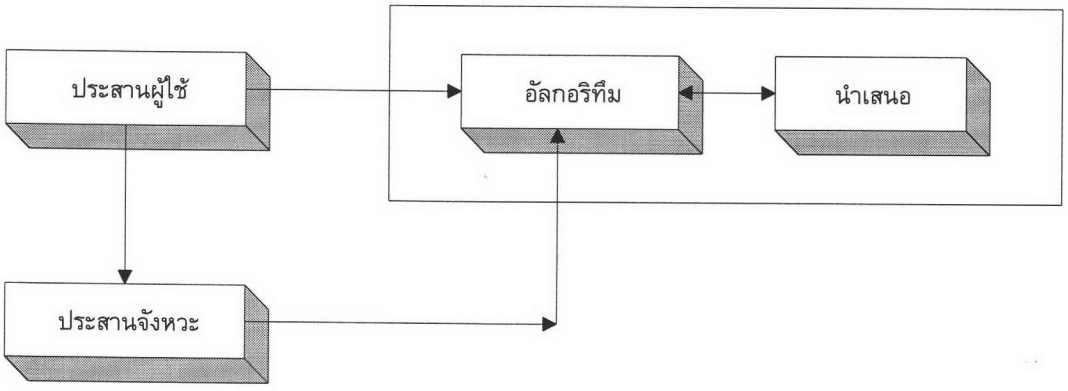
บทที่ 4

การออกแบบระบบจินตทัศน์อัลกอริทึมค้นหาข้อมูล

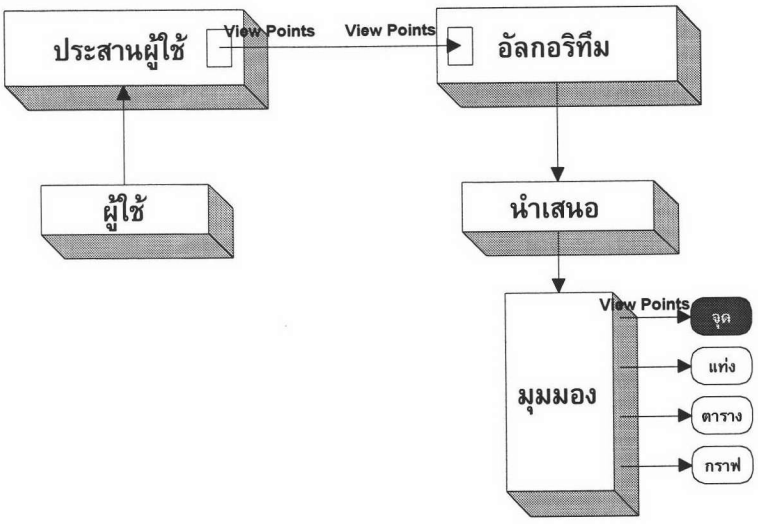
ส่วนประกอบของระบบจินตทัศน์อัลกอริทึมค้นหาข้อมูล

การพัฒนาระบบจินตทัศน์อัลกอริทึมได้แบ่งการทำงานออกเป็นสองโปรแกรมหลัก คือ โปรแกรมควบคุมที่ใช้เป็นส่วนประสานกับผู้ใช้ (User Interface) เพื่อให้ผู้ใช้เป็นผู้กำหนดหรือปรับปรุงสภาพแวดล้อมการทำงานของอัลกอริทึมได้เอง เช่นการกำหนดข้อมูล การเริ่มหรือจบการทำงาน การเร่งหรือหน่วงการทำงาน เป็นต้น และโปรแกรมอัลกอริทึมเป็นที่รวบรวมโปรแกรมย่อยอัลกอริทึมค้นหาข้อมูล 2 โปรแกรมย่อย และการหาตำแหน่งแบบแฮช 8โปรแกรมย่อยไว้ นอกเหนือจากโปรแกรมย่อยการทำงานของอัลกอริทึมแล้วในแต่ละโปรแกรมหลักก็จะมีโปรแกรมย่อยอื่นๆอีกหลายส่วนเพื่อให้การทำงานเป็นไปด้วยความถูกต้อง สามารถถ่ายทอดสถานะให้แสดงเป็นภาพที่สอดคล้องกับการทำงานของอัลกอริทึม ส่วนประกอบหลักของโปรแกรมทั้งสองสามารถแสดงเป็นภาพให้เห็นได้ดังภาพที่ 4.1 ซึ่งแสดงให้เห็นถึงการส่งข่าวสารถึงกันของโปรแกรม



รูปที่ 4.1 ส่วนประกอบหลักระบบจินตทัศน์อัลกอริทึมค้นหาข้อมูล

จากรูปที่ 4.1 จะเห็นว่าโปรแกรมที่เป็นส่วนติดต่อผู้ใช้กับโปรแกรมอัลกอริทึมจะมีการติดต่อกัน ในการติดต่อกันนั้นจะใช้คุณสมบัติของวินโดวส์ประการหนึ่งคือการแลกเปลี่ยนข้อมูลแบบพลวัต ดังที่กล่าวไว้แล้วก่อนหน้านี้ ตัวอย่างต่อไปนี้จะแสดงให้เห็นลักษณะของการกำหนดการทำงานของส่วนประสานผู้ใช้เมื่อต้องการให้ส่วนอัลกอริทึมทำการวาดภาพใหม่ให้เป็นสถานะเริ่มต้น (Reset View)



รูปที่ 4.2 การแลกเปลี่ยนข้อมูลแบบพลวัตให้นำเสนอมุมมองแบบจุด

จากภาพเมื่อผู้ใช้กำหนดผ่านทางส่วนติดต่อผู้ใช้ให้ทำการวาดมุมมองเป็นแบบจุด ส่วนติดต่อผู้ใช้จะแปลงการกำหนดของผู้ใช้เป็นข่าวสารที่มีชื่อกำกับว่า "View_Points" ข่าวสารนี้จะถูกนำไปใส่ในช่องทางแลกเปลี่ยนข้อมูลแบบพลวัต ทางส่วนอัลกอริทึมก็จะรับทราบข่าวสารได้ทันทีและเมื่อทำการแปลความหมายก็จะไปทำการวาดภาพมุมมองเป็นแบบจุด ในส่วนข่าวสารที่ผู้ใช้สามารถกำหนดโดยมีความหมายและอธิบายด้วยชื่อกำกับและค่าคงที่ได้ดังตารางที่ 4.1

ค่าคงที่		ความหมาย
REGENERATE	"#10"	เมื่อมีการเปลี่ยนข้อมูลขาเข้าซึ่งอาจเป็นเปลี่ยนค่ามากสุด/น้อยสุดของขอบเขตข้อมูล จำนวนข้อมูล ตำแหน่งในหาข้อมูล ฯลฯ เมื่อส่วนนำเสนอรับข่าวสารการเปลี่ยนแปลงข้อมูลข้าเข้าก็จะทำการอ่านข้อมูลขาเข้าใหม่เข้าทำงาน พร้อมปรับปรุงภาพในมุมมองต่างๆให้สอดคล้องกับข้อมูลชุดใหม่

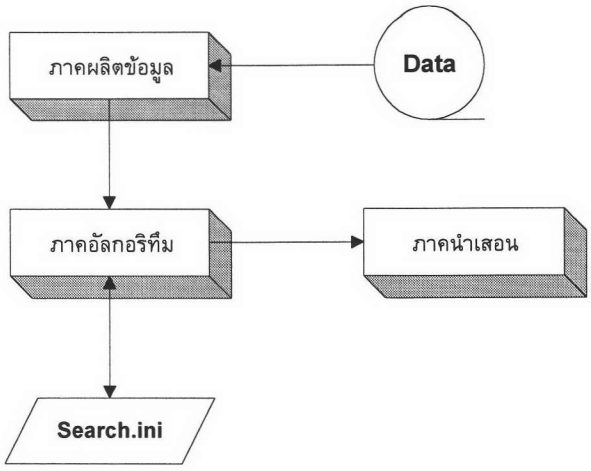
RESET_VIEW	"#11"	ให้มีการลบภาพที่นำเสนอเสร็จเรียบร้อยแล้ว และทำการวาดภาพเป็นสถานะเริ่มต้นเพื่อพร้อมสำหรับการทำงานรอบต่อไป
REPAINT_VIEW	"#12"	สร้างภาพมุมมองใหม่อีกหน้าต่างหนึ่งตามพารามิเตอร์ที่ได้รับว่าต้องทำการวาดมุมมองเป็นแบบใดซึ่งมี แบบจุด แบบแท่ง แบบตาราง แบบกราฟ
RESET_COLOR	"#13"	ให้ทำการอ่านค่าจากแฟ้มข้อมูลทำงานการแลกเปลี่ยนข้อมูล Application Program Interface, API ในส่วนของการกำหนดสีในส่วนต่างๆ ลบภาพมุมมองทุกภาพแล้ววาดใหม่ให้มีสีตามข่าวสารที่ได้รับใหม่
CHANGE_KEY	"#14"	ผู้ใช้ระบุตำแหน่งของข้อมูลที่ต้องการหาใหม่ ส่งสัญญาณบอกทางส่วนควบคุมว่ามีการเปลี่ยนตำแหน่งหรือค่าข้อมูลที่ต้องการหาแล้วให้ส่วนควบคุมอ่านข่าวสารได้จากแฟ้มข้อมูลทำงานบันทึกลงแฟ้มข้อมูลขาเข้า แล้วส่งข่าวสารให้ส่วนนำเสนอทราบโดยส่งเป็นค่าคงที่ "#10"
REPAINT_GRAPH	"#15"	เมื่ออัลกอริทึมการหาที่อยู่แบบแฮชทำงานเสร็จแล้วมีการนำเสนอในมุมมองกราฟจะส่งข่าวสารกลับมาให้ส่วนควบคุมซึ่งข่าวสารส่วนหนึ่งจะเป็นการบอกถึงค่ามากสุดของมาตราส่วนแกนตั้งส่วนควบคุมจะบันทึกค่าที่มากที่สุดลงในแฟ้มข้อมูลทำงาน เมื่อผู้ใช้เรียกหน้าต่างสรุปการดำเนินการของแต่ละอัลกอริทึมขึ้นมาดู และกดปุ่มการปรับปรุงมุมมองกราฟ ส่วนนำเสนอจะอ่านค่าจากแฟ้มข้อมูลทำงานในส่วนของมาตราส่วนแล้วทำการวาดกราฟใหม่ จะได้ทุกมุมมองที่เป็นกราฟมีมาตราส่วนแกนตั้งเหมือนกัน

