บทที่ 2

ทฤษฎีและแนวความคิดที่นำมาใช้ในการวิจัย

1. โปรแกรมบรรณาธิกร (Editor Program)

ลักษณะของโปรแกรมบรรณาธิกร (พันธุ์ปิติ เปี่ยมสง่า,2534) มีอยู่ 2 ลักษณะคือ โปรแกรมบรรณาธิกรบรรทัด(Line Editor) และ โปรแกรมบรรณาธิกรจอภาพ (Screen Editor) โดย

1.1 โปรแกรมบรรณาธิกรบรรทัด

เป็นบรรณาธิกรที่ผู้ใช้สามารถทำการแก้ไข ปรับปรุง เปลี่ยนแปลง หรือเรียบเรียงข้อความได้ทีละบรรทัด และเมื่อทำการแก้ไขปรับปรุงเสร็จจึงจะสามารถเลื่อนไปยังบรรทัดอื่นในโปรแกรมได้ คำสั่งควบคุมที่มีให้ใช้นั้นจะมีคำสั่งให้เลือกใช้ได้ไม่กี่คำสั่ง โดยมีเฉพาะคำสั่งควบคุมที่จำเป็นต้องใช้จริง ๆ เท่านั้น ดังนั้นจึงทำให้การใช้งานทำได้ไม่สะดวกขาดความคล่องตัว และยังประสบกับปัญหาในการเลื่อนบรรทัดเพื่อจะแก้ไขข้อความที่กำลังทำงานอยู่ เพราะสามารถทำการแก้ไขข้อความได้ครั้งละบรรทัดเท่านั้น เนื่องจากว่าโปรแกรมบรรณาธิกรบรรทัดมักจะถูกพัฒนาขึ้นในคอมพิวเตอร์ยุคแรก ๆ ซึ่งในยุคนั้นจะเป็นต้องใช้ทรัพยากรต่าง ๆ อย่างคุ้มค่าที่สุด เพราะอุปกรณ์คอมพิวเตอร์มีราคาแพงหาได้ยาก และมีอยู่น้อย ดังนั้นโปรแกรมจึงถูกพัฒนาให้พอใช้งานได้โดยใช้หน่วยความจำน้อยที่สุด มาจนถึงในยุคปัจจุบันโปรแกรมบรรณาธิกรลักษณะนี้ก็ยังคงมีใช้อยู่ เช่น โปรแกรมบรรณาธิกรที่ชื่อว่า เอ็ดลิน ซึ่งใช้ในระบบปฏิบัติการดอส แต่ก็มีผู้นิยมใช้น้อย จะใช้ก็ต่อเมื่อจำเป็น หรือไม่มีโปรแกรมบรรณาธิกรชนิดอื่นให้ใช้เท่านั้น

1.2 โปรแกรมบรรณาธิกรจอภาพ (Screen Editor)

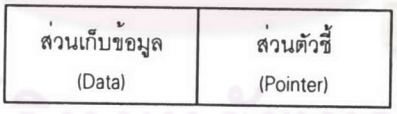
เป็นบรรณาธิกรที่ผู้ใช้สามารถทำการแก้ไข เปลี่ยนแปลง หรือเรียบเรียงข้อมูล ณ ตำแหน่งใด ๆ ในจอภาพได้ โดยสามารถแสดงข้อความได้เต็มจอภาพ ดังนั้น ในการใช้งานโดยทั่ว ๆ ไป โปรแกรมบรรณาธิกรจอภาพจึงมักมีผู้นิยมใช้มากกว่าโปรแกรมบรรณาธิกรบรรทัด เนื่องจากใช้งานได้สะดวก มีความคล่องตัวในการใช้งาน มีคำสั่งต่าง ๆ ให้เลือกใช้ได้มากกว่า ในปัจจุบันมีผู้ทำการพัฒนาโปรแกรมบรรณาธิกรลักษณะนี้กันมาก เช่น โปรแกรมบรรณาธิกรชื่ออีดิต ซึ่งใช้บนระบบปฏิบัติการดอส โปรแกรมบรรณาธิกรชื่อ วีไอ (Vi) ใช้บนระบบปฏิบัติการยูนิกซ์ เป็นต้น

ในการทำวิจัยครั้งนี้ได้ทำการพัฒนาเครื่องมือในการพิมพ์โปรแกรมภาษาต้นฉบับภาษาซี และภาษาปาสคาล ซึ่งเป็นโปรแกรมบรรณาธิกรชนิดจอภาพ ที่สามารถแสดงสีในคำสงวน ขยายรูปแบบคำสั่งบางคำสั่ง และสามารถส่งโปรแกรมไปทำการแปลโปรแกรม (Compile) ได้

2. โครงสร้างข้อมูลแบบรายการโยง (Linked Lists) (ดร.สุชาย ธนวเสถียร และ วิชัย จิวังกูร, 2521)

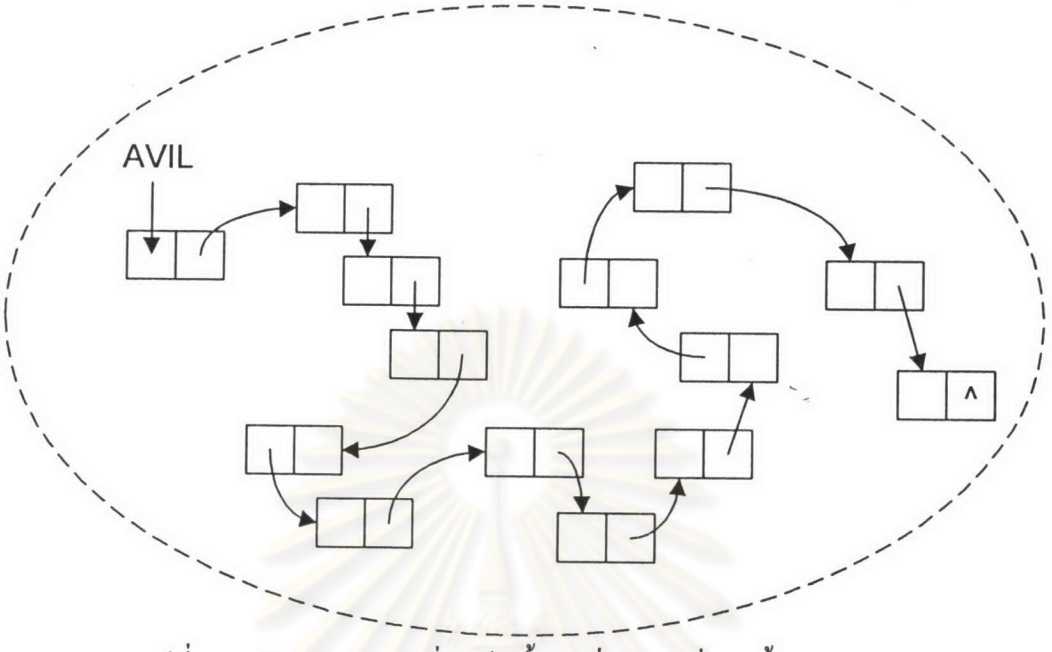
โครงสร้างแบบนี้จะให้ความสะดวกแก่การจัดการข้อมูลอย่างมาก แต่จะซับซ้อนกว่าแบบใช้แถวลำดับ (Array) ข้อจำกัดของการใช้แถวลำดับคือ แถวลำดับจะมีขนาดคงที่ เช่นถ้าใช้แถวลำดับขนาด 100 ช่อง เมื่อใช้หมด 100 ช่องก็จะใช้แถวลำดับนั้นไม่ได้อีกถ้าไม่อนุญาตให้ใช้พื้นที่นั้นซ้ำ แต่ถึงแม้ว่าจะสามารถใช้พื้นที่นั้นซ้ำได้ ก็ต้องมีการเลื่อนข้อมูลทั้งหมดไปอยู่ในส่วนปลาย หรือส่วนต้นของแถวลำดับให้หมด ซึ่งการทำเช่นนี้จะเสียเวลามาก ข้อจำกัดอีกประการคือในภาวะการทำงานแบบทันที (Real Time Processing) ที่ต้องมีการนำข้อมูลเข้า (Insertion) หรือดึงข้อมูลออก (Deletion) แถวลำดับจะไม่สะดวกแก่การใช้มาก ในภาวะการทำงานแบบนี้โครงสร้างแบบรายการโยงจะสะดวกกว่า

ลักษณะโครงสร้างข้อมูลแบบนี้ แต่ละโหนดจะประกอบด้วย 2 ส่วน คือ ส่วนที่เก็บข้อมูล (Data) และส่วนตัวชี้ (Pointer) เพื่อแสดงให้เห็นชัดว่าโหนดที่ต่อจากโหนดนั้นคือโหนดใด ดังรูปที่ 2.1



รูปที่ 2.1 ลักษณะโหนดของรายการโยง

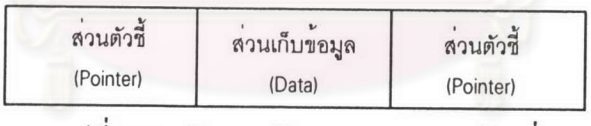
ในระบบการแทนโดยรายการโยงเราจะสมมติว่าหน่วยความจำจะประกอบด้วยชิ้นส่วนความจำ เป็นชิ้น ๆ ซึ่งจะเรียกว่าโหนด ชิ้นส่วนความจำทั้งหมดนี้จะคล้องกันเป็นรายการโยงเดี่ยวขนาดมหึมา ซึ่งที่รวมความจำนี้เรียกว่า หน่วยเก็บข้อมูลส่วนรวม เมื่อผู้ใช้ต้องการพื้นที่เก็บข้อมูลก็เรียกเอาโหนดจากหน่วยเก็บข้อมูลส่วนรวม ถ้าผู้ใช้ไม่ต้องการใช้ โหนดใดก็คืนโหนดนั้นไปยังหน่วยเก็บข้อมูลส่วนรวมเพื่อผู้อื่นจะได้ใช้ โหนดแรกของหน่วยเก็บข้อมูลส่วนรวมถูกชี้ด้วยตัวชี้ชื่อ AVIL ดังรูปที่ 2.2



