriented Programming) (23)

การเขียนโปรแกรมแบบโอโอพี คือ การเขียนโปรแกรมโดยมุ่งที่วัตถุเป็นหลัก ซึ่งเป็นวิธีหนึ่งที่ทำให้สามารถนำโปรแกรมเก่ากลับมาใช้ได้อีก และทำให้ประหยัดเวลาในการพัฒนาโปรแกรม ในการทำวิจัยครั้งนี้ได้ใช้เครื่องมือในการพัฒนาโปรแกรม โดยเลือกใช้ภาษาซีพลัสพลัส (C++) ซึ่งต้องใช้หลักการเขียนโปรแกรมแบบโอโอพีด้วย โดยหลักการของโอโอพีมีดังนี้คือ

1. เอนแคปซูเลชัน (Encapsulation) คือ ความเป็นเอกเทศ หมายถึงให้สามารถรวมรายละเอียดกับกิจกรรมไว้ในที่แห่งเดียวกัน และให้อยู่ในที่ปกป้องอันหนึ่ง เพื่อไม่ให้โปรแกรมส่วนอื่นเข้าถึงรายละเอียดหรือกิจกรรมได้โดยตรง อย่างน้อยที่สุดให้ได้รับการปกป้องในระดับระเบียนคือต้องดำเนินการผ่านอินสแตนซ์ของสกุล การเป็นเอกเทศมีจุดมุ่งหมายหลายประการคือ ทางด้านโอโอพีถือว่าเป็นการสร้างแบบจำลองของวัตถุ ทางด้านภาษาก็คล้ายกันคือ มองในแง่ความสมบูรณ์ของโปรแกรมที่ไม่ต้องกำหนดตัวแปร หรือเขียนโปรแกรมเพิ่มอีกเลยสำหรับวัตถุนั้น เป็นการทำงานเพียงครั้งเดียวที่ใช้ได้ตลอดไป อีกประเด็นหนึ่งคือ ตัวแปรที่ใช้เก็บรายละเอียดหากไม่ทำเป็นสกุล คือไม่มีความเป็นเอกเทศก็จะต้องกำหนดเป็นแบบครอบคลุม (Global) ซึ่งผิดหลักการเขียนโปรแกรมโดยทั่วไป เมื่อเป็นเอกเทศแล้วก็จะเป็นแบบครอบคลุมอยู่ภายในสกุลนั้น ทำให้ได้ใช้ข้อดีของตัวแปรแบบครอบคลุมและไม่เสียเวลาในการทำงาน

2. อินเฮอริแตนซ์ (Inheritance) คือ การสืบสกุล หมายถึง เมื่อสร้างสกุลหนึ่งแล้วทำให้สามารถสร้างสกุลลูก หลานและเหลนต่อไปได้ เรียกว่า ผู้สืบสกุล (Descendant) ของสกุลแรกและเรียกสกุลแรกว่า บรรพบุรุษ (Ancestor) การสืบสกุลหมายถึงการสืบมรดกด้วยคือ ผู้สืบสกุลสามารถใช้ตัวแปรและกิจกรรมในสกุลบรรพบุรุษได้เช่นเดียวกับที่เป็นของตนเอง แต่ในทางตรงกันข้าม บรรพบุรุษจะใช้ตัวแปรและกิจกรรมของผู้สืบสกุลไม่ได้ ความมุ่งหมายของการสืบสกุลคือเพื่อให้ปรับปรุง แก้ไขและต่อเติมโปรแกรมได้ โดยไม่ต้องกระทำในโปรแกรมต้นฉบับเดิม ในภาษาซีพลัสพลัส ผู้เขียนโปรแกรมบรรพบุรุษจะเป็นผู้กำหนดว่าจะปกป้องตัวแปร และกิจกรรมใดบ้าง โดยในแต่ละวัตถุ จะเรียกว่าคลาส (Class) ซึ่งภายในแบ่งออกเป็น 3 ส่วนด้วยกัน คือ

2.1 ส่วนตัว (Private) เป็นส่วนที่ใช้เก็บตัวแปรต่างๆ ที่ใช้ในการทำงานภายในวัตถุ โดยสมาชิกของคลาสอื่นๆจะไม่สามารถเข้ามาใช้ได้

2.2 ส่วนปกป้อง (Protected) เป็นส่วนที่อาจมีทั้งตัวแปรและการทำงานร่วมกัน ในส่วนปกป้องนี้จะเป็นส่วนที่ป้องกันไว้สำหรับคลาสที่มีการสืบทอดกันแบบโพลิมอร์ฟิซึม ซึ่งจะทำให้คลาสที่มาติดต่อด้วยนั้น สามารถเข้าใช้ได้ทั้งตัวแปรและการทำงานในคลาสนี้ และสมาชิกในคลาสเดียวกันก็สามารถใช้ความสามารถนี้ได้

2.3 ส่วนเผยแพร่ (Public) เป็นส่วนที่อาจเก็บทั้งตัวแปรและการทำงานร่วมกัน และเผยแพร่ให้ระบบทั้งหมดสามารถมาเรียกใช้ได้ ถ้ามีการประกาศไว้ก่อนแล้ว

3. โพลิมอร์ฟิซึม (Polymorphism) หมายถึง การเลือกปฏิบัติโดยกำหนดกิจกรรมในสกุลใหม่มีชื่อซ้ำกับสกุลบรรพบุรุษ ซึ่งทางภาษาให้กำหนดเป็นกิจกรรมที่เป็นเวอร์ชวล (Virtual)

4. โอเวอร์โหลด (Overload) หมายถึง ความสามารถในการใช้ชื่อกิจกรรมเดียวกันให้ทำงาน แตกต่างกันได้ในสกุลเดียวกัน