WINDOW_CASCADE	"#20"	สั่งการให้ภาคนำเสนอทำการจัดระเบียบมุมมองต่างๆให้ซ้อนกัน
WINDOW_TILE	"#21"	สั่งการให้ภาคนำเสนอทำการจัดระเบียบมุมมองต่างๆโดยให้เห็นมุมมองทั้งหมด
RUN_PLAY	"#30"	กำหนดการทำงานของภาคประสานจังหวะให้หยุดการให้จังหวะชั่วคราว
RUN_PAUSE	"#31"	กำหนดการทำงานของภาคประสานจังหวะให้จังหวะทีละหนึ่งจังหวะ
RUN_NEXT	"#32"	กำหนดการทำงานของภาคประสานจังหวะให้ทำงานต่อไปหลักจากกำหนดให้หยุดการให้จังหวะชั่วคราว "#33"
RUN_STOP	"#33"	กำหนดการทำงานของภาคประสานจังหวะให้หยุดการให้จังหวะ
VIEW_POINTS	"#40"	กำหนดให้ภาคนำเสนอ ทำการสร้างมุมมองแบบจุด
VIEW_STICK_BAR	"#41"	กำหนดให้ภาคนำเสนอ ทำการสร้างมุมมองแบบแท่ง
VIEW_TABLE	"#43"	กำหนดให้ภาคนำเสนอ ทำการสร้างมุมมองแบบตาราง
VIEW_HASH_GRAPH	"#44"	กำหนดให้ภาคนำเสนอ ทำการสร้างมุมมองแบบกราฟ
APPL_TERMINATE	"#99"	สั่งการให้ภาคอัลกอริทึมจบการทำงาน

ตารางที่ 4.1 แสดงการกำหนดค่าคงที่ใช้ในการส่งข้อมูลแบบพลวัต

ภาคจินตทัศน์อัลกอริทึม

ภาคจินตทัศน์อัลกอริทึมจะประกอบไปด้วยส่วนการทำงานของอัลกอริทึม ส่วนที่ผลิตข้อมูลขาเข้า และส่วนที่จะเปลี่ยนแปลงภาพตามการทำงานของอัลกอริทึมแสดงตามรูปที่ 4.3 ในแต่ละส่วนจะมีการทำงานดังนี้



รูปที่ 4.3 แสดงส่วนประกอบภาคจินตทัศน์อัลกอริทึม

1. ภาคอัลกอริทึม

คือโปรแกรมส่วนย่อยที่แสดงขั้นตอนการทำงานของค้นหาข้อมูล และการหาที่อยู่แบบแฮชแต่ละแบบ ในการพัฒนาระบบจินตทัศน์อัลกอริทึมค้นหาข้อมูลนี้นอกจากการที่เราจะต้องศึกษาการทำงานของอัลกอริทึมแต่ละแบบแล้ว จะต้องพิจารณาด้วยว่าตรงจุดในของขั้นตอนการทำงานของอัลอกริทึมที่จะเป็นจุดของการเปลี่ยนแปลงภาพการทำงานได้อย่างชัดเจนและผู้ใช้เกิดความเข้าใจ จากการศึกษาพบว่าจุดที่มีการเลื่อนตำแหน่งของตัวชี้ไปตามข้อมูลที่เปรียบเทียบจะสามารถแสดงคุณสมบัติดังกล่าวได้ชัดเจนที่สุด

รูปที่ 4.4 แสดงอัลกอริทึมค้นหาข้อมูลแบบทวิภาคโดยใช้ภาษาวิชวลเบสิก ในบรรทัดที่ 2 คือจุดที่เป็นตำแหน่งชี้ข้อมูลที่จะนำมาเปรียบเทียบกับข้อมูลที่ต้องการ และจะเป็นตำแหน่งของการเปลี่ยนแปลงภาพ ฟังก์ชัน Output_Search_Update เป็นโปรแกรมย่อยที่จะนำพารามิเตอร์ที่ได้ไปทำการเปลี่ยนแปลง ณ ตำแหน่งของข้อมูลที่ระบุ เมื่อทำการตรวจสอบค่าข้อมูลที่ตำแหน่งชี้กับข้อมูลที่ต้องการ ผลที่ได้จะเป็นหนึ่งในสามลักษณะคือ เท่ากัน มากกว่าน้อยกว่าซึ่งก็จะขึ้นกับอัลกอริทึมด้วยว่าผลที่ได้แต่ละแบบจะเปลี่ยนภาพอย่างไร กรณีของอัลกอริทึมค้นหาข้อมูลแบบทวิภาค ถ้าเท่ากันก็จะได้ตำแหน่งที่พบแล้วออกจากวงวนการทำงานเพื่อไปแสดงภาพหรือข่าวสารบางอย่างเพื่อบอกให้ผู้ใช้ทราบว่าพบข้อมูลที่ต้องการ ถ้าค่าตรงตำแหน่งชี้มากกว่าค่าข้อมูลที่ต้องการ แสดงว่าข้อมูลจะอยู่ในส่วนครึ่งแรก ในส่วนครึ่งหลังก็ไม่ต้องทำการตรวจสอบอีก (เมื่อข้อมูลมีการจัดเรียงแบบน้อยไปมาก) ถ้าค่าตรงตำแหน่งชี้น้อยกว่าค่าข้อมูลที่ต้องการก็จะให้ผลในทางกลับกัน ก็จะต้องนำวิธีดังกล่าวของอัลกอริทึมเปลี่ยนภาพให้ผู้ใช้ทราบว่าข้อมูลในส่วนใดที่ไม่ต้องทำการเปรียบเทียบ และส่วนใดที่ต้องทำการเปรียบเทียบในรอบต่อไป และจากรูปที่ 4.4 ซึ่งแสดงโปรแกรมการทำงานจะเห็นว่ามีการบันทึกรอบของการทำงานไว้ด้วยในบรรทัดที่ 3 ซึ่งรอบของการทำงานก็จะหมายถึงจำนวนครั้งในการเปรียบเทียบ

นอกจากนั้นระบบจินตทัศน์อัลกอริทึมควรจะอนุญาตให้ผู้ใช้ทำการจินตทัศน์ได้หลายๆอัลกอริทึมพร้อมกัน ผู้พัฒนาจะต้องคำนึงถึงจุดนี้เนื่องจากแต่ละอัลกอริทึมจะต้องได้โอกาสในการทำงานเท่าเทียมกันจึงจะทำให้การเปรียบเทียบประสิทธิภาพเป็นไปอย่างถูกต้อง

จากข้อมูลข้างต้นทำให้เราได้ทราบแล้วว่าประสิทธิภาพของอัลกอริทึมอยู่ที่จำนวนครั้งของการเปรียบเทียบ ดังนั้นเมื่ออัลกอริทึมหนึ่งเสร็จสิ้นจากเปรียบเทียบหนึ่งครั้งก็ควรจะหยุดรอให้อัลกอริทึมอื่นๆทำงานจนเสร็จการเปรียบเทียบหนึ่งครั้งเช่นกัน ดังนั้นตัวอย่างในรูปที่ 4.4จะเห็นได้ว่าเมื่ออัลกอริทึมค้นหาข้อมูลแบบทวิภาคได้ทำงานมาจนเสร็จการเปรียบเทียบหนึ่งครั้ง (บรรทัดที่ 5) ก็จะมีการเรียกใช้ฟังก์ชันชื่อ NotifySync( ) ผลของการเรียกใช้ฟังก์ชั่นนี้จะทำให้อัลกอริทึมนี้หยุดการทำงานชั่วคราวเพื่อเปิดโอกาสให้กับอัลกอริทึมอื่นทำงานจนเสร็จขั้นตอนการเปรียบเทียบหนึ่งรอบในทุกอัลกอริทึม (ในอัลกอริทึมอื่นก็จะมีการเรียกใช้ฟังก์ชัน NotifySync( ) เช่นกัน) จึงจะกลับมาทำงานในรอบต่อไป การทำงานลักษณะนี้จึงทำให้เราสามารถประกันได้ว่าทุกๆอัลกอริทึมที่เลือกเข้าทำงาน และกำลังดำเนินการอยู่จะได้รับโอกาสในการทำงานเท่าเทียมกัน สำหรับรายละเอียดของฟังก์ชัน NotifySync( ) นี้จะกล่าวถึงโดยละเอียดในภาคประสานจังหวะ