รูปที่ 2.2 ลักษณะของ หน่วยเก็บข้อมูลส่วนรวมซึ่งถูกชี้โดย AVIL

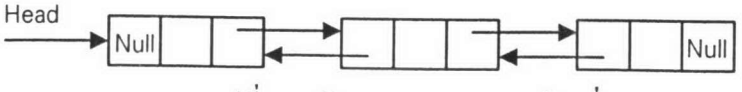
2.1 โครงสร้างข้อมูลแบบรายการโยงคู่ (Doubly Linked Lists) (ดร.สุชาย ธนวเสถียร และ วิชัย จิวังกูร,2521)

รายการโยงแบบนี้เป็นรายการโยงที่แต่ละโหนดมีสองตัวชี้ โดยตัวชี้ตัวหนึ่งจะชี้ไปยังโหนดที่อยู่ก่อนหน้า อีกตัวหนึ่งจะชี้ไปยังโหนดถัดไป ดังรูปที่ 2.3



รูปที่ 2.3 ลักษณะโหนดของรายการโยงคู่

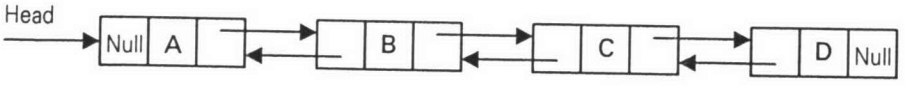
รายการโยงแบบนี้ในโหนดแรกจะมีตัวชี้ที่จะแสดงว่าเป็นส่วนหัวของข้อมูลซึ่งคือ Head และข้อมูลที่อยู่ก่อนหน้าไม่มีดังนั้น ตัวชี้ตัวแรกจะชี้ไปที่ว่าง (Null) และในโหนดสุดท้ายจะมีตัวชี้สองทิศทาง โดยตัวชี้ตัวหนึ่งชี้ไปยังข้อมูลที่อยู่ก่อนหน้า และตัวชี้อีกตัวหนึ่งจะชี้ไปที่ว่าง (Null) ซึ่งหมายถึงว่าสิ้นสุดของค่าชุดนี้แล้ว ดังรูปที่ 2.4



รูปที่ 2.4 ลักษณะของรายการโยงคู่

โครงสร้างข้อมูลชนิดนี้ได้นำมาใช้ในการเก็บข้อมูลของแต่ละบรรทัด

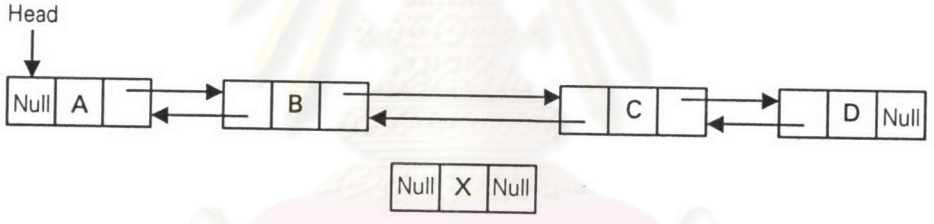
ตัวอย่าง ถ้าต้องการจะเก็บค่า A,B,C และ D ในรายการโยงชนิดนี้จะได้โครงสร้างดังรูป 2.5



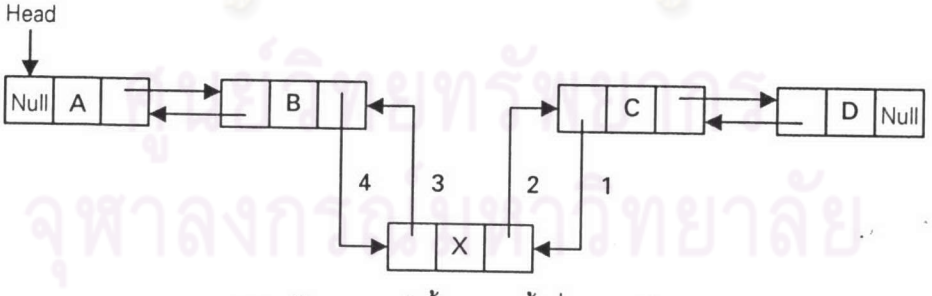
รูปที่ 2.5 ตัวอย่างรายการโยงคู่

2.1.1 การนำข้อมูลเข้าสู่รายการโยงคู่ (Insertion)

สมมติว่ามีรายการโยงอยู่แล้ว และจะนำโหนดซึ่งเก็บข้อมูล X เข้าสู่รายการโยงนี้ ที่ตำแหน่งถัดจากโหนดที่เก็บข้อมูล B ซึ่งกระบวนการนี้สำเร็จได้โดยการเปลี่ยนค่าตัวชี้สี่ค่า โดยทำตามลำดับ 4,3,2 และ 1 ดังรูปที่ 2.6 โดยวิธีการนี้ได้นำมาใช้ในกรณีที่มีการเพิ่มบรรทัดของข้อมูล



(ก) ก่อนนำข้อมูล X เข้าสู่รายการโยง



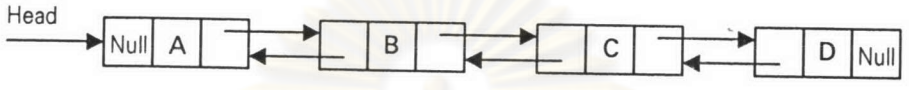
(ข) หลังจากการนำข้อมูล X เข้าสู่รายการโยง

รูปที่ 2.6 แสดงการนำโหนดที่มีข้อความ X เข้าสู่รายการโยงคู่

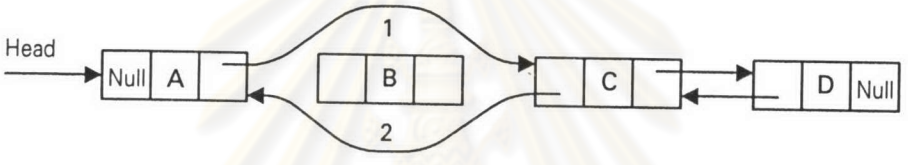
2.1.2 การนำข้อมูลออกจากรายการโยงคู่ (Deletion)

สมมติว่ามีรายการโยงอยู่แล้ว และต้องการนำโหนดที่เก็บข้อมูล B ออกจากรายการโยง สามารถกระทำได้ด้วยการเปลี่ยนค่าตัวชี้สองค่า โดยทำตามลำดับ 1,2 ดังแสดงในรูปที่ 2.7

วิธีการนี้ได้นำมาใช้ในกรณีที่มีการลบข้อมูลบรรทัดทิ้ง



(ก) ก่อนการนำข้อมูล B ออก

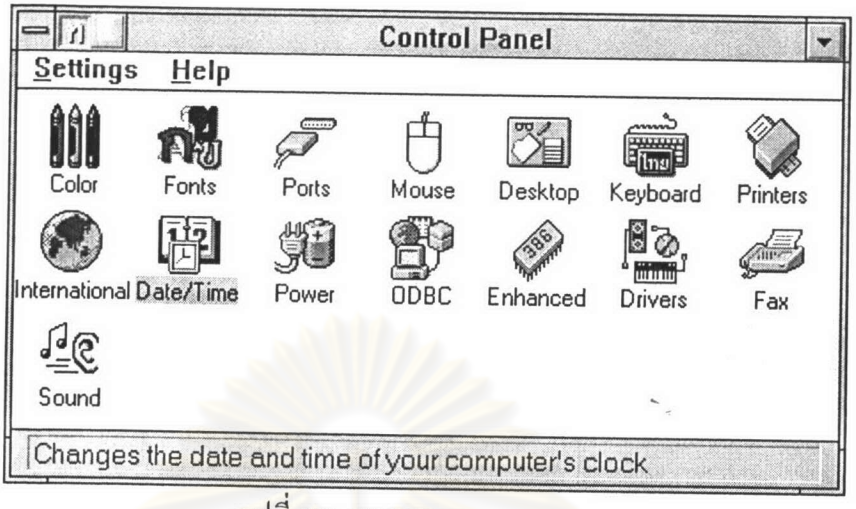


(ข) หลังการนำข้อมูล B ออก

รูปที่ 2.7 การนำโหนดออกจากรายการโยงคู่

3. ระบบวินโดวส์

โปรแกรมบนวินโดวส์เป็นโปรแกรมในรูปแบบที่เรียกว่า "การดำเนินการกับสภาพแวดล้อม" (จิรพัฒน์ จันทร์เจิดศักดิ์ และ วีระ นพนิราพาธ,2521) กล่าวคือ โปรแกรมจะจัดการสภาวะระบบให้ใหม่จากเดิมซึ่งจัดการโดยดอส (DOS) ให้กลายเป็นสภาวะใหม่ที่มีการทำงานแตกต่างออกไปโดยสิ้นเชิง เช่น มีการทำงานแบบกราฟิก มีการจัดบริหารหน่วยความจำใหม่ เป็นต้น นอกจากนี้เราอาจจะเรียกว่าเป็นโปรแกรมประเภทประสานกับผู้ใช้แบบกราฟิก (Graphic User Interface Program) หรือ กูอีโปรแกรม (GUI Program) นั่นคือผู้ใช้จะทำการติดต่อกับระบบคอมพิวเตอร์โดยใช้รูปภาพ หรือสัญญลักษณ์ต่าง ๆ ที่แสดงทางจอภาพแทนการใช้ข้อความหรือการใช้คำสั่งหลาย ๆ ขั้นตอน ผู้ใช้สามารถที่จะสั่งงานได้โดยใช้รูปภาพเหล่านั้น ซึ่งวิธีการนี้ได้มีการพิสูจน์แล้วว่า ผู้ใช้จะติดต่อกับคอมพิวเตอร์ได้ง่ายและสะดวกกว่าวิธีการใช้ข้อความ อุปกรณ์รับข้อมูลที่ใช้ในการติดต่อกับคอมพิวเตอร์ในระบบวินโดวส์นั้น จะใช้ได้ทั้งแป้นอักษรและเมาส์ ดังรูปที่ 2.8 ซึ่งโปรแกรมระบบวินโดวส์นี้จะทำการติดต่อกับอุปกรณ์ ต่าง ๆ เหล่านี้ ในรูปของข้อความ (Message)