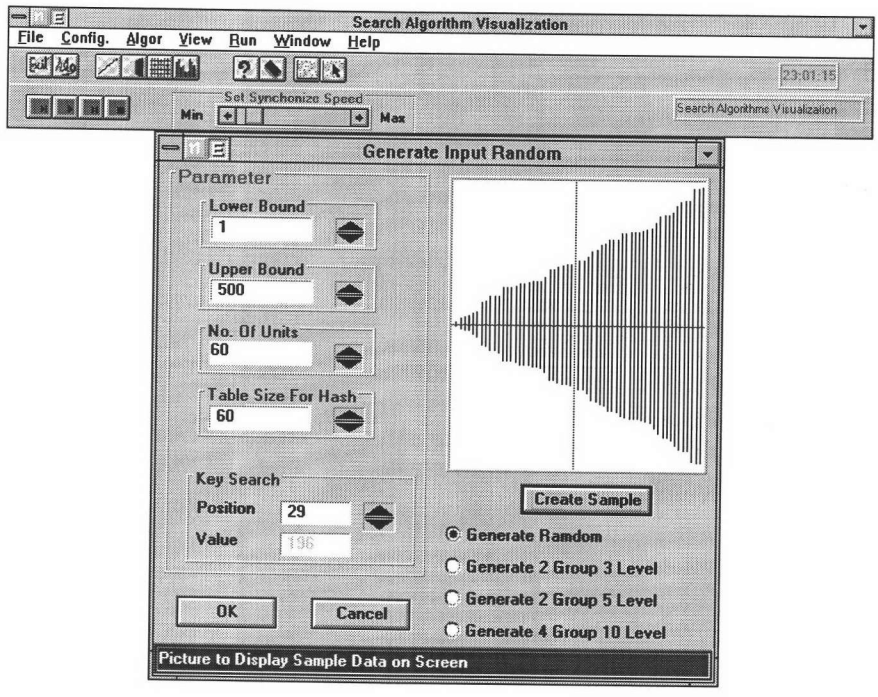
```
01)  Do
02)      Pos_mid = Int((Pos_top + Pos_bottom) / 2)
03)      nCompare = nCompare + 1
04)      Pos_Mark(Pos_mid) = True
05)      Ret_Value = NotifySync(FormMDI.hWnd, procnum)
06)      Call Output_Search_Update(Pos_mid, Pos_mid, Pos_mid, FoundSearched)
07)      If KeySearch = Data_Ord(Pos_mid) Then
08)          Found_Pos = Pos_mid
09)          Ret_Value = NotifySync(FormMDI.hWnd, procnum)
10)          Call Output_Search_Update(Pos_mid, 0, Pos_mid, FoundSearched)
11)          Exit Do
12)      ElseIf KeySearch < Data_Ord(Pos_mid) Then
13)          Ret_Value = NotifySync(FormMDI.hWnd, procnum)
14)          Call Output_Search_Update(Pos_mid, Pos_top, Pos_mid, AfterSearched)
15)          Pos_top = Pos_mid - 1
16)      Else
17)       Ret_Value = NotifySync(FormMDI.hWnd, procnum)
18)      Call Output_Search_Update(Pos_bottom, Pos_mid, Pos_mid, 1AfterSearched)
```

```
19)        Pos_bottom = Pos_mid + 1
20)      End If
21)  Loop Until (KeySearch = Data_Ord(Pos_mid) Or (Pos_top < Pos_bottom))
```

รูปที่ 4.4 แสดงโปรแกรมอัลกอริทึมค้นหาข้อมูลแบบทวิภาคกับการเปลี่ยนแปลงภาพ

2. ภาคผลิตข้อมูลขาเข้า

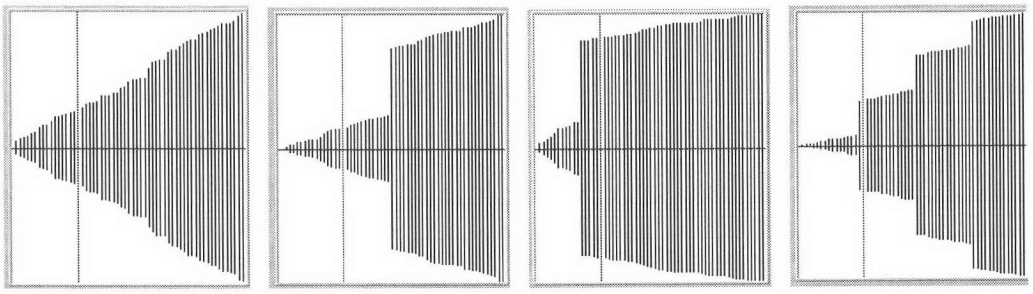
หลังจากที่เลือกอัลกอริทึมแล้ว ผู้ใช้สามารถเลือกภาคผลิตข้อมูลขาเข้าเพื่อทำการระบุลักษณะของข้อมูลขาเข้าที่จะให้กับอัลกอริทึมระหว่างการทำงาน สำหรับโครงงานวิจัยนี้ผู้วิจัยได้ออกแบบให้ผู้ใช้สามารถเลือกลักษณะของข้อมูลได้ 2 แบบ คือ เป็นข้อมูลแบบสุ่มโดยระบบ โดยกำหนดขอบเขตสูงสุด ต่ำสุดของค่าข้อมูล จำนวนข้อมูล ขนาดของตารางแฮช และค่าข้อมูลที่ต้องการหา ดังแสดงในรูปที่ 4.5



รูปที่ 4.5 แสดงหน้าต่างผลิตข้อมูลขาเข้าแบบสุ่ม

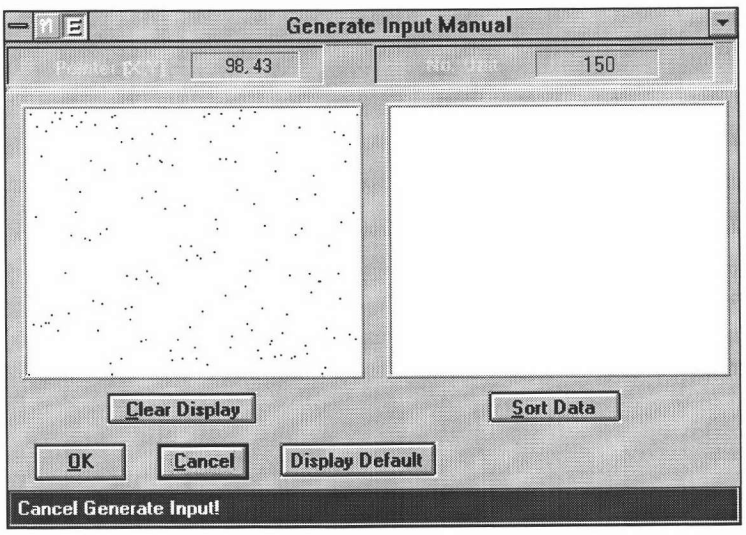
จากรูปเมื่อผู้ใช้กำหนดขอบเขตแล้ว ในการสร้างข้อมูลแบบสุ่มก็ยังระบุได้ว่าต้องการให้ข้อมูลมีการแบ่งกลุ่มด้วยหรือไม่เช่น ไม่มีการแบ่งกลุ่มเลยก็จะเลือกในหัวข้อ Generate Random มีการแบ่งเป็นสองกลุ่มเท่ากันในหัวข้อ Generate 2 Group 3 Level แบ่งเป็นสองกลุ่มไม่เท่ากันกลุ่มแรกมีจำนวนน้อยกว่าอีกกลุ่ม และแบ่งเป็นสี่กลุ่มเท่ากัน ตามรูปที่ 4.6 (a), (b), (c) และ (d)

ตามลำดับ เพื่อใช้ในขั้นตอนทดสอบว่าถ้าลักษณะของข้อมูลมีการเปลี่ยนแปลงจะมีผลต่อประสิทธิภาพการทำงานของอัลกอริทึมแต่ละประเภทหรือไม่



รูปที่ 4.6 แสดงการสร้างข้อมูลแบบสุ่มแต่ละแบบ

อีกวิธีหนึ่งจะให้ผู้ใช้สามารถทำการสร้างรูปแบบของข้อมูลได้อย่างอิสระโดยกดปุ่มเมาส์ด้านซ้ายมือค้างไว้แล้วทำการลากลงในพื้นที่ที่เตรียมไว้ในการสร้างข้อมูลตามรูปที่ 4.7

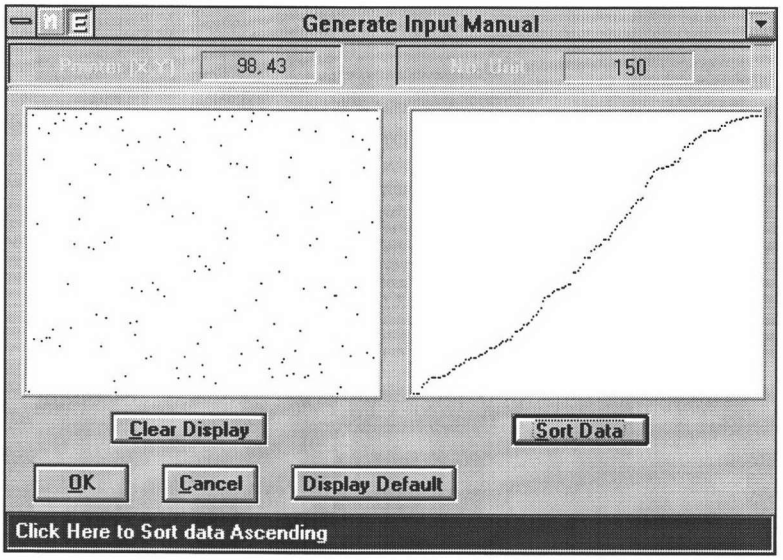


รูปที่ 4.7 แสดงหน้าต่างผลิตข้อมูลขาเข้าโดยผู้ใช้

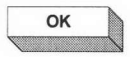
ทางด้านบนซ้ายมือจะมีกรอบข้อความแสดงตำแหน่งของเมาส์ แสดงเป็นค่าของคู่ลำดับ (X,Y) ซึ่ง X หมายถึงตำแหน่งของข้อมูล และ Y หมายถึงค่าของข้อมูล ผู้ใช้สามารถสร้างรูปแบบของข้อมูลได้อย่างหลากหลายเพื่อทดสอบการทำงานของแต่ละอัลกอริทึม หลังจากที่ผู้ใช้สร้างข้อมูลเป็นที่พอใจแล้ว ต้องการดูการจัดเรียงข้อมูลเพื่อดูลักษณะการรวมกลุ่ม หรือกระจายของค่า

ข้อมูลจะเป็นอย่างไร ก็ให้กดปุ่มสั่งงาน

ข้อมูลที่สร้างในกรอบทางด้านซ้ายจะถูกจัดเรียงบลำดับ (Sort) ในลักษณะจากน้อยไปมาก (Ascending) ตามรูปที่ 4.8



รูปที่ 4.8 แสดงข้อมูลผู้ใช้กำหนดเอง และทำการจัดเรียง

เมื่อผู้ใช้ยืนยันการสร้างข้อมูลโดยกดปุ่มสั่งงาน 

ระบบจะทำการบันทึกข้อมูลลงในแฟ้มข้อมูลโดยผู้วิจัยได้กำหนดให้ชื่อของแฟ้มข้อมูลที่เก็บข้อมูลขาเข้าเป็น 2 แฟ้มข้อมูล แฟ้มแรกจัดเก็บข้อมูลแบบจัดเรียงลำดับชื่อ "DATAORD.DAT" แฟ้มที่สองเป็นแฟ้มที่จัดเก็บข้อมูลที่สร้างจากการสุ่มของระบบกรณีเลือกสุ่มข้อมูลโดยระบบ หรือได้จากผู้ใช้กำหนดเองบันทึกลงในแฟ้มข้อมูลชื่อ "DATARND.DAT" สำหรับการจัดเก็บแฟ้มข้อมูลขาเข้านี้ผู้วิจัยได้เลือกเป็นแฟ้มข้อมูลลำดับ (sequential file) เพื่อให้ง่ายต่อการใช้งานและผู้ใช้สามารถเรียกแฟ้มข้อมูลนี้ขึ้นมาดูได้โดยโปรแกรมบรรณาธิการ (Editor) ที่มีอยู่ทั่วไป หลังจากที่ผู้ใช้ได้ทำการสร้างข้อมูลขาเข้าแล้วและทำการยืนยันการสร้างข้อมูลใหม่ ก็จะมีการส่งข่าวสารบอกให้แก่ภาคอัลกอริทึมทราบ เพื่อให้ภาคอัลกอริทึมทำการดึงข้อมูลชุดใหม่ไปใช้พร้อมกับทำการเปลี่ยนแปลงมุมมองของตัวแทนข้อมูลใหม่ด้วย

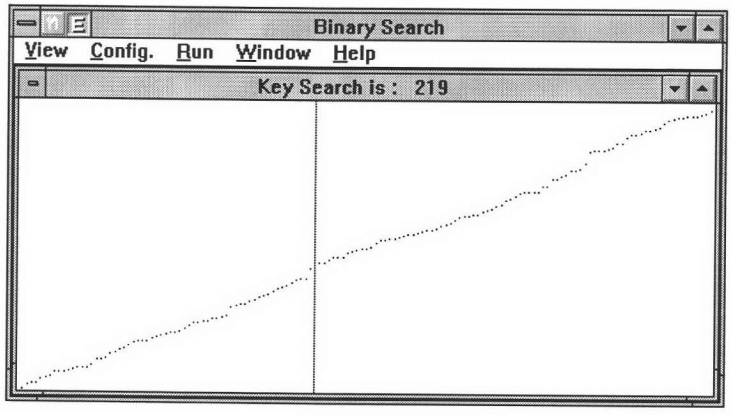
3. ภาคนำเสนอ

หลังจากที่ผู้ใช้ได้ทำตามหัวข้อ 1 และ 2 แล้วก็มาถึงส่วนที่ระบบจะนำข้อมูลมาแทนด้วยตัวแทนข้อมูลแบบใด ซึ่งผู้ใช้จะเป็นผู้กำหนดโดยเรียกการแสดงตัวแทนข้อมูลว่ามุมมอง (View) สำหรับในระบบจินตทัศน์อัลกอริทึมค้นหาข้อมูล จะนำเสนอมุมมองสองชนิดคือ มุมมอง

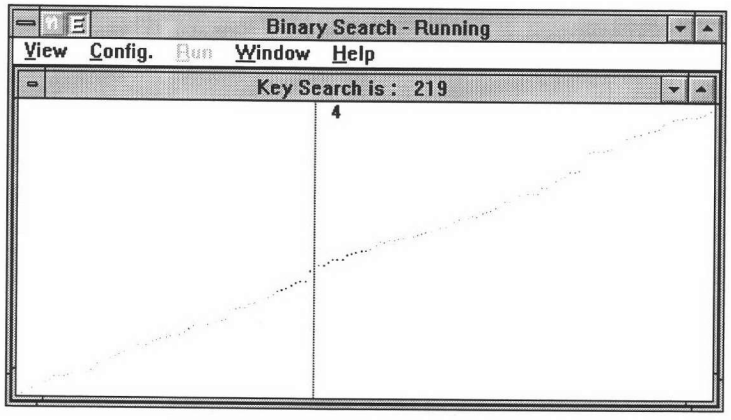
แบบจุด มุมมองแบบแท่ง ส่วนอัลกอริทึมการหาตำแหน่งที่อยู่แบบแฮชก็จะนำเสนอในสองมุมมองเช่นกันคือ มุมมองแบบตาราง และมุมมองแบบกราฟ โดยมีรายละเอียดดังนี้

3.1 มุมมองแบบจุด (Points View)

ในการนำเสนอมุมมองแบบจุดมักจะใช้เมื่อมีการสร้างข้อมูลขาเข้าเป็นจำนวนมาก ตัวอย่างการนำเสนอมุมมองแบบจุดได้แสดงไว้ในรูปที่ 4.9 ค่าข้อมูลแต่ละค่าจะถูกแทนด้วยจุด แต่ละจุดในมุมมองจะเกิดจากคู่ลำดับ (x,y) โดยค่า x จะแทนตำแหน่งของข้อมูลและ y จะแทนค่าของข้อมูล ในรูปที่ 4.10 แสดงการเปลี่ยนแปลงของมุมมองในขณะที่อัลกอริทึมค้นหาข้อมูลแบบทวิภาคกำลังทำงาน



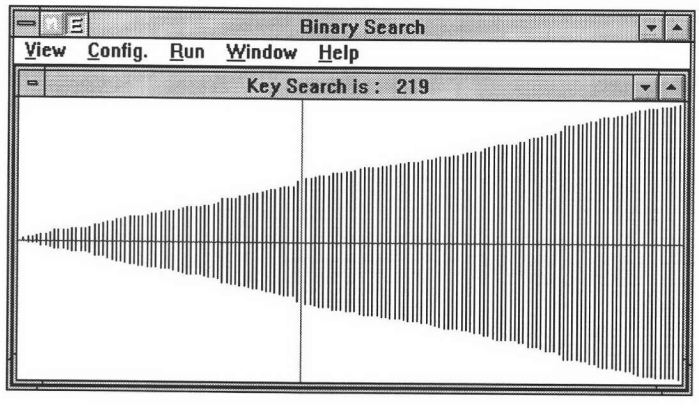
รูปที่ 4.9 มุมมองแบบจุดตัวแทนข้อมูลก่อนค้นหาข้อมูล



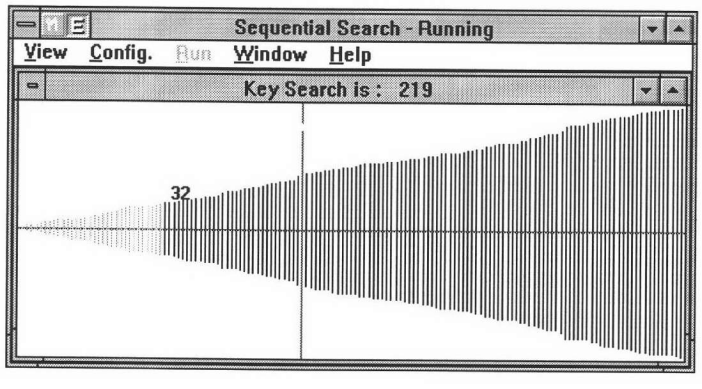
รูปที่ 4.10 แสดงการเปลี่ยนแปลงขณะค้นหาข้อมูลแบบทวิภาคในมุมมองแบบจุด

3.2 มุมมองแบบแท่ง (Stick Bars View)

มุมมองแบบแท่งเป็นอีกมุมมองหนึ่งที่นิยมใช้ในการนำเสนอการทำงานของอัลกอริทึมค้นหาข้อมูล เพราะสามารถเห็นลักษณะข้อมูลได้ดีทั้งการจัดเรียงและความแตกต่างของค่าข้อมูล ดังแสดงในรูปที่ 4.11 ความสูงของแท่งข้อมูลแต่ละแท่งจะแทนค่าของข้อมูลแท่งที่มีความสูงมากจะแทนข้อมูลที่มีค่ามากกว่าแท่งที่มีความสูงน้อย แท่งหนึ่งแท่งจะแทนค่าของข้อมูลหนึ่งค่าและตำแหน่งของแท่งในมุมมองก็จะแทนตำแหน่งของข้อมูล เมื่ออัลกอริทึมทำงานตามรูปที่ 4.12 ก็จะเห็นการเปลี่ยนแปลงมุมมอง



รูปที่ 4.11 แสดงมุมมองแบบแท่งก่อนค้นหาข้อมูล

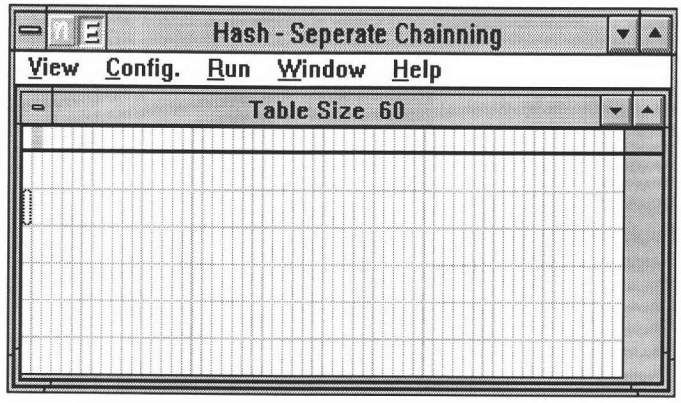


รูปที่ 4.12 แสดงการเปลี่ยนแปลงขณะค้นหาข้อมูลแบบลำดับในมุมมองแบบแท่ง

3.3 มุมมองแบบตาราง (Table View)