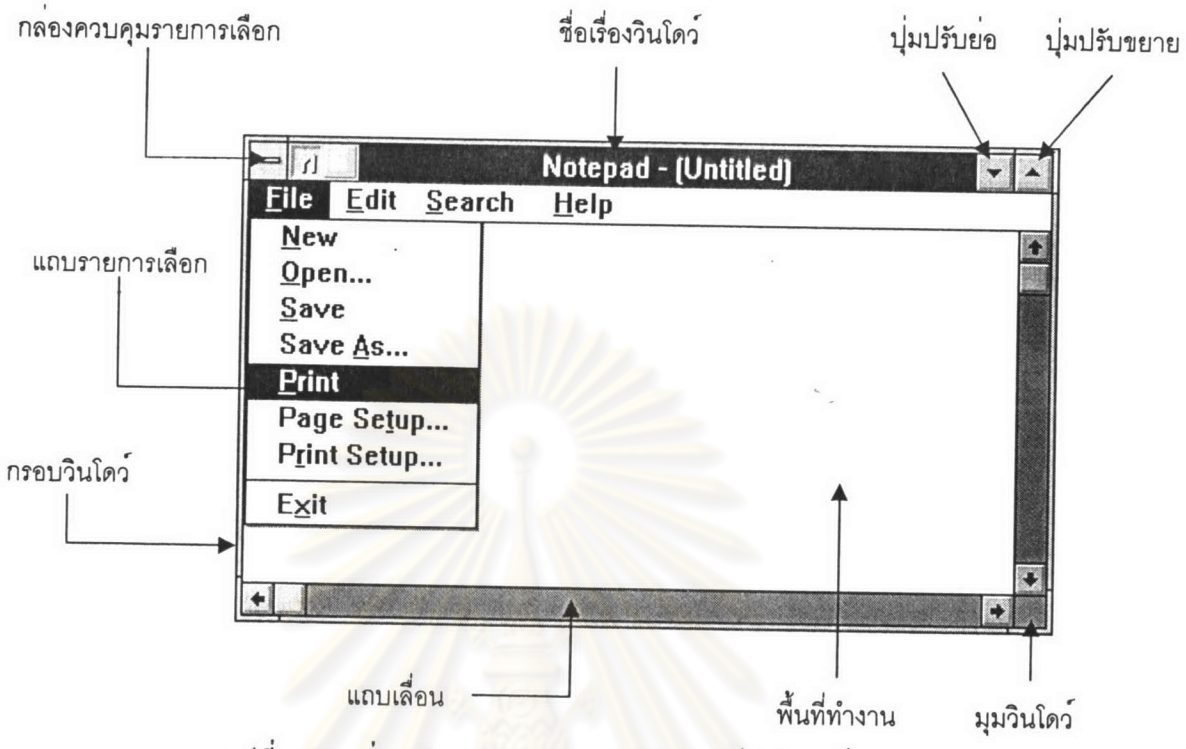
รูปที่ 2.8 GUI Program

โปรแกรมบนวินโดวส์ แบ่งออกได้เป็น 3 ส่วน คือ ส่วนเคอร์เนล (Kernel) ส่วนผู้ใช้ (User) และ ส่วนของการเชื่อมโยงระหว่างอุปกรณ์กับผู้ใช้โดยรูปภาพกราฟิก ทั้ง 3 ส่วนนี้ เป็นโครงสร้างหลัก เมื่อวินโดวส์ทำงาน ตัวของมันจะทำหน้าที่คล้ายกับ ดอสเอ็กซ์เทนเดอร์ (DOS Extender) เพื่อที่จะทำงานกับซีพียู ในภาวะอารักขา (Protect mode) กล่าวคือ สามารถที่จะเรียกใช้หน่วยความจำได้มากกว่าดอส และทำงานแบบหลายงานได้ รวมถึงการป้องกันเมื่อโปรแกรมหลายโปรแกรมที่ทำงานพร้อมกันไม่ให้ยุ่งเกี่ยวกัน

ลักษณะจอภาพในระบบวินโดวส์

ส่วนประกอบต่าง ๆ ของจอภาพในระบบวินโดวส์ สามารถกำหนดไว้ในแฟ้มทรัพยากร (Resource File) ดังรูปที่ 2.9

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 2.9 ส่วนประกอบของจอภาพในระบบวินโดวส์

จากรูปดังกล่าวข้างต้น ส่วนประกอบต่าง ๆ มีดังต่อไปนี้

1. กล่องควบคุมรายการเลือก (Control Menu Box) หรือรายการเลือกของระบบ (System Menu) จะเป็นรูปแคร่ยาว (Space Bar) อยู่ตรงมุมบนซ้ายมือของวินโดว์ สามารถใช้แป้นอักขระเรียกใช้งานได้ด้วยการกดแป้น Alt+space bar หน้าที่ของรายการเลือกนี้ คือ ให้ผู้ใช้สามารถที่จะทำการเปลี่ยนขนาดหรือย้ายตำแหน่งของวินโดว์นั้น ๆ ทำให้วินโดว์อยู่ในรูปของสัญรูป (Icon) และเรียกโปรแกรมประยุกต์ (Application) อื่น ขึ้นมาทำงาน

2. แถบชื่อเรื่อง (Title Bar) เป็นแถบอยู่บนสุดของวินโดว์ สำหรับเป็นที่แสดงชื่อโปรแกรมประยุกต์ หรือชื่อเอกสารนั้น หากวินโดว์นั้นเป็น วินโดว์ที่กำลังทำงานอยู่ สีของแถบชื่อเรื่องจะมีสีที่แตกต่างไปจากวินโดว์อื่น

3. ชื่อเรื่องวินโดว์ (Window Title) จะแตกต่างไปตามชนิดของวินโดว์ อาจเป็นชื่อของโปรแกรมประยุกต์ หรือชื่อเอกสาร หรือชื่อกลุ่ม (Group) หรือชื่อของสารระบบ (Directory) เป็นต้น

4. แถบรายการเลือก (Menu Bar) เป็นรายการของข้อมูลที่โปรแกรมประยุกต์ หรือเอกสารนั้นที่มีให้ใช้งาน

5. แถบเลื่อน (Scroll Bar) เป็นลักษณะของแถบลูกศรอยู่ 2 ด้าน และมีกล่องสี่เหลี่ยม เรียกว่า กล่องแถบเลื่อน (Scroll Box) อยู่ตรงกลาง เพื่อใช้สำหรับเลื่อนไปดูข้อมูลที่ไม่สามารถแสดงได้หมดในวินโดว์เดียว

6. ปุ่มปรับย่อ (Minimize) และปุ่มปรับขยาย (Maximize) เป็นลักษณะของลูกศรที่อยู่มุมบนขวาของวินโดว์ ปุ่มปรับย่อใช้สำหรับทำให้วินโดว์นั้นอยู่ในรูปของสัญรูป ปุ่มปรับขยายใช้สำหรับทำให้ วินโดว์ขยายได้เต็มพื้นที่ของการทำงาน (Working Area)

7. กรอบวินโดว์ (Window Border) ใช้สำหรับย่อหรือขยายวินโดว์ไปทางแนวระนาบหรือแนวตั้ง

8. มุมวินโดว์ (Window Corner) เป็นกรอบที่อยู่ตรงมุมของวินโดว์ ใช้ในการย่อหรือขยายขนาดของวินโดว์ในทั้งสองแนวพร้อม ๆ กัน

9. พื้นที่ทำงาน (Working Area) เป็นพื้นที่ว่างตรงกลาง สำหรับผู้ใช้งาน เช่น พื้นที่ในการวาดรูปในโปรแกรมเพนท์บรัช (Program Paint Brush) เป็นต้น

รูปแบบของโปรแกรมประยุกต์

จุดประสงค์หลักของวินโดวส์ คือสร้างระบบติดต่อกับผู้ใช้แบบใหม่ที่จะใช้ง่ายกว่าเดิม โปรแกรมประยุกต์จะติดต่อกับผู้ใช้โดยผ่านระบบใหม่ เรียกได้ว่าเป็นเครื่องมือแบบใหม่ อันประกอบไปด้วย

1. วินโดว์ (Window)
2. รายการเลือก (Menu)
3. กรอบข้อความ (Dialog Box)
4. ข้อความ (Message)

1. **วินโดว์** เป็นสิ่งที่ติดต่อกับผู้ใช้อย่างพื้นฐาน ที่โปรแกรมประยุกต์ทุกโปรแกรมจะต้องมีใช้เสมอ วินโดว์จะประกอบไปด้วย แถบชื่อเรื่อง แถบรายการเลือก แถบเลื่อน กรอบข้อความ ฯลฯ