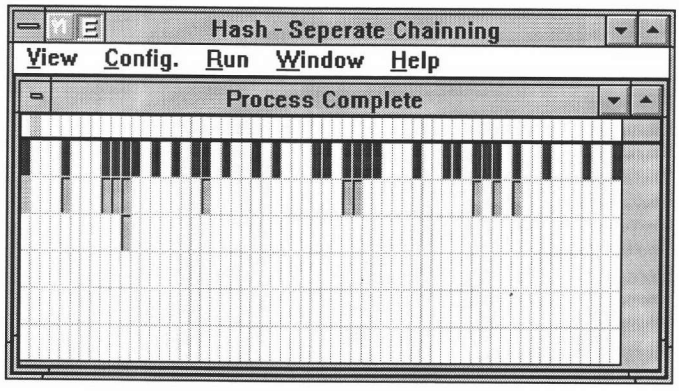
อัลกอริทึมการหาที่อยู่แบบแฮชจะมีลักษณะการทำงานโดยนำเขตข้อมูลหลักของแต่ละระเบียนมาทำการคำนวนเพื่อให้ได้ตำแหน่งในตารางที่เตรียมไว้ เพื่อเป็นการจำลองตามหลักการดังกล่าว การนำเสนอมุมมองแบบตารางจะทำให้ผู้ใช้เห็นการทำงานของอัลกอริทึมคล้ายสภาพจริง

I17509769



รูปที่ 4.13 แสดงมุมมองแบบตาราง

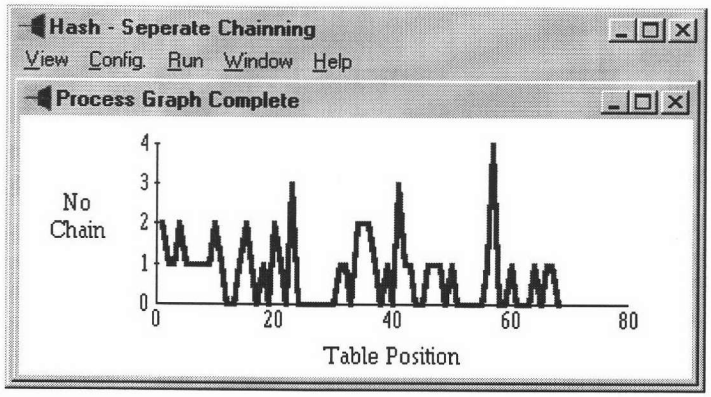
จากรูปที่ 4.13 จะแบ่งพื้นที่ออกเป็นแถวและสดมภ์ ให้มีจำนวนสดมภ์เท่ากับขนาดของตาราง (TableSize) ส่วนจำนวนแถวจะกำหนดให้เป็น 10 % ของข้อมูลทั้งหมด แต่เมื่อเริ่มทำงานครั้งแรกจะกำหนดขนาดไม่เกิน 10 แถวเมื่ออัลกอริทึมทำงานจำนวนแถวอาจมีการขยายเพิ่มมากขึ้นได้ตามความจำเป็น โดยการเปลี่ยนแปลงภาพกำหนดให้มีตารางแถวเดียวอยู่บนตารางของแฮชจะเรียกว่าตารางระบุตำแหน่งทำงาน ให้มีขนาดของสดมภ์ที่เท่ากันพอดี เมื่อได้ตำแหน่งจากฟังก์ชันแฮช นำค่าตำแหน่งบันทึกที่ตารางระบุตำแหน่งก่อนโดยนำเข้ารูปแบบของบิต (Bitmap) ที่สร้างไว้ก่อนแล้วซึ่งมี 3 แฟ้มข้อมูลคือ btrue.bmp เก็บรูปแบบของบิตที่เป็นสีเข้มเพื่อแสดงกรณีการหาตำแหน่งได้ตำแหน่งว่าง bprob.bmp เมื่อเกิดการซ้ำตำแหน่งแต่ค่าข้อมูลเท่ากัน และ bcoll.bmp เมื่อเกิดการซ้ำตำแหน่ง โดยในตารางระบุตำแหน่งจะนำเข้าเพียงแฟ้ม btrue.bmp เท่านั้นเพื่อให้เห็นชัดว่ากำลังทำสดมภ์อะไรอยู่ จากนั้นค่อยทำการตรวจสอบตัวบ่งชี้ (Flag) ที่กำหนดการซ้ำตำแหน่ง เพื่อนำเข้ารูปแบบของบิตที่ถูกต้องลงในตารางแฮชซึ่งอยู่ด้านล่างของตารางระบุตำแหน่งตามรูปที่ 4.14



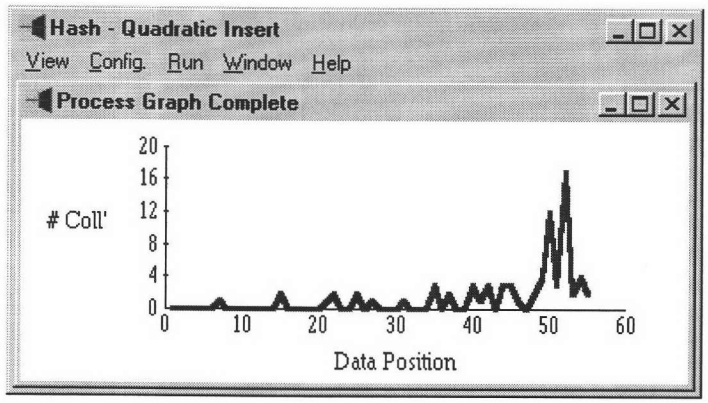
รูปที่ 4.14 แสดงการนำเข้ารูปแบบของบิตในแต่ละสดมภ์ในมุมมองแบบตาราง

3.4 มุมมองแบบกราฟ (Graph View)

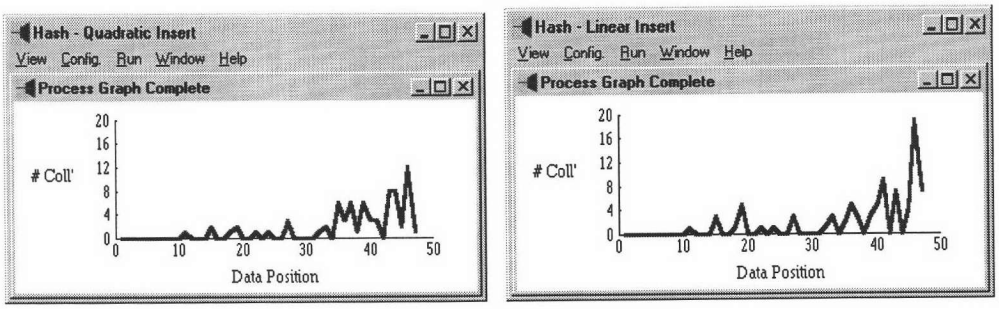
จะใช้มุมมมองแบบกราฟกับการทำงานของอัลกอริทึมการหาตำแหน่งที่อยู่แบบแฮชเท่านั้น เพื่อให้เห็นจำนวนครั้งของการชนกันของแต่ละตำแหน่งว่าตำแหน่งใด หรือบริเวณใดมีการชนกันมาก ทั้งนี้ความหมายของกราฟจะถูกแบ่งออกเป็นสองกรณีคือเมื่อมุมมองกราฟแสดงการทำงานของอัลกอริทึมแบบไม่หาที่อยู่ใหม่ และแบบรายการโยง ค่าในแนวนอนจะหมายถึงตำแหน่งในตารางแฮช ส่วนแนวตั้งจะหมายถึงจำนวนครั้งที่มีการชนกันของตำแหน่งในตารางแฮชนั้นๆตามรูปที่ 4.15 ส่วนอีกกรณีคือเมื่อมุมมองกราฟแสดงการทำงานของฟังก์ชันแฮชในแบบอื่น ค่าในแนวนอนจะหมายถึงตำแหน่งของข้อมูลที่เข้าทำงาน ส่วนแนวตั้งจะหมายถึงจำนวนครั้งของการชนก่อนที่ค่าของข้อมูลจะหาตำแหน่งที่อยู่ได้ดังรูปที่ 4.16 และเมื่อมีการทำงานหลายอัลกอริทึมสเกลแกนตั้งอาจมีค่าไม่เท่ากันขึ้นอยู่กับจำนวนการชนที่มากที่สุดของแต่ละข้อมูล เพื่อให้การดูการเปรียบเทียบในรูปกราฟเห็นความแตกต่างได้อย่างชัดเจนเมื่ออัลกอริทึมทุกอัลกอริทึมทำงานเสร็จจะทำการปรับสเกลแกนตั้งให้เท่ากันตามรูปที่ 4.17



รูปที่ 4.15 มุมมองแบบกราฟแสดงอัลกอริทึมแฮชแบบรายการโยง



รูปที่ 4.16 มุมมองแบบกราฟแสดงอัลกอริทึมแฮชแบบสมการกำลังสอง



รูปที่ 4.17 แสดงการปรับสเกลแนวตั้งเมื่อจบการทำงานอัลกอริทึม

ภาคประสานจังหวะ

ในการทำงานที่ต้องมีหลายอัลกอริทึมทำงานพร้อมๆกันเพื่อดูประสิทธิภาพการทำงาน สิ่งที่สำคัญคือจะทำอย่างไรให้แต่ละงานได้รับโอกาสในการทำงานที่เท่าเทียมกัน ในการทำงานของอัลกอริทึมที่มีการทำงานในแบบวงวน จุดที่เป็นจุดที่เหมาะสำหรับการให้โอกาสหรือหยุดการให้โอกาสก็คือรอบหนึ่งของวงวน เปรียบเหมือนกับการให้สัญญาณของไวทยากรในการควบคุมการบรรเลงดนตรี หรือตำรวจจราจรตามแยกที่คอยให้สัญญาณหยุดหรือไปได้ของแต่ละทางเพื่อให้ได้โอกาสเท่าๆกันไม่ยื้อแย่งหรือครอบครองเพียงด้านเดียว สำหรับโครงงานวิจัยนี้มีการเรียกใช้ฟังก์ชันการประสานจังหวะโดยผ่านทางคลังโปรแกรมเชื่อมโยงแบบพลวัต ซึ่งเป็นคลังโปรแกรมที่เก็บฟังก์ชันต่างๆ โดยคลังโปรแกรมดังกล่าวมีชื่อว่า "SYNCLIB.DLL" ภายในจะประกอบด้วยฟังก์ชัน ซึ่งสามารถจัดกลุ่มตามหน้าที่การทำงานได้เป็น 2 กลุ่มดังนี้คือ ฟังก์ชันสำหรับผู้ควบคุมหรือผู้ให้สัญญาณประสานจังหวะ และฟังก์ชันสำหรับผู้ถูกควบคุมหรือผู้รับสัญญาณประสานจังหวะ มีรายละเอียดดังนี้

1. ผู้ควบคุมหรือผู้ให้สัญญาณประสานจังหวะ ประกอบด้วยฟังก์ชันย่อย

1.1 “ConfigSync” เป็นการแจ้งให้ผู้ถูกควบคุมทราบว่าจะมีการให้จังหวะเป็นอย่างไร

1.2 “NextSync” ใช้เพื่อการส่งสัญญาณการประสานจังหวะให้ผู้ถูกควบคุม

1.3 “DropSync” ใช้การบอกยกเลิกการให้บริการการประสานจังหวะ

รายละเอียดของการประกาศฟังกชัน และความหมายของพารามิเตอร์แต่ละตัวศึกษาได้จากเอกสารคลังข้อมูลประสานจังหวะ SYNCDLL.TXT ตัวอย่างการนำฟังก์ชันของผู้ควบคุมมาใช้สามารถแสดงได้ดังนี้

```
Sub mnuRunStart_Click ( )
        Call ConfigSync(1, 0, -1)
        Ret_Value = NextSync()
End Sub
```

2. ผู้ถูกควบคุมหรือผู้รับสัญญาณประสานจังหวะ จะมีการใช้ฟังก์ชันของคลังโปรแกรมประสานจังหวะดังนี้

2.1 "RegisterSync" การขอลงทะเบียนเพื่อรับสัญญาณการประสานจังหวะ

2.2 "NotifySync" การรับสัญญาณการประสานจังหวะ ในส่วนของฟังก์ชันนี้จะใช้ในการวัดประสิทธิภาพการทำงานของอัลกอริทึมกล่าวคือ ในการวัดประสิทธิภาพการทำงานของอัลกอริทึมแต่ละชนิดมักจะใช้จำนวนครั้งของการดำเนินการขั้นพื้นฐาน (สมชาย ประสิทธิ์จูตระกูล,1993) เป็นตัววัด อัลกอริทึมใดที่มีการทำงานขั้นพื้นฐานน้อยกว่าอีกอัลกอริทึมหนึ่งย่อมแสดงได้ว่าอัลกอริทึมนั้นมีประสิทธิภาพมากกว่า สำหรับปัญหาในเรื่องค้นหาข้อมูลจะใช้การเปรียบเทียบค่าของข้อมูลเป็นการดำเนินการขั้นพื้นฐาน เพื่อให้เกิดความเสมอภาคในการวัดประสิทธิภาพการทำงานของอัลกอริทึม อัลกอริทึมใดทำงานมาจนถึงการดำเนินการขั้นพื้นฐานจะต้องหยุดการทำงานชั่วคราวเพื่อให้โอกาสแก่อัลกอริทึมอื่นๆได้ทำงานจนถึงการดำเนินการขั้นพื้นฐานของตนเองจึงหยุดเช่นกัน การทำงานจึงเป็นไปในลักษณะสลับไปมาจนเสร็จครบทุกอัลกอริทึม ทำเช่นนี้ได้อัลกอริทึมต้องมีการเรียกใช้ฟังก์ชันการรอรับสัญญาณ "NotifySunc" จะเป็นการหยุดการทำงานชั่วคราวแล้วรอรับสัญญาณในรอบต่อไป

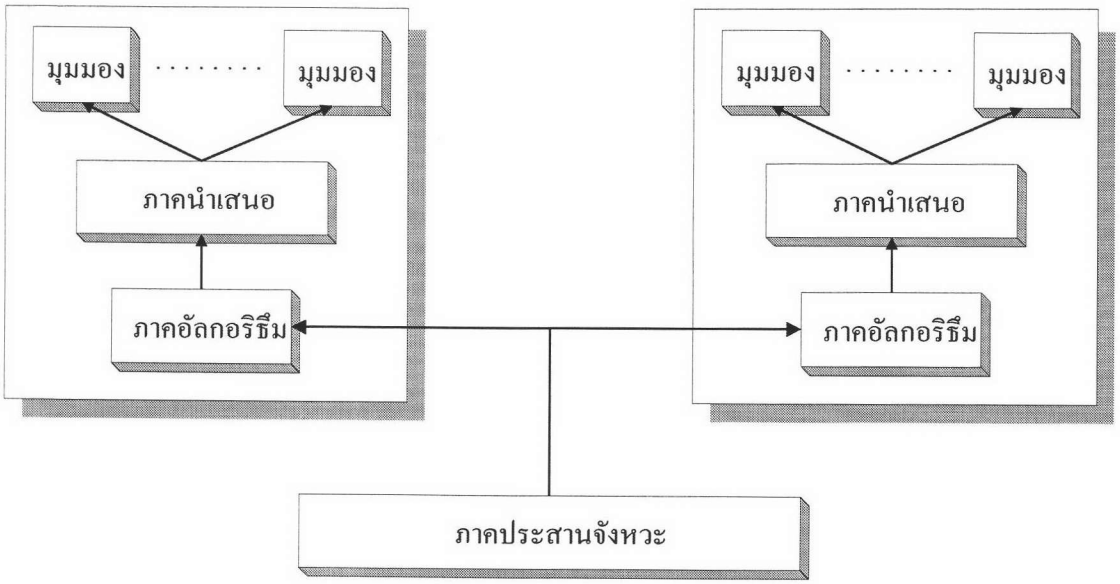
2..3 "DropSync" การบอกยกเลิกการรับบริการการประสานจังหวะ ฟังก์ชันนี้จะมีการเรียกใช้เมื่ออัลกอริทึมได้ทำงานเสร็จสิ้น เพื่อบอกให้ส่วนควบคุมไม่ต้องส่งสัญญาณประสานจังหวะให้อีก เมื่อมีการบอกเลิกการรับบริการจำนวนผู้ถูกควบคุมจะน้อยลงส่งผลให้อัลกอริทึมที่เหลืออยู่ได้รับสัญญาณประสานจังหวะแต่ละรอบที่เร็วขึ้น การทำงานของอัลกอริทึมก็จเร็วขึ้นด้วย ตัวอย่างการใช้ฟังก์ชันของผู้ถูกควบคุมในอัลกอริทึมค้นหาข้อมูลแบบทวิภาคแสดงได้ดังรูปที่ 4.18

```
Sub mnuRunStart_Click ()
   procnum = RegisterSync()
   Ret_Value = NotifySync(FormMDI.hWnd, procnum)
           :
           :  Process Search / Hash Algorithm
   i = DropSync(procnum)
End Sub
```

```
Sub Alg_Binary_Search ()
            :
   Do
      If KeySearch = Data_Ord(Pos_mid) Then
         Ret_Value = NotifySync(FormMDI.hWnd, procnum)
            :            *** Found Data ***
         Exit Do
      ElseIf KeySearch < Data_Ord(Pos_mid) Then
         Ret_Value = NotifySync(FormMDI.hWnd, procnum)
            :            *** Reset Upper Bonud ***
      Else
         Ret_Value = NotifySync(FormMDI.hWnd, procnum)
            :            *** Reset Lower Bound ***
      End If
   Loop Until (KeySearch = Data_Ord(Pos_mid) Or (Pos_top < Pos_bottom))
End Sub
```

รูปที่ 4.18 แสดงการใช้ฟังก์ชันการขอรับบริการ/การบอกเลิกการประสานจังหวะ

จากโปรแกรมคอมพิวเตอร์จะเห็นว่าส่วนอัลกอริทึมมีการเรียกใช้ฟังก์ชั่นประสานจังหวะ ซึ่งส่วนอัลกอริทึมจะสัมพันธ์กันกับภาคนำเสนอ เพราะจะต้องแสดงการเปลี่ยนแปลงการทำงานของอัลกอริทึมออกไปที่มุมมองต่างๆ สามารถแสดงเป็นภาพให้เห็นได้จากรูปที่ 4.19