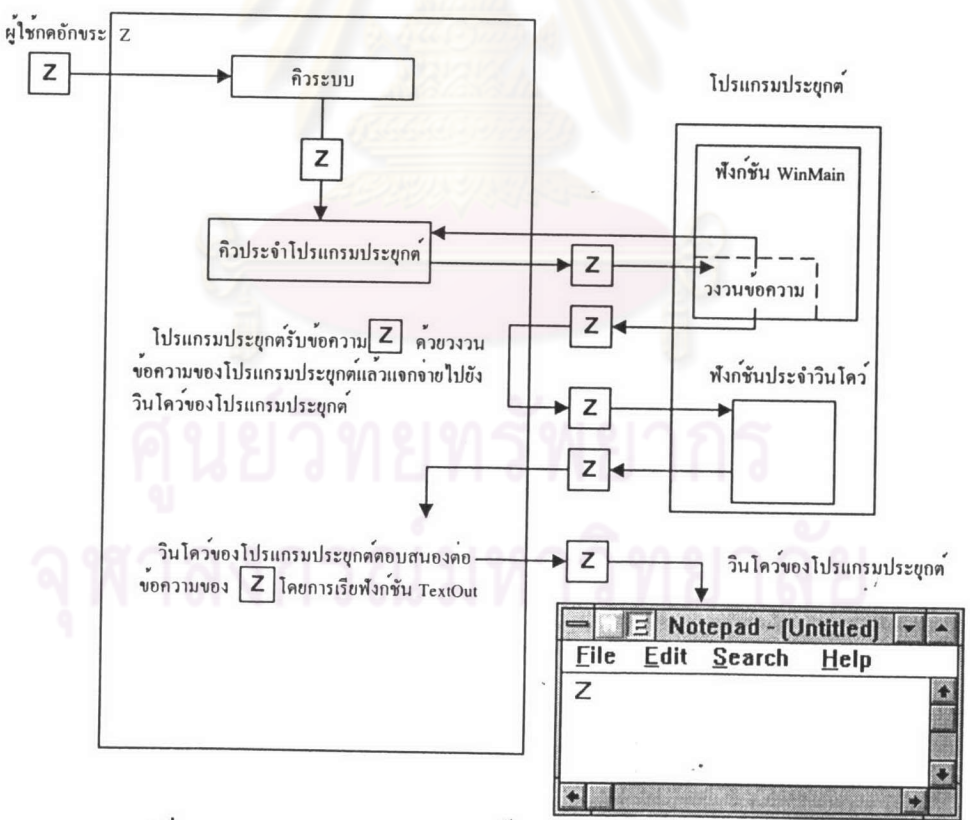
2. **รายการเลือก** เป็นสิ่งที่ใช้รับข้อมูลหลักของโปรแกรมประยุกต์ที่ทำงานบนวินโดวส์ ประกอบไปด้วยรายการคำสั่งเรียงกันไป ซึ่งจะให้ผู้ใช้ได้เลือกดู และใช้งานได้ง่าย สำหรับหน้าที่การจัดการ การแสดงรายการเลือก และการเลือกรายการในรายการเลือก เป็นหน้าที่ของระบบวินโดวส์ แต่หลังจากผู้ใช้เลือกรายการแล้ว ระบบวินโดวส์ก็จะส่งคำสั่งที่ผู้ใช้เลือก ผ่านมาทางคิวข้อความ (Message Queue) เพื่อให้โปรแกรมประยุกต์ได้ตอบสนองต่อคำสั่งนั้น ๆ ต่อไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3. **กรอบข้อความ** เป็นวินโดว์ชนิดหนึ่งที่ใช้งานชั่วคราว เพื่อให้โปรแกรมประยุกต์ใช้ติดต่อกับผู้ใช้ เช่น รับคำสั่งในรายละเอียดให้มากขึ้น

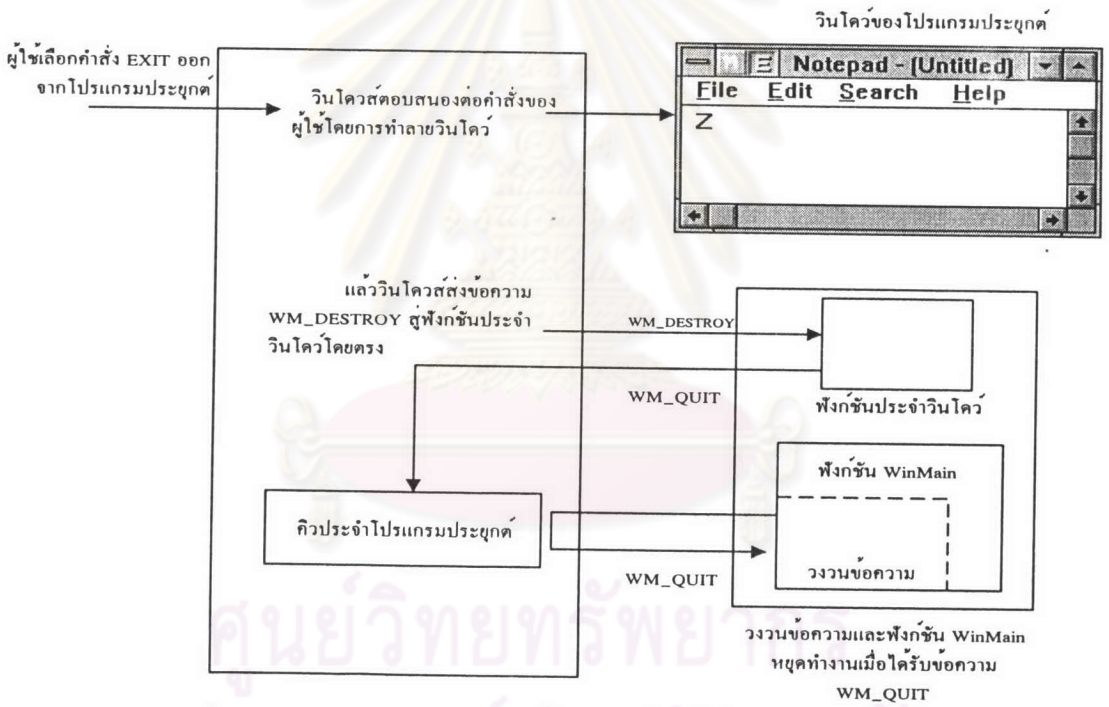
4. **ข้อความ** ด้วยเหตุที่ว่าโปรแกรมประยุกต์จะรับข้อมูลเข้าโดยผ่านทาง คิวข้อความ ส่วนประกอบหลักของโปรแกรมประยุกต์จึงต้องมีส่วนหนึ่งที่เรียกว่า วงวนข้อความ (Message Loop) เพื่อที่จะรับข้อความ และแจกจ่ายไปยังวินโดว์ที่เหมาะสม

ตัวอย่างเช่นเมื่อผู้ใช้กดแป้นอักขระ Z ระบบวินโดวส์จะรับทราบและสร้างข้อความสำหรับแป้นอักขระนั้นไปวางไว้บนคิวระบบ (System Queue) แล้วส่งต่อไปยังคิวประจำโปรแกรมประยุกต์ที่เหมาะสม จากนั้นวงวนข้อความในโปรแกรมประยุกต์จะอ่านข้อความแล้วแจกจ่ายไปยังวินโดว์ที่เหมาะสมของโปรแกรมประยุกต์ ซึ่งก็จะได้เป็นข้อความของตัวหนังสือในรูปของรหัสแอนซี (ANSI) และตอบสนองต่อข้อความด้วยการเรียกฟังก์ชัน TextOut แสดงออกมาเป็นอักษร Z ดังรูปที่ 2.10



รูปที่ 2.10 แสดงการจัดการวงวนข้อความเมื่อผู้ใช้กดอักขระ Z

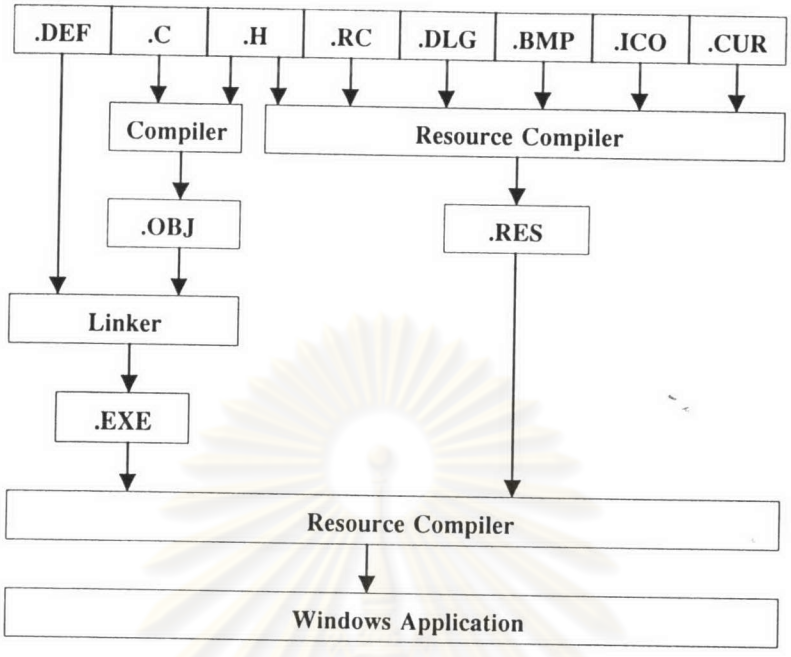
การส่งข้อความ นอกจากจะส่งทางคิวประจำโปรแกรมประยุกต์แล้ว ระบบวินโดวส์ยังสามารถส่งข้อความบางชนิดได้โดยตรงให้กับวินโดว์ของโปรแกรมประยุกต์ เช่น เมื่อผู้ใช้เลือกรายการ Exit จากรายการเลือกของโปรแกรมประยุกต์ ระบบวินโดวส์จะตอบรับการทำลายวินโดว์โดยการส่งข้อความชื่อ WM_DESTROY ไปยังฟังก์ชันประจำวินโดว์โดยตรงโดยไม่ผ่านคิวข้อความ หลังจากนั้นฟังก์ชันประจำวินโดว์ก็จะตอบสนองโดยส่งสัญญาณให้แก่ฟังก์ชัน WinMain ว่าวินโดว์นั้นได้ถูกทำลายแล้วโปรแกรมประยุกต์ควรจะเลิกทำงาน โดยใช้ฟังก์ชัน PostQuitMessage ส่งข้อความชื่อ WM_QUIT ไปยังคิวประจำโปรแกรมประยุกต์ เมื่อวงวนข้อความทำงานแล้วได้รับข้อความนี้ก็จะเลิกทำงาน ดังแสดงในรูปที่ 2.11



รูปที่ 2.11 แสดงการจัดการวงวนข้อความเมื่อผู้ใช้เลือกรายการ Exit

หลักการพัฒนาโปรแกรมประยุกต์บนระบบวินโดวส์

หลักการพัฒนาโปรแกรมเพื่อทำงานบนไมโครซอฟต์วินโดวส์อยู่บนพื้นฐานของการพัฒนาโปรแกรมแบบมัลติมอดูล (Multimodule Programming) กล่าวคือจะประกอบด้วยแฟ้มประเภทต่าง ๆ ที่ถูกแปล และนำมาเชื่อมต่อกันเป็นโปรแกรมประเภทแฟ้ม .exe ในที่นี้จะยกตัวอย่างภายใต้การพัฒนาโปรแกรมด้วยภาษาซี ซึ่งเป็นภาษาที่ใช้ในการพัฒนาวิทยานิพนธ์นี้ โดยแฟ้มต่าง ๆ มีความสัมพันธ์กันดังแสดงในรูปที่ 2.12



รูปที่ 2.12 แสดงส่วนประกอบและขั้นตอนการพัฒนาโปรแกรม

จากรูปดังกล่าวข้างต้น ส่วนประกอบของการพัฒนาโปรแกรมประยุกต์บนวินโดวส์มีดังนี้

1. แฟ้มต้นฉบับ (Source File)

เป็นแฟ้มโปรแกรมที่ประกอบด้วยคำสั่งต่าง ๆ ที่เขียนขึ้นโดยใช้ภาษาซี เป็นแฟ้มประเภท .c เช่น editor.c ภายในแฟ้มจะประกอบด้วยฟังก์ชันต่าง ๆ ตามขั้นตอนการทำงานที่กำหนด โดยโครงสร้างของแฟ้มจะต้องประกอบด้วยส่วนสำคัญดังนี้

1.1 การรวม (Include) แฟ้มประเภท .h เช่น windows.h ซึ่งเป็นแฟ้มข้อมูลส่วนหัวที่ต้องมีเสมอ เนื่องจากเป็นแฟ้มที่เก็บการประกาศฟังก์ชันต่าง ๆ ที่ระบบวินโดวส์เตรียมไว้ให้ใช้ได้แก่ ประเภทข้อมูล ค่าคงที่ และโครงสร้างข้อมูลต่าง ๆ

1.2 ฟังก์ชัน WinMain เป็นฟังก์ชันที่เป็นจุดเริ่มต้นของการทำงานคล้ายกับฟังก์ชัน main ของโปรแกรมภาษาซีทั่ว ๆ ไป โดยจะต้องใช้ชื่อนี้เท่านั้น และมีหน้าที่ดังนี้

1.2.1 การลงทะเบียนวินโดว์คลาส (Register Window Class) เป็นการกำหนดค่าให้กับตัวแปรโครงสร้างชนิด WNDCLASS เพื่อกำหนดรายละเอียดลักษณะพื้นฐานของวินโดว์ที่ต้องการสร้างในโปรแกรม เช่น ชื่อของกระบวนงานวินโดว์ (Window Procedure) สัญรูป และตัวชี้ตำแหน่งของวินโดว์นั้น ๆ การลงทะเบียนนี้จะกระทำเพียงครั้งแรกที่โปรแกรมถูกเรียกขึ้นมาทำงาน โดยถ้าก่อนหน้านี้มีโปรแกรมเดียวกันทำงานอยู่ก่อนแล้ว ระบบวินโดวส์ก็จะไม่ทำการลงทะเบียนวินโดว์คลาสอีก ซึ่งเป็นผลดีทำให้ไม่เปลืองเนื้อที่หน่วยความจำ

1.2.2 การสร้างวินโดว์ (Creating Window) ในขั้นตอนนี้จะเป็นการสร้างวินโดว์หลักของโปรแกรม โดยจะมีการกำหนดชนิดของวินโดว์ที่ต้องการสร้าง ขนาดและตำแหน่งเริ่มต้นของวินโดว์ รายการเลือก การสร้างวินโดว์นี้จะเป็นการสร้างขึ้นภายในตัวไมโครซอฟต์วินโดวส์เท่านั้น ยังไม่ถูกแสดงผลบนจอภาพ จะต้องเรียกฟังก์ชัน ShowWindow และ UpdateWindow โดยฟังก์ชัน ShowWindow จะทำหน้าที่สร้างวินโดว์ตามลักษณะที่ต้องการบนหน้าจอ และฟังก์ชัน UpdateWindow จะทำหน้าที่ส่งข้อความ WM_PAINT ไปให้กับกระบวนงานวินโดว์ (Window Procedure) เพื่อให้พื้นที่ภายในวินโดว์ถูกวาดตามที่โปรแกรมกำหนด

1.2.3 วงวนข้อความ (Message Loop) เป็นส่วนที่โปรแกรมจะเชื่อมต่อกับระบบวินโดวส์ โดยจะเป็นการทำงานในลักษณะวนไปเรื่อย ๆ จนกว่าต้องการจะจบการทำงานของโปรแกรม ภายในวงวนจะมีฟังก์ชันการทำงานสองฟังก์ชัน ได้แก่ TranslateMessage ที่ทำหน้าที่แปลงข้อความเกี่ยวกับแผงแป้นอักขระให้มีค่าที่สามารถนำไปใช้งานได้ และ DispatchMessage ที่ทำหน้าที่ส่งข้อความกลับไปยังระบบวินโดวส์ เพื่อให้ระบบวินโดวส์จัดการส่งไปยังฟังก์ชันประจำวินโดว์ต่อไป

1.2.4 กระบวนงานวินโดว์ (Window Procedure) ในส่วนนี้ถือว่าเป็นส่วนสำคัญของการเขียนโปรแกรมเพื่อทำงานบนไมโครซอฟต์วินโดวส์ เพราะว่าเป็นส่วนที่ทำหน้าที่ประมวลผลข้อความต่าง ๆ ที่ถูกส่งมา โดยชื่อของฟังก์ชันจะต้องกำหนดไว้ในขั้นตอนการลงทะเบียนวินโดว์คลาส เพื่อให้วินโดวส์รู้ว่าต้องส่งข้อความไปประมวลผลที่ฟังก์ชันใด โดยทั่วไปรูปแบบภายในของฟังก์ชันจะอยู่ในรูปของการใช้คำสั่ง switch -case statement เพื่อเลือกเอาข้อความไปประมวลผลเฉพาะตามต้องการ ส่วนข้อความที่ไม่ต้องการจะต้องถูกส่งไปที่ DefWindowProc ซึ่งเป็นฟังก์ชันภายในระบบวินโดวส์ที่ถูกกำหนดให้ทำงานกับข้อความต่าง ๆ ที่ไม่ได้ถูกประมวลผล ซึ่งค่าที่ส่งกลับจาก DefWindowProc นี้ จะถูกส่งกลับคืนสู่ระบบวินโดวส์อีกทีหนึ่ง

2. แฟ้มข้อมูลนิยาม (Definition File)

เป็นแฟ้มเก็บชื่อโปรแกรม เซกเมนต์ และหน่วยความจำที่ต้องการ เป็นแฟ้มประเภท .def เช่น editor.def ช่วยในขั้นตอนการเชื่อมต่อโปรแกรม โดยประกอบด้วยข้อมูลทางเทคนิคของโปรแกรมที่สร้างขึ้น เช่น ขนาดของสแตก ขนาดของหน่วยความจำขณะเริ่มต้นโปรแกรม ชื่อของโปรแกรมที่จะถูกเรียกให้ทำงาน คุณสมบัติของส่วนรหัส (Code Segment) และ ส่วนข้อมูล (Data Segment) รวมทั้งชื่อของกระบวนงานวินโดว์ทั้งหมด

3. แฟ้มข้อมูลส่วนหัว (Header File)

เป็นแฟ้มข้อมูลส่วนหัวเช่นเดียวกับการเขียนโปรแกรมภาษาซีทั่วไป เป็นแฟ้มประเภท .h เช่น windows.h

4. แฟ้มข้อมูลทรัพยากร (Resource Script File)

เป็นแฟ้มข้อมูลที่ใช้กำหนดการใช้ทรัพยากรต่าง ๆ ของโปรแกรม เป็นแฟ้มประเภท .rc เช่น editor.rc ประกอบด้วย การบรรยายการสร้างรายการเลือกและแป้นลัด ข้อความชื่อและชนิดของแฟ้มข้อมูลแบบแผนที่บิต (Bitmap File) แฟ้มข้อมูลสัญรูป แฟ้มข้อมูลตัวชี้ตำแหน่ง รวมทั้งแฟ้มกล่องคำโต้ตอบ (Dialog Box) ต่าง ๆ

5. แฟ้มข้อมูลคำโต้ตอบ (Dialog File)

เป็นแฟ้มข้อมูลที่ใช้ในการบรรยายการสร้างกล่องคำโต้ตอบต่าง ๆ เป็นแฟ้มประเภท .dlg เช่น help.dlg

6. แฟ้มข้อมูลแบบแผนที่บิต (Bitmap File)

เป็นแฟ้มข้อมูลรูปภาพแบบแผนที่บิตที่ใช้ในโปรแกรม เป็นแฟ้มประเภท .bmp เช่น sym.bmp

7. แฟ้มข้อมูลสัญรูป (Icon File)

เป็นแฟ้มข้อมูลรูปภาพสัญรูปที่ใช้ในโปรแกรม เป็นแฟ้มประเภท .ico เช่น edtool.ico

8. แฟ้มข้อมูลตัวชี้ตำแหน่ง (Cursor File)

เป็นแฟ้มข้อมูลรูปภาพตัวชี้ตำแหน่งที่ใช้ในโปรแกรม เป็นแฟ้มประเภท .cur เช่น editor.cur

แฟ้มข้อมูลทั้งหมดนี้จะถูกแปลและนำมาเชื่อมต่อกัน จนกลายเป็นโปรแกรมที่สามารถทำงานบนไมโครซอฟต์วินโดวส์ได้ สำหรับรายละเอียดปลีกย่อยอื่น ๆ ตลอดจนเทคนิคชั้นสูงสามารถศึกษาเพิ่มเติมได้จากคู่มือสำหรับการเขียนโปรแกรมเพื่อใช้งานบนไมโครซอฟต์วินโดวส์ (Petzold, 1992)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4. **ขั้นตอนที่ใช้ในการวิจัย**

การเรียงข้อมูล (Sorting) (ดร.สุชาย ธนวเสถียร และ วิชัย จิวังกูร,2521)