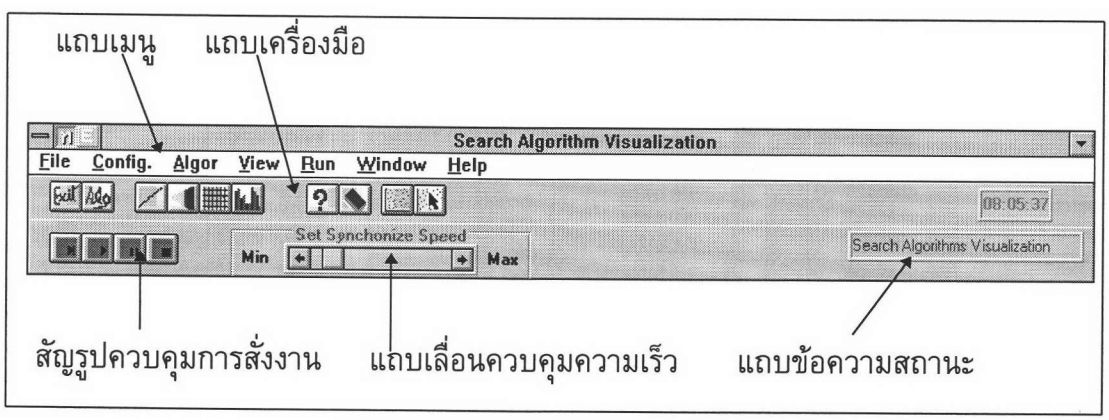
รูปที่ 4.19 ภาคประสานจังหวะกับการนำเสนอ

<u>ระบบจินตทัศน์อัลกอริทึมค้นหาข้อมูลภายใต้สภาพปฏิบัติการไมโครซอฟต์วินโดวส์</u>

ระบบจินตทัศน์อัลกอริทึมค้นหาข้อมูลนี้ผู้วิจัยก็ได้ทำการพัฒนาภายใต้ระบบปฏิบัติการไมโครซอฟต์วินโดวส์ โดยให้มีรูปแบบการติดต่อกับผู้ใช้เป็นไปตามมาตรฐานการโปรแกรมบนวินโดวส์ ในส่วนความสามารถของวินโดวส์ที่ถูกนำมาใช้ในการวิจัยครั้งนี้ประกอบด้วย

1. การประสานกับผู้ใช้แบบกราฟิก (Graphical User Interface : GUI)

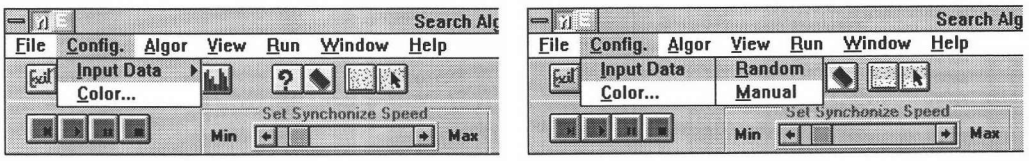
บนวินโดวส์ใช้รูปภาพในการสื่อความหมายและสั่งงาน เช่น สัญรูป (icon) แถบเลื่อน (scroll bar) และเมนู (menu) ในส่วนประสานผู้ใช้ของระบบจินตทัศน์ก็ได้ทำในลักษณะเช่นเดียวกันดังรูปที่ 4.20



รูปที่ 4.20 หน้าต่างส่วนติดต่อผู้ใช้

มีรายละเอียดของส่วนประกอบต่างๆดังนี้

1.1 แถบเมนู (Menu Bar) ประกอบด้วยรายการคำสั่งต่างๆโดยจะเรียงตามกันไปตามแนวนอนด้านบน ในแต่ละหัวข้อของรายการคำสั่งอาจเป็นการสั่งงานเลย หรือเป็นหัวข้อของรายการคำสั่งหลักซึ่งจะมีรายการคำสั่งย่อยอีก ตามรูปที่ 4.21



รูปที่ 4.21 แสดงแถบเมนูเมื่อเลือกคำสั่งหลักและรายการคำสั่งย่อย

รายละเอียดของรายการคำสั่ง

คำสั่งหลัก	คำสั่งย่อย 1	คำสั่งย่อย 2	รายละเอียด
File	Exit		จบการทำงาน
Config	Input Data		
		Random	สร้างข้อมูลขาเข้าโดยระบบ
		Manual	สร้างข้อมูลขาเข้าโดยผู้ใช้
	Color		กำหนดสีในส่วนต่างๆ
Algor	Select Search or Hash Alg.		เลือกอัลกอริทึมเข้าทำงาน
View	Create View		
		Points	กำหนดมุมมองแบบจุด
		Stick Bars	กำหนดมุมมองแบบแท่ง
		Hash Table	กำหนดมุมมองแบบตาราง
		Hash Graph	กำหนดมุมมองแบบกราฟ
		Evaluate	ให้แสดงหน้าต่างสรุปผล
	Reset View		ลบทุกมุมมองแล้วสร้างใหม่เป็นสถานะเริ่มต้น
Run	Step		กำหนดการให้สัญญาณทีละจังหวะ

	Play		เริ่มให้สัญญาณ
	Pause		หยุดการให้สัญญาณชั่วคราว
	Stop		ยกเลิกการให้สัญญาณ
Windows	Cascade		จัดหน้าต่างใหม่ให้ซื่อหน้าต่างเรียงกัน
	Tit Vertical		จัดหน้าต่างใหม่แบ่งตามแนวตั้ง
	Tit Horizontal		จัดหน้าต่างใหม่จัดตามแนวนอน
	Arrange Icon		จัดสัญรูปใหม่ให้เรียงเป็นระเบียบ
Help	About Project		ให้แสดงหน้าต่างรายละเอียดผู้พัฒนา

ตารางที่ 4.2 รายละเอียดคำสั่งในแถบเมนู

1.2 แถบเครื่องมือ เป็นปุ่มควบคุมที่ใช้สัญรูปแทนการสั่งงาน แถบเครื่องมือจะมีการสั่งงานเหมือนกับรายการคำสั่งในแถบเมนูแต่จะแสดงด้วยภาพแทน ช่วยอำนวยความสะดวกในการใช้เพื่อลดขั้นตอนในกรณีแถบเมนูมีการสั่งหลายเมนูย่อย ผู้ใช้สามารถเลือกคำสั่งได้อย่างรวดเร็วโดยเลื่อนเมาส์ไปที่ปุ่มควบคุมที่มีรูปภาพกำกับอยู่และทำการกดปุ่มเมาส์ด้านซ้าย ปุ่มควบคุมต่างๆในระบบจินตทัศน์ที่พัฒนาจะประกอบด้วย

หมายถึง ออกจากโปรแกรมระบบจินตทัศน์อัลกอริทึม

,, เลือกประเภทของอัลกอริทึมมาทำงาน

,, เลือกนำเสนอมุมมองแบบจุด

,, เลือกนำเสนอมุมมองแบบแท่ง

,, เลือกนำเสนอมุมมองแบบตาราง

,, เลือกนำเสนอมุมมองแบบกราฟ

,, เลือกแสดงจำนวนครั้งของการเปรียบเทียบในแต่ละอัลกอริทึม

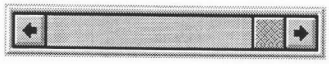
,, ย้อนสถานะของมุมมองแต่ละมุมมองกลับไปที่จุดตั้งต้น

,, สร้างข้อมูลขาเข้าแบบตัวเลขสุ่มโดยระบบ

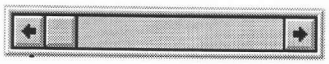
,, สร้างข้อมูลขาเข้าโดยผู้ใช้สร้างเอง

1.3 แถบเลื่อนควบคุมความเร็ว เป็นส่วนที่ใช้ในการควบคุมความเร็วในการทำงานและการนำเสนอการเปลี่ยนแปลงของภาพในมุมมองต่างๆได้ตามที่ผู้ใช้ต้องการ ในแถบเลื่อน

ควบคุมความเร็วนี้จะมีตัวเลื่อนอยู่ภายในซึ่งผู้วิจัยได้ทำการออกแบบตามความรู้สึกของผู้ใช้คือ หากตัวเลื่อนอยู่ทางขวามือสุดแสดงว่า อัลกอริทึมจะทำงานในภาวะที่เร็วที่สุดตามความเร็วของซีพียูแต่หากตัวเลื่อนอยู่ทางซ้ายมือสุดแสดงว่าอัลกอริทึมได้รับการหน่วงให้ทำงานช้าลง วิธีการใช้งานแถบเลื่อนสามารถทำได้โดยการลากเมาส์มาที่ลูกศรทางซ้ายหรือทางขวาเพื่อบอกทิศทางในการเลื่อนจากนั้นก็กดปุ่มเมาส์ด้านซ้ายตัวเลื่อนจะเลื่อนไปตามทิศทางของลูกศรทันที



หมายถึง การทำงานของอัลกอริทึมในสถานะเร็วที่สุด



,, การทำงานของอัลกอริทึมในสถานะช้าที่สุด

รูปที่ 4.22 แสดงสถานะของแถบเลื่อนควบคุมความเร็ว

1.4 ชุดสัญรูปควบคุมการสั่งงาน เป็นชุดที่ใช้ควบคุมการทำงานอัลกอริทึมโดยมีความหมายดังนี้

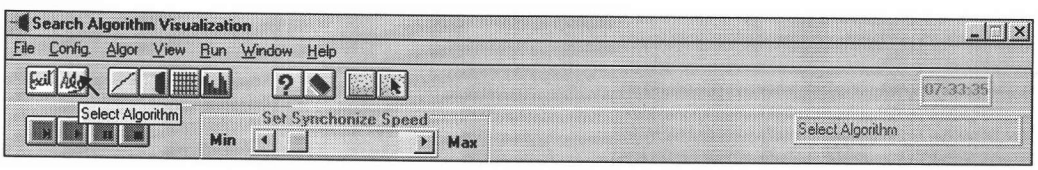
หมายถึง ควบคุมให้อัลกอริทึมทำงานทีละขั้นตอนเมื่อกดปุ่มควบคุมนี้หนึ่งครั้ง

,, การสั่งให้อัลกอริทึมเริ่มทำงาน หรือทำงานต่อเมื่อมีการหยุดชั่วคราว

,, การสั่งให้อัลกอริทึมหยุดทำงานชั่วคราว

,, การยกเลิกการควบคุมการทำงาน

1.5 แถบสถานะ เป็นส่วนที่จะแสดงข้อความสั้นๆสำหรับอธิบายความหมายโดยย่อของแถบเครื่องมือต่างๆ ข้อมูลในแถบสถานะจะเปลี่ยนแปลงเมื่อผู้ใช้ลากเมาส์ไปอยู่เหนือปุ่มควบคุมต่างๆ ดังรูปที่ 4.23