การเรียงข้อมูล หมายถึง การจัดเรียงลำดับข้อมูลตามตรรกใดตรรกหนึ่ง ซึ่งมีความจำเป็นหลายประการ นอกจากจะทำให้สะดวกต่อการค้นหาและตีความแล้ว ยังมีผลต่อกระบวนการดึงข้อมูลมาใช้ ในการทำวิจัยในครั้งนี้ได้มีการนำวิธีการเรียงข้อมูลมาใช้ในการเรียงลำดับคำสงวนตามลำดับตัวอักษร

การเรียงข้อมูลแบ่งได้เป็น 2 ลักษณะใหญ่ ๆ คือ

1. **การเรียงข้อมูลแบบภายใน (Internal Sorrting)**

การเรียงข้อมูลแบบภายใน หมายถึงว่าจำนวนข้อมูลมีขนาดไม่ใหญ่กว่าขนาดของพื้นที่ความจำที่กำหนดให้กับผู้ใช้แต่ละราย ดังนั้นข้อมูลทั้งหลายจะเก็บอยู่ในหนวยความจำหลัก โดยผู้ใช้สามารถใช้แบบการคำนวณสำหรับการเรียงข้อมูล อ่านข้อมูลแต่ละชิ้นจากหน่วยความจำหลักหรือเขียนข้อมูลสู่หน่วยความจำหลักได้เลย โดยไม่จำเป็นต้องใช้หน่วยความจำรองช่วย การจัดเรียงลำดับข้อมูลภายในโดยทั่วไปมีหลายวิธีด้วยกัน วิธีที่สำคัญมีดังนี้

1.1 แบบการเปรียบเทียบ (Sorting by Comparison) แบ่งได้เป็น 2 ชนิด คือ

1.1.1 แบบการเปรียบเทียบปกติ วิธีการทำคือ นำเอาค่าของข้อมูลแต่ละระเบียนมาทำการเปรียบเทียบกันทุกตัว ข้อมูลชุดใดที่มีค่าน้อยกว่าจะถูกสลับที่

1.1.2 แบบการเปรียบเทียบโดยการนับ เป็นการเรียงลำดับแบบการเปรียบเทียบผสมกับการนับ การทำคือจะหาว่าข้อมูลแต่ละชุดอยู่ในลำดับที่เท่าไหร่ของข้อมูลทั้งหมด จะไม่มีการสลับตำแหน่งของข้อมูล

1.2 แบบการแทรก (Insertion Sort) การเรียงลำดับแบบนี้จะถือเสมือนว่าข้อมูลแต่ละชุดมีการเรียงลำดับอยู่ก่อนแล้ว ดังนั้นการแทรกชุดของข้อมูลจะกระทำโดยการนำข้อมูลชุดที่จะแทรก ทำการแทรกเข้าไปในตำแหน่งที่ข้อมูลนั้นควรจะอยู่ โดยเพิ่มเข้าไปทีละชุดจนครบ

1.3 แบบเชลล์ (Shell Sort) การเรียงลำดับแบบนี้ เป็นการปรับปรุงวิธีการเรียงลำดับแบบแทรก นั่นคือแทนที่จะเรียงข้อมูลทั้งชุดโดยตรง เราจะแบ่งข้อมูลทั้งชุดออกเป็นส่วนย่อย ๆ แล้วเรียงส่วนย่อย ๆ เหล่านั้นแทน

1.4 แบบฟอง (Bubble Sort) เป็นวิธีการเรียงลำดับโดยมีหลักการคือ จะเปรียบเทียบค่า 2 ค่าที่ติดกัน ถ้ามันไม่ได้อยู่ในลำดับที่เรากำหนด เช่น จากน้อยไปมาก ก็ให้แลกเปลี่ยนตำแหน่งของค่าทั้ง 2 ค่านั้น แล้วเอาค่าน้อย (หรือค่ามาก ถ้าเป็นการเรียงจากค่ามากไปหาค่าน้อย) เปรียบเทียบกับค่าถัดไปอีกเป็นเช่นนี้ตลอดไปจนกว่าจะอยู่ในลำดับที่ถูกต้อง

1.5 แบบควิกซอร์ต (Quick Sort) การเรียงลำดับแบบนี้ เป็นการปรับปรุงจากการเรียงลำดับแบบฟอง โดยการใช้ผลจากการเปรียบเทียบในครั้งก่อน เพื่อตัดสินใจว่าจะเปรียบเทียบคีย์ใดต่อไป

1.6 แบบเรดิกซ์-เอกซ์เชนจ์ (Redix-exchange Sort) การเรียงลำดับแบบนี้จะทำโดยการเปลี่ยนค่าคีย์ทุกตัวให้เก็บอยู่ในรูปของเลขฐานสอง การเปรียบเทียบจะทำทีละบิต ว่าเป็น 0 หรือ 1 ซึ่งมีหลักการทำงานคล้ายกับแบบควิกซอร์ต

ในการทำวิจัยครั้งนี้ ได้เลือกใช้วิธีการเรียงข้อมูลแบบควิกซอร์ต เพราะวิธีการนี้เหมาะสมกับโครงสร้างข้อมูลที่ใช้ในการพัฒนาโปรแกรม

2. **การเรียงข้อมูลแบบภายนอก (External Sorting)**

ถ้าข้อมูลมีจำนวนมากเกินกว่าที่จะบรรจุลงในพื้นที่ความจำหลักได้หมดในคราวเดียว จึงจำเป็นจะต้องแบ่งข้อมูลออกเป็นส่วนย่อย ๆ ซึ่งแต่ละส่วนมีขนาดใหญ่พอที่จะอยู่ในหน่วยความจำหลักได้ และจะได้รับการเรียงโดยใช้แบบการคำนวณสำหรับการเรียงแบบภายในตามที่ผู้ออกแบบเห็นเหมาะสม หลังจากนั้นก็ต้องมีการนำข้อมูลย่อยของส่วนที่เรียงแล้วไปเก็บไว้ในจานแม่เหล็กหรือเทปเป็นการชั่วคราว เพื่อรอที่จะรวมกับข้อมูลส่วนอื่น ๆ ที่เรียงแล้ว การเรียงแบบภายนอกจะต้องคิดถึงเวลาที่สูญเสียไป อันเนื่องจากการถ่ายเทข้อมูลระหว่างจากแม่เหล็กหรือเทปกับหน่วยความจำหลักด้วย เวลาที่สูญเสียไปในการถ่ายเทข้อมูลระหว่างหน่วยความจำหลักกับจานแม่เหล็กหรือเทป จะเป็นตัวระบุความดีเลวของแบบการคำนวณแบบเรียงภายนอก ในการทำวิจัยครั้งนี้ได้เลือกใช้การเรียงข้อมูลแบบภายใน เพราะใช้กับข้อมูลที่มีขนาดไม่ใหญ่มาก

การค้นหาข้อมูล (Searching) (ดร.สุชาย ธนวเสถียร และ วิชัย จิวังกูร,2521)

การค้นหาข้อมูล หมายถึง การค้นหาข้อมูลที่มีลักษณะเหมือนคุณสมบัติที่กำหนดไว้ล่วงหน้า ขั้นตอนในการค้นหาจะเร็วหรือช้าขึ้นกับขั้นตอนวิธี (Algorithm) ที่ใช้ และขั้นตอนวิธีที่ใช้ก็ขึ้นกับลักษณะการเก็บข้อมูลอีกต่อหนึ่ง โครงสร้างข้อมูลที่ใช้จะเป็นตัวบ่งบอกถึงขั้นตอนวิธีที่ใช้ค้นหาข้อมูลโดยตรง ในกรณีที่ข้อมูลเก็บอยู่ในหน่วยความจำหลักทั้งหมด การค้นหาจะช้าหรือเร็วขึ้นอยู่กับจำนวนครั้งที่ต้องทำการเปรียบเทียบ ส่วนในกรณีข้อมูลมีขนาดใหญ่มากจนไม่สามารถบรรจุอยู่ในหน่วยความจำหลักได้หมด ส่วนหนึ่งต้องเก็บไว้ในหน่วยความจำรอง ในภาวะเช่นนี้ต้องใช้เทคนิคการเสาะค้นข้อมูลอีกแบบหนึ่ง ซึ่งต้องคำนึงถึงเวลาที่สูญเสียไปในระหว่างการถ่ายเทข้อมูลระหว่างหน่วยความจำหลักและหน่วยความจำรองด้วย

การค้นหาข้อมูลมีหลายแบบด้วยกัน เช่น

1. การค้นหาแบบทวิภาค (Binary Search) วิธีนี้จะใช้ระบบแบ่งครึ่งข้อมูลทั้งหมดออกเป็นสองส่วนการค้นหาตัวที่ต้องการจะต้องเลือกหาทีละส่วน โดยใช้วิธีเดิม คือแบ่งเป็นสองส่วนย่อย หากพบในส่วนหนึ่งก็ไม่ต้องเสียเวลาหาในอีกส่วนหนึ่ง

2. การค้นหาแบบครอบคลุม (Global Search) เป็นการค้นหาทั่วทั้งหมดของข้อมูลว่ามีข้อความหรือข้อมูลที่ต้องการหรือไม่

3. การค้นหาแบบลำดับ (Sequential Search) เป็นการค้นหาโดยใช้วิธีการเปรียบเทียบข้อความที่ต้องการกับข้อมูลที่เก็บอยู่โดยเรียงไปทีละชุดของข้อมูล วิธีนี้ทำได้ง่ายที่สุดแต่จะช้าที่สุด แต่มีข้อดีคือ ข้อมูลไม่ต้องเรียงลำดับ ดังนั้นการเพิ่มข้อมูลหรือนำข้อมูลออกจะสะดวกมาก ถ้าข้อมูลนั้นเก็บแบบรายการโยง

ในการทำวิจัยครั้งนี้ ได้ใช้วิธีการค้นหาแบบทวิภาค เนื่องจากมีการเรียงลำดับของข้อมูลอยู่แล้วซึ่งจะเหมาะสมกับการค้นหาแบบนี้

5. ภาษาซีและภาษาปาสคาล

ภาษาปาสคาล (บุญเลิศ เอี่ยมทัศนา,2521)

ภาษาปาสคาลแตกต่างกับภาษาระดับสูงอื่น ๆ ตรงที่ภาษาเหล่านั้นได้ถูกออกแบบมาสำหรับใช้งานกับคอมพิวเตอร์โดยตรง แต่ภาษาปาสคาลถูกออกแบบมาเพื่อเป็นภาษาสำหรับเรียนเขียนโปรแกรม โดยที่ ดร.เวียร์ต (Profeesor Doctor Niklaus Wirth) ผู้สร้างภาษานี้ขึ้นในปี พ.ศ. 2514 มีวัตถุประสงค์เพื่อให้เป็นภาษาสำหรับฝึกการเขียนโปรแกรมอย่างมีระบบและมีระเบียบ

เมื่อ ดร.เวียร์ต ได้ออกแบบภาษาปาสคาลแล้ว ได้มีผู้นำไปเขียนเป็นโปรแกรมตัวแปลภาษาใช้กันอย่างแพร่หลาย ทั้งในเครื่องคอมพิวเตอร์ขนาดใหญ่และไมโครคอมพิวเตอร์ กล่าวกันว่าสำหรับไมโครคอมพิวเตอร์ โปรแกรมตัวแปลภาษาปาสคาลที่ได้รับความนิยมมากที่สุดได้แก่ เทอร์โบปาสคาลของบริษัทบอร์แลนด์

ตัวภาษาปาลคาลมีสมรรถนะไม่ยิ่งหย่อนไปกว่าภาษาระดับสูงอื่น ๆ และดีกว่าอีกหลายภาษาด้วย ดังนั้นการศึกษาภาษาปาสคาลจึงได้ทั้งภาษาที่จะใช้งานกับเครื่องคอมพิวเตอร์โดยตรงและวิธีการเขียนโปรแกรมที่ถูกต้องเพื่อนำไปใช้ในการเขียนโปรแกรมในภาษาระดับสูงอื่นและภาษาเครื่อง

ภาษาปาสคาลเป็นภาษาที่สนับสนุนให้เขียนโปรแกรมแบบเป็นมอดูล (Modular Programming) เป็นภาษาที่ต้องเขียนโปรแกรมเป็นโครงสร้าง (Structure Programming) และเป็นภาษาที่ให้เขียนโปรแกรมในแบบเรียกซ้ำ (Recursive Programming)

คำสงวน เป็นคำที่ใช้กำหนดรูปแบบต่าง ๆ ในโปรแกรม คำเหล่านี้จะนำมาใช้เป็นชื่อไม่ได้ ในภาษาปาสคาลมาตรฐานมีคำสงวนดังนี้

AND	ELSE	IN	PROCEDURE	UNTIL
ARRAY	END	LABEL	PROGRAM	VAR
BEGIN	FILE	MOD	RECORD	WHILE
CASE	FOR	NIL	REPEAT	WITH
CONST	FORWARD	NOT	SET	
DIV	FUNCTION	OF	THEN	
DO	GOTO	OR	TO	
DOWNTO	IF	PACKED	TYPE	

ในวิทยานิพนธ์นี้จะมีการแสดงสีในคำสงวนดังกล่าวข้างต้น

ข้อความสั่งกระทำการในภาษาปาสคาล (Executable Statement in PASCAL) (Schneider,1981) ข้อความสั่งแบ่งเป็นกลุ่ม ๆ ได้ดังนี้

ข้อความสั่งกำหนดค่า (Assignment Statement)

เป็นข้อความสั่งสำหรับกำหนดค่าให้แก่ตัวแปร มีรูปแบบดังนี้

ตัวแปร := นิพจน์ (Expression)

ข้อความสั่งนำเข้า/นำออก (Input/Output Statement)

เป็นข้อความสั่งสำหรับการนำเข้าค่าของตัวแปรกับอุปกรณ์ภายนอก เช่น แผงแป้นอักขระ และนำออกค่าคงที่หรือค่าของตัวแปรกับอุปกรณ์ภายนอก เช่น แผงจอภาพ โดยมีรูปแบบดังนี้

ข้อความสั่งนำเข้า read(ตัวแปร) หรือ readln(ตัวแปร)

ข้อความสั่งนำออก write(ตัวแปร) หรือ writeln(ตัวแปร)

ข้อความสั่งประกอบ (Compound Statement)

เป็นข้อความสั่งที่รวมเอาหลายข้อความสั่งเข้าด้วยกัน โดยใช้คำขึ้นต้นว่า Begin จบด้วยคำว่า End ซึ่งจะถือว่าเป็น 1 ข้อความสั่งประกอบ โดยมีรูปแบบดังนี้

Begin

ข้อความสั่ง 1;

ข้อความสั่ง 2;

ข้อความสั่ง n

End

การแยกข้อความสั่งจะแยกกันด้วยเครื่องหมาย ";" แต่ข้อความสั่งสุดท้ายไม่ต้องมีเครื่องหมายนี้ ส่วนหลัง End ไม่ใส่เครื่องหมายนี้ไว้เนื่องจากจะมีทั้งที่ต้องใส่และไม่ต้องใส่ ขึ้นอยู่กับว่าจะนำข้อความสั่งประกอบนี้ไปแทนข้อความสั่งใด

ข้อความสั่งมีเงื่อนไข (Conditional Statement)

เป็นข้อความสั่งที่ให้เลือกปฏิบัติตามเงื่อนไข ได้แก่ If If-Else และ Case ซึ่งมีรายละเอียดดังนี้

ข้อความสั่ง If (If Statement) เป็นการเลือกปฏิบัติเมื่อเงื่อนไขเป็นจริง โดยมีรูปแบบดังนี้

If เงื่อนไข **Then** ข้อความสั่ง;

ข้อความสั่ง If-Else (If-Else Statement) เป็นการเลือกปฏิบัติถ้าเงื่อนไขเป็นจริงจะปฏิบัติข้อความสั่ง 1 และถ้าเป็นเท็จจะปฏิบัติข้อความสั่ง 2 โดยมีรูปแบบดังนี้

If เงื่อนไข **Then** ข้อความสั่ง 1 **Else** ข้อความสั่ง 2;

ข้อความสั่ง Case (Case Statement) เป็นการเลือกปฏิบัติเมื่อมีกรณีให้เลือกหลายกรณี และนิพจน์ที่นำมาเป็นเงื่อนไขต้องให้ค่าเป็นข้อมูลแบบมีลำดับที่ (Scalar Type) โดยมีรูปแบบดังนี้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Case เงื่อนไข **Of**
1 : ข้อความสั่ง 1;
2 : ข้อความสั่ง 2;

n : ข้อความสั่ง n;
Else
ข้อความสั่ง ;
End;

ข้อความสั่งกระโดด (Jump Statement)

เป็นข้อความสั่งที่เป็นการสั่งให้กระโดดไปยังที่หมาย ในภาษาปาสคาลมีอยู่ข้อความเดียวคือ ข้อความ Goto (Goto Statement) ซึ่งมีข้อห้ามว่าห้ามกระโดดเข้าหรือออกจากถ้อยแถลงโครงสร้างหรือโปรแกรมย่อย ซึ่งข้อความสั่งนี้จะใช้คู่กับ LABEL โดยมีรูปแบบดังนี้

Goto Label1;
Label1: ข้อความสั่ง ;

ข้อความสั่งสร้างวงวน (Loop Constructive Statement)

เป็นข้อความสั่งที่ทำงานเป็นวงวน (Loop) ได้แก่ While Repeat และ For ซึ่งมีรายละเอียดดังนี้

ข้อความสั่ง While (While Statement) เป็นข้อความสั่งทำงานเป็นวงวนที่ทำการตรวจสอบเงื่อนไขก่อน หากเงื่อนไขเป็นจริงจะเข้าทำงานในวงวน และจะทำซ้ำอยู่จนกว่าเงื่อนไขเป็นเท็จ หากเงื่อนไขเป็นเท็จแต่แรกจะไม่เข้าทำงานในวงวน มีรูปแบบดังนี้

While เงื่อนไข **Do** ข้อความสั่ง;

ทางเข้าของข้อความสั่งนี้อยู่ตรงคำว่า While แล้วปฏิบัติตามข้อความสั่ง ซึ่งต้องมีเพียงข้อความสั่งเดียว หากต้องมีหลายข้อความสั่งให้ทำเป็นข้อความสั่งประกอบ

ข้อความสั่ง Repeat (Repeat Statement) เป็นข้อความสั่งทำงานเป็นวงวนจนกว่าเงื่อนไขจะเป็นจริง โดยมีรูปแบบดังนี้

Repeat ข้อความสั่ง **Until** เงื่อนไข;

ทางเข้าของข้อความสั่งนี้อยู่ตรงคำว่า Repeat แล้วปฏิบัติตามข้อความสั่งซึ่งจะมีหลายข้อความสั่งก็ได้ โดยปฏิบัติซ้ำ ๆ กัน จนกว่าเงื่อนไขเป็นจริง

ข้อความสั่ง For (For Statement) เป็นข้อความสั่งทำงานเป็นวงวนที่มีการกำหนดจำนวนครั้งที่แน่นอนในการวน โดยมีรูปแบบดังนี้

For ตัวแปร := ค่าต้น **To** ค่าสุดท้าย **Do** ข้อความสั่ง ;

หรือ **For** ตัวแปร := ค่าต้น **DownTo** ค่าสุดท้าย **Do** ข้อความสั่ง ;

แบบแรกที่ใช้ **To** จะเพิ่มค่าตัวแปรทีละหนึ่งจนกว่าจะมีค่าเท่ากับค่าสุดท้าย

แบบหลังที่ใช้ **DownTo** จะลดค่าตัวแปรลงทีละหนึ่งจะกว่าจะเท่ากับค่าสุดท้าย

จากข้อความสั่งกระทำการในภาษาปาสคาลดังกล่าวข้างต้น ในวิทยานิพนธ์ฉบับนี้ได้มีการขยายรูปแบบของข้อความสั่งดังต่อไปนี้

1. ข้อความสั่ง Case
2. ข้อความสั่ง If-Else
3. ข้อความสั่ง For
4. ข้อความสั่ง Repeat
5. ข้อความสั่ง While