รูปที่ 4.23 แสดงแถบสถานะในแถบเครื่องมือ

2. การทำงานแบบหลายภารกิจ (Multitasking)

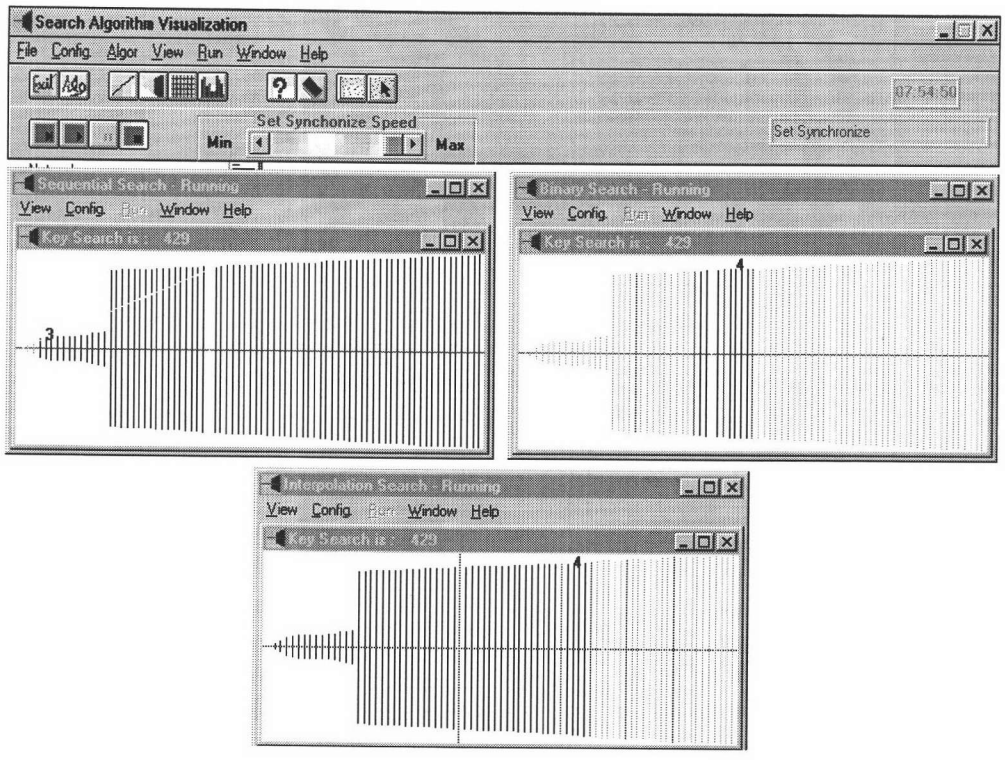
การทำงานบนวินโดวส์มีลักษณะของ non - preemtive multitasking ผู้พัฒนาโปรแกรมจะต้องทำการเขียนโปรแกรมในลักษณะที่ไม่ทำให้เกิดการจองทรัพยากรของเครื่องโดย

เฉพาะ ซีพียู (CPU) แต่เพียงผู้เดียวซึ่งในจุดนี้จะต้องมีความระมัดระวังเป็นอย่างยิ่งเพราะถ้ามีงานหลายงานทำงานอยู่บนวินโดวส์แล้วมีโปรแกรมใดโปรแกรมหนึ่งจองการทำงานของซีพียูไว้จะทำโปรแกรมอื่นไม่ได้โอกาสเข้าใช้ซีพียูเลย จากรูปที่ 4.24 แสดงอัลกอริทึมการวาดจุดลงบนมุมมองซึ่งจะเห็นว่ามีการทำงานแบบวงวนซึ่งเป็นช่วงที่โปรแกรมใช้ทรัพยากรของซีพียูตลอดเวลา ดังนั้นในตัวแปลภาษาวิชวลเบสิกได้ให้ฟังก์ชันพิเศษชื่อ DoEvents ซึ่งมีหน้าที่สำหรับหยุดการทำงานของโปรแกรมชั่วคราวแล้วไปทำการต่อแถวรอรับบริการ และให้วินโดวส์กลับไปสนใจโปรแกรมอื่นๆที่อยู่ในแถวรอรับบริการจนครบและกลับมาถึงรอบที่จะได้รับบริการหลังจากหยุดทำงานด้วยฟังก์ชั่น DoEvents โดยจำทำงานในคำสั่งถัดไปจากฟังก์ชัน DoEvents ซึ่งหากไม่ทำเช่นนี้จะทำให้วินโดวส์ไม่สามารถไปทำงานอื่นๆได้เลยจนกว่าจะทำการสร้างจุดลงในมุมมองจนเสร็จ

```
Sub DrawViewPoints(frmView As Form)
    ::
    idx= Ubound (unit)
    screen.MousePointer = 12
    for i = 0 To idx
            frmView.Pset (i, unit (i) )
            DoEvents
    Next i
End Sub
```

รูปที่ 4.24 แสดงโปรแกรมคอมพิวเตอร์แบบหลายภารกิจบนวินโดวส์

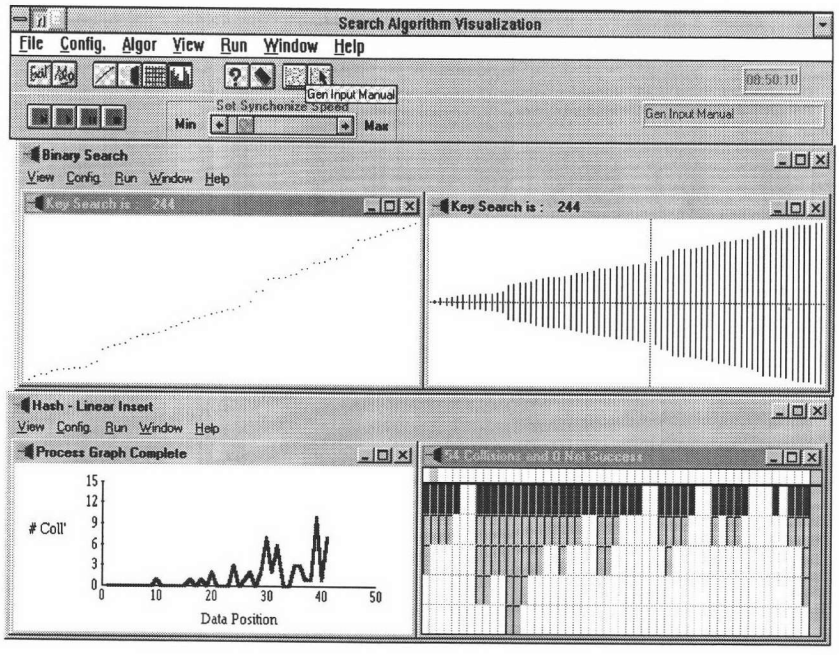
การทำงานแบบหลายภารกิจบนวินโดวส์จะเป็นส่วนที่ใช้ในการแสดงการทำงานของหลายๆอัลกอริทึมได้ในเวลาเดียวกัน เพื่อดูความแตกต่างของการทำงานและประเมินประสิทธิภาพการทำงานดังแสดงในรูปที่ 4.25



รูปที่ 4.25 แสดงการทำงานแบบหลายภารกิจ

3. การประสานแบบหลายเอกสาร (Multiple Document Interface :MDI)

เป็นคุณสมบัติอันหนึ่งที่เอื้ออำนวยต่อการแสดงและจัดการวินโดวส์ย่อย (MDIChild) หลายๆ วินโดวส์ที่อยู่ภายใต้วินโดวส์หลัก (Form MDI) ดังที่ได้กล่าวไว้ในขอบเขตของการวิจัยว่าระบบจินตทัศน์ที่จะพัฒนานี้จะต้องสามารถนำเสนอมุมมองได้หลายมุมมองพร้อมๆกัน คุณสมบัติของวินโดวส์นี้จึงเอื้ออำนวยให้ระบบจินตทัศน์แสดงมุมมองของการนำเสนอได้หลายมุมมองดังแสดงในรูปที่ 4.26 ทำให้ผู้ใช้สามารถเลือกดูความแตกต่างของแต่ละมุมมองในเวลาเดียวกัน เพื่อดูว่ามุมมองแบบใดให้การนำเสนอที่เห็นเด่นชัดและสื่อความหมายที่ผู้ใช้สามารถเข้าใจได้ง่าย



รูปที่ 4.26 แสดงมุมมองแบบหลายเอกสาร

4. การใช้คลังโปรแกรมเชื่อมโยงแบบพลวัต (Dynamic Link Library : DLL)

ดังที่ได้กล่าวมาก่อนหน้านี้แล้วว่าเป็นที่รวบรวมฟังก์ชันการทำงานต่างๆ เมื่อมีการเรียกใช้ฟังก์ชันในคลังโปรแกรมก็จะมีการนำคลังโปรแกรมดังกล่าวเข้าสู่หน่วยความจำและเชื่อมโยงกับโปรแกรมที่เรียกใช้ หากมีหลายๆโปรแกรมเรียกใช้ฟังก์ชันเดียวกันฟังก์ชันที่ถูกเรียกจะใช้เนื้อที่ในหน่วยความจำเพียงชุดเดียว คลังโปรแกรมเชื่อมโยงแบบพบวัตแยกต่างหากจากโปรแกรมประยุกต์แต่จะถูกเชื่อมโยงเมื่อมีการเรียกใช้เท่านั้น ดังนั้นหากมีการเปลี่ยนคลังโปรแกรมก็ไม่จำเป็นต้องนำโปรแกรมที่เรียกใช้มาทำการแปล (compile) ใหม่ด้วย ซึ่งคลังโปรแกรมที่ใช้ในการวิจัยพัฒนาระบบ จะเป็นคลังโปรแกรมที่ทำงานในส่วนประสานจังหวะการทำงาน