หมายเหตุ ข้อความที่แสดงเป็นตัวเข้มในรูปแบบข้อความสั่ง จะแสดงสีบนจอภาพ

ภาษาซี (สุขกรี เสรีวัลย์สถิตย์)

ภาษาซี เป็นภาษาที่ออกแบบและพัฒนาโดย เดนนิส ริดชี่ (Dennis Ritchie) แห่งห้องทดลองเบลล์ (Bell laboratories) โดยมุ่งให้เป็นภาษาสำหรับใช้พัฒนาโปรแกรมจัดการระบบงานบนยูนิกซ์ ทั้งนี้เนื่องจากระบบจัดการยูนิกซ์ ส่วนใหญ่จะถูกพัฒนาขึ้นโดยใช้ภาษาซี จึงเป็นการชี้ให้เห็นว่าภาษาซี เหมาะสำหรับงานการพัฒนาโปรแกรมระบบ และมักจะถูกถือว่าเป็นทั้งภาษาระดับต่ำ (Low level language) และภาษาระดับสูง (High level language) ที่กล่าวว่าเป็นภาษาระดับต่ำเนื่องจากภาษาซีมีวิธีการเข้าถึงในระดับที่ต่ำที่สุดของฮาร์ดแวร์ (Hardware) ในอีกด้านหนึ่งก็เป็นภาษาระดับสูงด้วย เนื่องจากมีการใช้ข้อมูลและโครงสร้างการควบคุมการทำงานของโปรแกรมเป็นอย่างเดียวกับภาษาระดับสูงอื่น ๆ ทั้งสองด้านของภาษานี้เป็นสิ่งที่เกื้อหนุนซึ่งกันและกัน ความสามารถในระดับต่ำจะทำให้ภาษาซีสามารถใช้กับเฉพาะเครื่องได้ และความสามารถในระดับสูงทำให้ภาษาซีเป็นอิสระจากฮาร์ดแวร์ และสามารถใช้งานกับฮาร์ดแวร์ที่

หลากหลายได้ ภาษาซีสามารถสร้างรหัสภาษาเครื่องซึ่งตรงกับชนิดของข้อมูลนั้นได้เอง ลักษณะเช่นนี้ทำให้โปรแกรมภาษาซีที่พัฒนาบนเครื่อง ๆ หนึ่ง สามารถนำไปใช้ในอีกเครื่องหนึ่งได้

คำสงวน (Kernighan B. and Ritchi D.,1988) ภาษาซีมีคำสงวนมาตรฐาน ซึ่งไม่สามารถนำไปใช้เป็นอย่างอื่นได้มีดังนี้

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

ข้อความสั่งกระทำการในภาษาซี (Executable Statement in C) (Kernighan B. and Ritchi D.,1988) ข้อความสั่งแบ่งเป็นกลุ่ม ๆ ได้ดังนี้

ข้อความสั่งกำหนดค่า (Assignment Statement)

เป็นข้อความสั่งสำหรับกำหนดค่าให้แก่ตัวแปร มีรูปแบบดังนี้

ตัวแปร = นิพจน์ (Expression) ;

ข้อความสั่งประกอบ (Compound Statement)

เป็นข้อความสั่งที่รวมเอาหลายข้อความสั่งเข้าด้วยกัน ในภาษาซีจะใช้วงเล็บปีกกา { และ } เพื่อรวบรวมข้อความสั่งเข้าด้วยกัน หรือเรียกว่าบล็อก โดยมีรูปแบบดังนี้

```
{  ข้อความสั่ง 1;
   ข้อความสั่ง 2;

   ข้อความสั่ง n; }
```

ข้อความสั่งมีเงื่อนไข (Conditional Statement)

เป็นข้อความสั่งที่ให้เลือกปฏิบัติตามเงื่อนไข ได้แก่ If-Else และ Switch ซึ่งมีรายละเอียดดังนี้

ข้อความสั่ง If-Else (If-Else Statement) ใช้ในการแสดงการตัดสินใจ โดยมีรูปแบบดังนี้

```
if (นิพจน์)
   ข้อความสั่ง 1;
else
   ข้อความสั่ง 2;
```

โดยส่วน **else** จะมีหรือไม่มีก็ได้ การดำเนินงานเริ่มที่การคำนวณค่านิพจน์ในวงเล็บ ถ้าผลลัพธ์มีค่าไม่เป็นศูนย์ ก็จะดำเนินงานข้อความสั่ง 1 ถ้าผลลัพธ์มีค่าเป็นศูนย์และมีส่วน **else** ก็จะดำเนินงานข้อความสั่ง 2

ข้อความสั่ง Switch (Switch Statement) เป็นการเลือกปฏิบัติเมื่อมีกรณีให้เลือกหลายกรณี ซึ่งทำการทดสอบนิพจน์กับค่าคงที่จำนวนเต็ม และแยกไปทำงานให้สอดคล้องตามการทดสอบ โดยมีรูปแบบดังนี้

```
switch (นิพจน์) {
  case นิพจน์ค่าคงที่ : ข้อความสั่ง ;
  case นิพจน์ค่าคงที่ : ข้อความสั่ง ;
  }
```

ข้อความสั่งสร้างวงวน (Loop Constructive Statement)

เป็นข้อความสั่งที่ทำงานเป็นวงวน (Loop) ได้แก่ While For และ Do-while ซึ่งมีรายละเอียดดังนี้

ข้อความสั่ง While (While Statement) เป็นข้อความสั่งทำงานเป็นวงวนที่ทำการประเมินผลค่านิพจน์ก่อน ซึ่งถ้าไม่ใช่ศูนย์ ก็จะดำเนินการข้อความสั่งนั้น จนกระทั่งนิพจน์มีค่าเป็นศูนย์ โดยมีรูปแบบดังนี้

```
while (นิพจน์)
   ข้อความสั่ง;
```

ข้อความสั่ง For (For Statement) เป็นข้อความสั่งทำงานเป็นวงวนที่ทำการประเมินผลค่านิพจน์ 2 ก่อน แต่ข้อความสั่ง For จะมีการกำหนดค่าเริ่มต้นและปรับปรุงค่าเริ่ม

ต้นนี้ด้วยนิพจน์ 1 ที่ส่วนต้นของวงวน เนื่องจากจะได้เห็นส่วนควบคุมง่ายและชัดเจน โดยมีรูปแบบดังนี้

for (นิพจน์ 1; นิพจน์ 2; นิพจน์ 3)
ข้อความสั่ง;

ข้อความสั่ง Do-while (Do-while Statement) เป็นข้อความสั่งทำงานเป็นวงวนที่ทำการประเมินค่านิพจน์ตรงส่วนท้ายของวงวน นั่นคือจะมีการดำเนินงานข้อความสั่งอย่างน้อยหนึ่งครั้งเสมอ โดยมีรูปแบบดังนี้

do
ข้อความสั่ง ;
while (นิพจน์)

ข้อความสั่งกระโดด (Jump Statement)

เป็นข้อความซึ่งสั่งให้เปลี่ยนการควบคุมอย่างไม่มีเงื่อนไข ได้แก่ Goto Break Continue และ Return ซึ่งมีรายละเอียดดังนี้

ข้อความสั่ง Goto (Goto Statement) เป็นข้อความสั่งที่กำหนดให้กระโดดไปยังที่หมาย โดยตำแหน่งของการกระโดดเป็นชื่อเช่นเดียวกับตัวแปรเพียงแต่ต้องตามด้วยเครื่องหมาย ":" ชื่อตำแหน่งการกระโดดนี้ใช้นำหน้าข้อความสั่งใดก็ได้ในฟังก์ชันเดียวกันกับที่มีใช้ข้อความสั่ง Goto ขอบเขตของชื่อนี้จำกัดอยู่ภายในฟังก์ชันที่มันอยู่ โดยมีรูปแบบดังนี้

goto Label1;
Label1: ข้อความสั่ง ;

ข้อความสั่ง Break (Break Statement) เป็นข้อความสั่งที่ใช้ได้เฉพาะในข้อความสั่งสร้างวงวน หรือข้อความสั่ง Switch เท่านั้น โดยจะมีผลให้จบการทำงานของข้อความสั่งในสุดที่ครอบมันอยู่ การควบคุมจะถูกส่งไปยังข้อความสั่งหลังข้อความสั่งวงวน หรือข้อความสั่ง Switch ที่ครอบมันอยู่ โดยมีรูปแบบในแต่ละข้อความสั่ง ดังนี้

```
while (นิพจน์) {
  ข้อความสั่ง 1;
  if (นิพจน์)
    break;
  ข้อความสั่ง 2
}
```

```
switch (นิพจน์) {
   case นิพจน์ค่าคงที่ : ข้อความสั่ง; break;
   case นิพจน์ค่าคงที่ : ข้อความสั่ง; break;
}
```

ข้อความสั่ง Continue (Continue Statement) เป็นข้อความสั่งที่ใช้ได้เฉพาะภายในข้อความสั่งวงวนเท่านั้น ข้อความสั่งนี้มีผลให้ข้อความสั่ง For While หรือ Do ที่ครอบคลุมมันอยู่ทำงานรอบต่อไป ในกรณีของ While และ Do ก็คือดำเนินการในส่วนทดสอบ ส่วนในกรณี For นั้น จะเริ่มทำงานที่ขั้นตอนปรับปรุงค่า โดยมีรูปแบบดังนี้

```
while (นิพจน์) {
    ข้อความสั่ง ;
    continue;
}
```

```
do {
    ข้อความสั่ง;
    continue;
} while (นิพจน์)
```

```
for (นิพจน์ 1; นิพจน์2; นิพจน์ 3)
    { ข้อความสั่ง;
      continue;
    }
```

จากข้อความสั่งกระทำการในภาษาซีดังกล่าวข้างต้น ในวิทยานิพนธ์ฉบับนี้ได้มีการขยายรูปแบบของข้อความสั่งดังต่อไปนี้

1. ข้อความสั่ง If-Else
2. ข้อความสั่ง For
3. ข้อความสั่ง Switch
4. ข้อความสั่ง While

หมายเหตุ ข้อความที่แสดงเป็นตัวเข้มในรูปแบบข้อความสั่ง จะแสดงสีบนจอภาพ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย