



ปัญหาพิเศษ

เรื่อง

การพัฒนาต้นแบบโปรแกรมจัดการเอกสารบนวินโดว

โดย

นายมนตรี วิบูลย์รัตน์  
นางสาวโชติกานต์ แก้วมณี  
นางสาวรุ่งทิวา พุดมฝั่ง

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

ช.ท.  
81132ก  
2537

619522156

เลขหมู่.....  
เลขทะเบียน.....  
วัน,เดือน,ปี.....

**Development Prototyping Program Word Processing on Windows**

**Mr. Montri Wiboonrat**

**Miss Chotikarn Kaewmanee**

**Miss Rungtiwa Putpueng**

**A Special Project Submitted in Partial Fulfilment of the  
Requirement for the Degree of Bachelor of Science  
Department of Applied Mathematics and Computer Science  
Faculty of Science**

**King Mongkut's Institute of Technology Ladkrabang**

**1993**

หัวข้อปัญหาพิเศษ

โดย

ภาควิชา

อาจารย์ที่ปรึกษา

การพัฒนาต้นแบบโปรแกรมจัดเอกสารบนวินโดวส์

นายมนตรี วิบูลย์รัตน์

นางสาวโชติกานต์ แก้วมณี

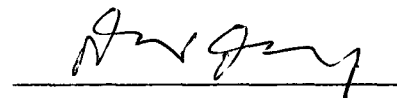
นางสาวรุ่งทิภา พุดมฝั่ง

คณิตศาสตร์ประยุกต์และวิทยาการคอมพิวเตอร์

อาจารย์ปัญญาพล หอระตะ

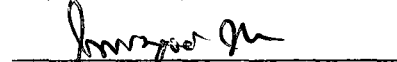
รองศาสตราจารย์ภคคินี ชิตสกุล

ภาควิชาคณิตศาสตร์ประยุกต์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง อนุมัติให้นับโครงการพิเศษฉบับนี้  
เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต



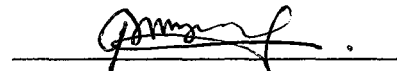
(รองศาสตราจารย์ภคคินี ชิตสกุล)

รักษาการแทนหัวหน้าภาควิชา



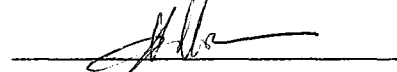
(อาจารย์ไพโรบลย์ พันธรักษ์พงษ์)

ประธานกรรมการสอบโครงการพิเศษ



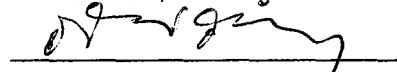
(อาจารย์ภคณา ไตรสุรัตน์)

กรรมการสอบโครงการพิเศษ



(อาจารย์ปัญญาพล หอระตะ)

อาจารย์ที่ปรึกษาปัญหาพิเศษ



(รองศาสตราจารย์ภคคินี ชิตสกุล)

อาจารย์ที่ปรึกษาปัญหาพิเศษ

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์ประยุกต์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

## บทคัดย่อ

ปัญหาพิเศษฉบับนี้ มีวัตถุประสงค์ที่จะทำการพัฒนาต้นแบบโปรแกรมการจัดเอกสารบนวินโดว โดยนำแนวความคิดแบบโครงสร้างเชิงวัตถุมาใช้ในการพัฒนาระบบ ซึ่งจะเน้นให้เห็นถึงการนำไปใช้ ในการพัฒนาต้นแบบโปรแกรมการจัดเอกสารบนวินโดวให้มีการทำงานแบบการเชื่อมต่อแบบหลายเอกสาร และสร้างเครื่องมืออำนวยความสะดวกต่าง ๆ มากมาย เช่น ทูลบาร์ สเตตัสบาร์ การพัฒนาต้นแบบโปรแกรมการจัดเอกสารบนวินโดว แบ่งเป็น 2 ส่วนคือ ขั้นตอนแรกการศึกษาทฤษฎีเชิงวัตถุ และทฤษฎีการเชื่อมต่อแบบหลายเอกสาร ขั้นที่สองออกแบบโครงสร้างระบบและทำการเขียนโปรแกรม

ผลจากปัญหาพิเศษ จะได้ แอปพลิเคชันที่ได้จากการพัฒนาต้นแบบโปรแกรมการจัดเอกสารบนวินโดว ที่สามารถอำนวยความสะดวกในการทำงานเกี่ยวกับการจัดเอกสาร

## **Abstract**

The purpose of this special project is to development prototype program word processing on Windows by using object oriented structure. The topic is in order to develop this program for multiple document interface, and make many tools such as toolbar or statusbar. The document consist of two following step. The first step is studies object oriented thearem and multiple document interface thearem. The second step is designs structure and writes program.

The result of this project is application from development prototyping program word processing on Windows that is facility for work about document.

## กิตติกรรมประกาศ

ปัญหาพิเศษฉบับนี้สำเร็จลงได้ด้วยดีก็เพราะหลายเหตุปัจจัย โดยเฉพาะอย่างยิ่ง

อาจารย์ภักคินี	ชิตสกุล
อาจารย์พัชญา	พิทักษ์ไพรวรรณ
อาจารย์ปัญญาพล	หอระตะ
อาจารย์ศรัณย์	อินทโกสุม

ที่ได้ให้แนวทางในการวิจัย ตลอดจนคำปรึกษาอันก่อให้เกิดแนวความคิดที่สามารถแก้ปัญหาต่าง ๆ ที่เกิดขึ้นในระหว่างการทำกรวิจัย นอกจากนี้ยังช่วยแนะนำแนวทางในการดำเนินงานและตรวจทานแก้ไข ด้วยความเอาใจใส่เป็นอย่างดี

ขอบคุณเจ้าหน้าที่ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ทุกท่านที่สนับสนุนในการใช้ห้องปฏิบัติการคอมพิวเตอร์ และให้ความสะดวกในการเบิกอุปกรณ์ต่าง ๆ ที่ใช้ในการวิจัย

คณะผู้จัดทำขอกราบขอบพระคุณอาจารย์ทุกท่านที่ได้ประสาทวิชาความรู้ทั้งในภาคทฤษฎี ภาคปฏิบัติแก่ผู้จัดทำ จนกระทั่งงานวิจัยสัมฤทธิ์ผลได้ด้วยดีทุกประการ

ขอขอบพระคุณ

คณะผู้จัดทำ

## สารบัญ

หน้าอนุมัติ

บทคัดย่อปัญหาพิเศษภาษาไทย

บทคัดย่อปัญหาพิเศษภาษาอังกฤษ

กิตติกรรมประกาศ

สารบัญรูปภาพ

สารบัญ	หน้า
บทที่ 1 บทนำ	
1.1 ความเป็นมาของปัญหา.....	1
1.2 วัตถุประสงค์ของปัญหาพิเศษ.....	1
1.3 ขอบเขตของปัญหาพิเศษ.....	1
1.4 ขั้นตอนในการดำเนินงาน.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ความหมายและทฤษฎี	
2.1 ทฤษฎีการโปรแกรมเชิงวัตถุ.....	3
2.1.1 ความหมายของการโปรแกรมเชิงวัตถุ.....	3
2.1.2 การทำงานของออบเจ็ค.....	6
2.1.3 หลักการสำคัญ.....	7
2.1.3.1 คลาสและซับคลาส.....	7
2.1.3.2 เอนแคปซูเลชัน.....	10
2.1.3.3 โพลิมอร์ฟิซึม.....	11
2.1.3.4 คุณสมบัติการถ่ายทอด.....	12
2.1.4 ข้อดีของภาษาเชิงวัตถุ.....	13

2.2	ทฤษฎีการเชื่อมต่อแบบหลายเอกสาร.....	14
2.2.1	ความหมายของการเชื่อมต่อแบบหลายเอกสาร.....	14
2.2.2	โครงสร้างของแอปพลิเคชันเอ็มดีไอ.....	16
2.2.3	ส่วนประกอบพื้นฐานของแอปพลิเคชันเอ็มดีไอ.....	18
2.2.3.1	เมสเสจจูป.....	18
2.2.3.2	เฟรมวินโดว์.....	19
2.2.3.3	เอ็มดีไอไคลเอนต์วินโดว์.....	20
2.2.3.4	วินโดว์ลูกเอ็มดีไอ.....	21
2.2.4	พูลบาร์.....	25
2.2.5	สเตตัสบาร์.....	26
บทที่ 3	การออกแบบระบบ	
3.1	บทนำ.....	27
3.2	โครงสร้างของระบบ.....	30
3.3	การใช้งานคลิปปอร์ด.....	74
3.3.1	การก๊อปปี้ข้อมูลลงคลิปปอร์ด.....	75
3.3.2	การปะข้อความจากคลิปปอร์ด.....	76
บทที่ 4	การประเมินผล.....	77
บทที่ 5	สรุปผลและเสนอแนะ	
5.1	สรุปผล.....	81
5.2	แนวทางการวิจัยและข้อเสนอแนะ.....	82
ภาคผนวก		
บรรณานุกรม		

## สารบัญรูปภาพ

รูปที่	หน้า
2.1.1 แสดงให้เห็นโครงสร้างของโปรแกรมตามนิยาม ทั้งสองแบบ.....	4
2.1.2 แสดงโครงสร้างความสัมพันธ์ระหว่างออปเจ็ค ข้อมูล และวิธีการ.....	6
2.1.3 แสดงโครงสร้างของคลาส Paragraph.....	8
2.1.4 แสดงคลาสและซัพคลาสของคลาส Document.....	9
2.2.1 แสดงตัวอย่างหน้าจอแบบเอ็มดีไอ.....	14
2.2.2 แสดงโครงสร้างของแอฟฟลิเคชันแบบเอ็มดีไอ.....	17
2.2.3 แสดงตัวอย่างหน้าจอแบบเอ็มดีไอที่ประกอบด้วย เฟรมวินโดว์, ไคลเอนวินโดว์, วินโดว์ลูก.....	19
2.2.4 แสดงตัวอย่างแอฟฟลิเคชันเอ็มดีไอ.....	23
2.2.5 แสดงแอฟฟลิเคชันเอ็มดีไอที่เพิ่มทูลบาร์ สเตตัสบาร์ และเมนูเฮลป์ .....	24
2.2.6 แสดงไดอะล็อกเอดิเตอร์กับการออกแบบทูลบาร์.....	24
3.1 แสดงโครงสร้างโปรแกรมการพัฒนาด้านแบบโปรแกรม จัดการเอกสารบนวินโดว์.....	31
3.2 แสดงหน้าจอที่ได้จากการรันโปรแกรม word.exe.....	32
3.3 แสดงผลที่ได้จากการเลือกเมนูคำสั่ง File หรือ Alt + F .....	33
3.4 แสดงผลที่ได้จากการเลือกเมนู New ในเมนู File .....	34
3.5 แสดงผลที่ได้จากการเลือกเมนูคำสั่ง Open ในเมนู file เพื่อทำการเลือก File ที่จะทำการเปิดไฟล์ .....	35
3.6 แสดงผลที่ได้จากการเลือกเมนูคำสั่ง Save หรือ Save As เพื่อต้องการให้ผู้ใช้บันทึกเพิ่มข้อมูลลงใน Path ที่ถูกต้อง.....	36
3.7.1 แสดงผลที่ได้จากคำสั่ง Print Preview แสดงผลหนึ่งหน้าจอ .....	38
3.7.2 แสดงผลที่ได้จากการเลือกเมนู Print Preview แบบแสดงผล 2 หน้าจอ .....	39
3.8 แสดงหน้าจอที่ได้จากการเลือกคำสั่ง Print .....	40
3.9 แสดงผลที่ได้จากการเลือกคำสั่ง Print Setup .....	41

3.10	แสดงผลที่ได้จากการเลือกเมนู Edit.....	42
3.11	แสดงการทำแถบสว่างเพื่อเลือกข้อความที่ต้องการคัดลอก.....	44
3.12	แสดงหน้าจอคำสั่งทั้งหมดของเมนู Search .....	46
3.13	แสดงผลที่ได้จากการเลือกคำสั่ง Find .....	47
3.14	แสดงผลที่ได้จากการเลือกคำสั่ง Replace .....	48
3.15	แสดงผลที่ได้จากการเลือกคำสั่งของ Insert Picture .....	49
3.16	แสดงผลที่ได้จากการเลือกคำสั่ง Insert Sound .....	50
3.17	แสดงรายละเอียดของข้อมูลที่เก็บในคลิปบอร์ด ( เป็นผลที่ได้จากการเลือกคำสั่ง ClipBoard Viewer ) .....	51
3.18	แสดงผลที่ได้จากการเลือกคำสั่ง Fonts .....	52
3.19	แสดงหน้าจอของ ImageDraw Editor .....	53
3.20	แสดงตัวอย่างรูปที่ได้จากการวาดโดยใช้ ImageDraw Editor ( โดยการเปิดเมนู File และเลือกคำสั่ง New ).....	54
3.21	แสดงผลที่ได้จากการเลือกคำสั่ง Open ในเมนู File .....	55
3.22	แสดงผลที่ได้จากการเลือกคำสั่ง Save หรือ Save As ในเมนู File .....	56
3.23	แสดงผลที่ได้จากการเลือกคำสั่ง Print ในเมนู File .....	57
3.24	แสดงผลที่ได้จากการเลือกคำสั่ง Print Setup ในเมนู File .....	58
3.25	แสดงคำสั่งทั้งหมดในเมนู Edit .....	59
3.26	แสดงคำสั่งทั้งหมดในเมนู Help .....	60
3.27	แสดงผลที่ได้จากการเลือกคำสั่ง Font บน Tool Bar ของ ImageDraw Editor .....	61
3.28	แสดงผลที่ได้จากการเลือกคำสั่ง Pen Color หรือ Brush Color บน Tool Bar ของ ImageDraw Editor .....	62
3.29	แสดงคำสั่งย่อยของเมนู View .....	63
3.30	แสดงผลที่ได้จากการปิดสแตตัสบาร์ .....	64
3.31	แสดงผลที่ได้จากการปิดทูลบาร์ .....	64
3.32	แสดงการจัดหน้าจอ เมื่อใช้คำสั่ง Cascade .....	65
3.33	แสดงหน้าจอเมื่อเลือกคำสั่ง Tile Vertical .....	66

3.34	แสดงหน้าจอเมื่อเลือกคำสั่ง Tile Horizontal .....	67
3.35	แสดงหน้าจอเมื่อเลือกคำสั่ง Arrange Icon .....	68
3.36	แสดงหน้าจอเมื่อเลือกคำสั่ง Close All .....	69
3.37.1	แสดงหน้าจอเมื่อเลือกคำสั่ง Contents จากเมนู Help .....	70
3.37.2	แสดงรายละเอียดต่อจากรูปที่ 3.37.1 .....	71
3.37.3	แสดงรายละเอียดต่อจากรูปที่ 3.37.2 .....	71
3.38	แสดงหน้าจอเมื่อเลือกคำสั่ง Keyboard ในเมนู Help .....	72
3.39	แสดงหน้าจอเมื่อเลือกคำสั่ง Using help ในเมนู Help .....	73
3.40	แสดงผลที่ได้จากการเลือกคำสั่ง About ในเมนู Help .....	74

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาของปัญหา

จากประสบการณ์การศึกษาโปรแกรมการจัดการเอกสารบนวินโดวส์ ( Word Processing on Windows ) พบว่าโปรแกรกดังกล่าวนี้เป็นที่น่าสนใจ ประกอบกับยังไม่มีต้นแบบโปรแกรมจัดการเอกสารบนวินโดวส์ ( Development Prototyping Program Word Processing on Windows ) ที่จะ เป็นแนวทางสำหรับผู้ที่ต้องการทำงานวิจัยในด้านนี้มาก่อน จึงได้มีความคิดที่จะพัฒนาต้นแบบโปรแกรมจัดการเอกสารบนวินโดวส์ ซึ่งเป็นโปรแกรมสำหรับการสร้างเอกสารจัดรูปแบบเอกสารที่สัมพันธ์กัน โดยสารสนเทศที่นำมาจัดการจะประกอบไปด้วยข้อความและรูปภาพ โดยการพัฒนาต้นแบบโปรแกรมจัดการเอกสารนี้จะใช้โครงสร้างข้อมูลเชิงวัตถุ ( Object oriented Programming ) ซึ่งจะสามารถแก้ไขข้อมูลได้สะดวกมากขึ้น และในปัจจุบันการโปรแกรมเชิงวัตถุกำลังเป็นที่นิยมอย่างมาก ประกอบกับมีเครื่องมือที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุอย่างแพร่หลาย

#### 1.2 วัตถุประสงค์ของปัญหาพิเศษ

1. เพื่อศึกษาทฤษฎี และกรรมวิธีการสร้างและการจัดการเอกสาร
2. เพื่อให้เกิดทักษะ และประสบการณ์ด้านการพัฒนาโปรแกรม

#### 1.3 ขอบเขตของปัญหาพิเศษ

1. สามารถสร้างไฟล์(File)เอกสารและเปิดไฟล์เอกสารหลาย ๆ ไฟล์ขึ้นมาแก้ไข ( Multi Document Interface )
2. มีความสามารถในการค้นหาคำ( Search ) และแก้ไขคำ ( Replace ) ที่ต้องการได้อย่างต่อเนื่อง
3. สามารถพิมพ์เอกสารที่สร้าง ออกเครื่องพิมพ์โดยเลือกชนิดของเครื่อง พิมพ์ที่ต้องการได้
4. โปรแกรมนี้จะพัฒนาภายใต้โปรแกรมไมโครซอฟท์วินโดวส์ เวอร์ชัน 3.1 ( Microsoft Windows Version 3.1 ) ขึ้นไป

5. เครื่องมือที่ใช้ในการพัฒนาโปรแกรมประกอบด้วย โปรแกรมวิซวลเบสิก เวอร์ชัน 2.0 ขึ้นไป, โปรแกรมบอร์แลนด์ C++ สำหรับวินโดวส์ และ โปรแกรมภาษาซี สำหรับวินโดวส์

6. รูปแบบของแฟ้มข้อมูลรูปภาพจะเป็นแบบบิตแมพ ( BMP )

7. มีเครื่องมือที่ใช้ในการสร้างภาพและปรับตกแต่งภาพได้

8. มีเครื่องมือที่ใช้ในการเรียกแฟ้มข้อมูลเสียง ( .SPM และ .WAV ) ได้

#### 1.4 ขั้นตอนในการดำเนินงาน

1. ศึกษาโปรแกรมวิซวลเบสิก เวอร์ชัน 2.0 ขึ้นไป, โปรแกรมบอร์แลนด์ C++ สำหรับ วินโดวส์ และ โปรแกรมภาษาซีสำหรับวินโดวส์

2. ศึกษาการเขียนโปรแกรมเชิงวัตถุ ( Object Oriented Programming )

3. ออกแบบต้นแบบโปรแกรมจัดเอกสารบนวินโดวส์

4. พัฒนาต้นแบบโปรแกรมจัดเอกสารบนวินโดวส์ตามที่ได้ออกแบบไว้

5. ทดลองต้นแบบโปรแกรมจัดเอกสารบนวินโดวส์ตามที่ได้พัฒนา

6. จัดทำเอกสารประกอบโปรแกรม

#### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้เครื่องมือที่ช่วยให้ผู้ใช้สามารถสร้างและจัดการกับเอกสารได้

2. เป็นแนวทางสำหรับผู้ที่ต้องการทำงานวิจัย ในด้านที่เกี่ยวข้องได้

## บทที่ 2

### ความหมายและทฤษฎี

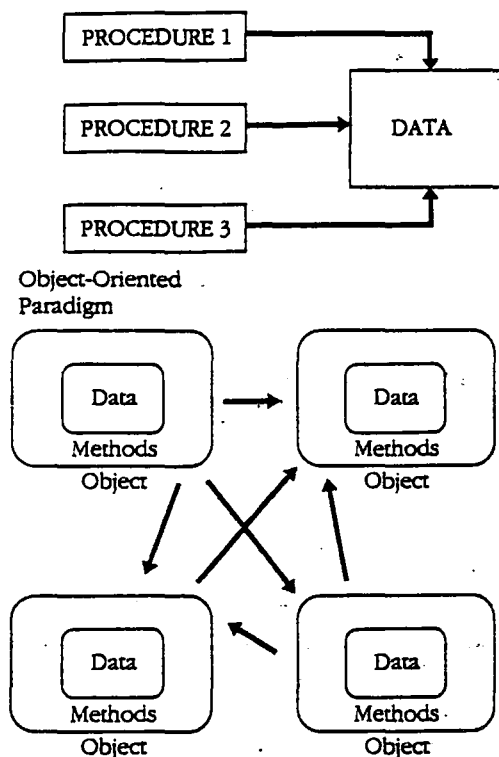
#### 2.1 ทฤษฎีการโปรแกรมเชิงวัตถุ

##### 2.1.1 ความหมายของการโปรแกรมเชิงวัตถุ

การโปรแกรมเชิงวัตถุ ( Object-Oriented Programming หรือ OOP ) เป็นวิธีการเขียนโปรแกรมแบบหนึ่งที่อ้างอิงแนวความคิดแบบเชิงวัตถุ ( Object-Oriented ) โดยจะมองโปรแกรมเป็นระบบที่สนใจ และมีออบเจ็กต์เป็นสมาชิกในโปรแกรม ในออบเจ็กต์หนึ่งจะประกอบไปด้วยส่วนโค้ดหรือรoutinesจัดการต่าง ๆ และส่วนดาต้าได้แก่ ตัวแปรในโปรแกรม

ในลักษณะการเขียนโปรแกรมแบบเดิมนั้น จะพยายามแบ่งงานทั้งหมดเป็นงานเล็ก ๆ แบ่งโปรแกรมออกเป็นโพธิ์เซียร์หรือรoutinesย่อย เพื่อจัดการส่วนต่าง ๆ ในโปรแกรม ดังนั้นคำว่าโปรแกรมในนิยามแบบเดิมจึงหมายถึงเซตของโพธิ์เซียร์ เมื่อมองโปรแกรมเป็นเซตของ โพธิ์เซียร์แล้ว ขั้นตอนการการออกแบบโปรแกรม จึงเป็นขั้นตอนของการจัดโครงสร้างของโพธิ์เซียร์ต่าง ๆ ในโปรแกรม ตามลำดับก่อนหลังจากใหญ่ไปหาเล็ก เรียกว่า การออกแบบจากบนลงล่าง ( Top-down desing ) ซึ่งจะเห็นว่ามิชอบกพรองคือการเขียนโปรแกรมเดิม ไม่อาจแสดงการใช้งานข้อมูลสำคัญต่าง ๆ ที่มีในโปรแกรมได้อย่างเป็นระเบียบดี กล่าวคือ ทุก ๆ โพธิ์เซียร์ในโปรแกรมมีสิทธิใช้งานข้อมูลทุกตัวเมื่อไรก็ได้ โดยที่ไม่อาจทราบอย่างแท้จริงว่า ข้อมูลตัวใดมีความสำคัญกับโพธิ์เซียร์หรือรoutinesใดบ้าง นั่นก็คือ การเขียนโปรแกรมแบบเดิมไม่สามารถ ควบคุมการใช้งานของข้อมูลในโปรแกรมได้

แต่สำหรับในการเขียนโปรแกรมเชิงวัตถุ นั้น โปรแกรมมีนิยามเป็นเซตของออบเจ็กต์ โดยที่ออบเจ็กต์ถูกกำหนดให้เป็นหน่วยใหม่ที่สร้างขึ้นมา เพื่อรวมทั้งส่วนข้อมูลและโค้ดที่จัดการกับข้อมูลนั้นไว้ด้วยกัน เมื่อนิยามของการโปรแกรมเปลี่ยนไปการออกแบบโปรแกรมจึงกลายเป็นการออกแบบโครงสร้าง และความเกี่ยวข้องระหว่างออบเจ็กต์แต่ละตัวในโปรแกรมแทน



รูปที่ 2.1.1 แสดงให้เห็นโครงสร้างของโปรแกรมตามนิยามทั้งสองแบบ

จากรูป เป็นการเปรียบเทียบการเขียนโปรแกรมแบบเดิม ที่มีการแบ่งโปรแกรมออกเป็นโพรซีเจอร์หรือรoutinesย่อย เพื่อจัดการส่วนต่างๆ ในโปรแกรม โดยที่ทุก ๆ โพรซีเจอร์ในโปรแกรมมีสิทธิใช้งานข้อมูลทุกตัวเมื่อไรก็ได้ เมื่อเปรียบเทียบกับการเขียนโปรแกรมเชิงวัตถุ ที่มองแต่ละส่วนของโปรแกรมเป็นออบเจ็ค โดยที่แต่ละออบเจ็คจะมีวิธีการ (method) และข้อมูลของตัวเองออบเจ็คนั้น ๆ เอง นอกจากนี้ยังสามารถเรียกใช้วิธีการและข้อมูลของออบเจ็คตัวอื่น ๆ ได้

เมื่อมีการนิยามโปรแกรมขึ้นมาใหม่จึงมีการสร้างกฎข้อบังคับต่าง ๆ ขึ้นมา สำหรับรองรับโครงสร้างของนิยามใหม่ กฎต่าง ๆ เหล่านี้จะอยู่ในรูปของหลักการสำคัญต่าง ๆ ของการโปรแกรมเชิงวัตถุที่มีหลายประการด้วยกัน อันประกอบไปด้วย คลาส (Class)

เอนแคปซูเลชัน ( Encapsulation )    คุณสมบัติการสืบทอด ( Inheritance )    โพลิมอร์ฟิซึม ( Polymorphism )    และอื่น ๆ ซึ่งจะกล่าวต่อไป

ชนิดข้อมูลนามธรรม ( Abstract data type ) เป็นหลักของการโปรแกรมเชิงวัตถุ ที่มีการรวมชนิดข้อมูล ( type ) และการปฏิบัติการ ( Operation ) เข้าไว้ด้วยกัน

การนิยามคลาส ( class definition ) คือการกำหนดการทำงานของชนิดข้อมูลนามธรรม โดยการนิยามวิธีการปฏิบัติการของชนิดข้อมูลนั้น นอกจากนี้การนิยามคลาสนี้ยังกำหนดโครงสร้างข้อมูลของชนิดข้อมูลด้วย โดยปกติแล้วโครงสร้างข้อมูลนี้จะเข้าถึงได้เฉพาะภายในคลาส เรียกว่าเป็น ชนิดข้อมูลแบบท้องถิ่น ( private ) แต่ถ้าข้อมูลนั้นเข้าถึงได้จากภายนอกคลาส เรียกว่าเป็น ชนิดข้อมูลแบบทั่วไป ( public )

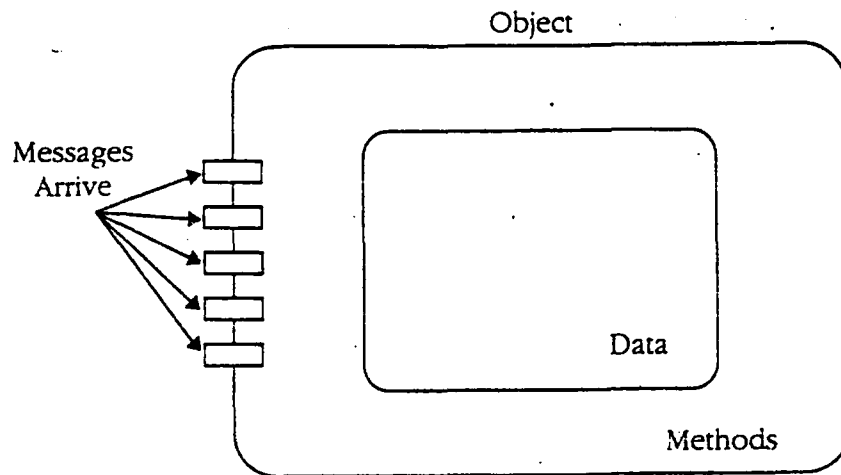
ตัวอย่างเช่น การนิยามคลาส จำนวนเต็ม ชนิดของข้อมูล คือ จำนวนเต็ม (integer) มีการปฏิบัติการ คือ บวก, ลบ, คูณ,หาร ดังนั้น ออบเจ็คในคลาส คือ 1, 2, 3, 4,... หรือ การนิยามคลาส จำนวนนับ มีการปฏิบัติการ คือ จำนวนนับที่เป็นเลขคู่ ดังนั้น ออบเจ็คในคลาส คือ 2, 4, 6, 8, ...

วิธีการ ( method ) คือ การนำชนิดข้อมูลมาใช้งาน ตามวิธีการการตอบสนองต่อเมสเสจ การเรียกใช้วิธีการทำได้โดยการส่งเมสเสจไปยังออบเจ็ค

การใช้งานของชนิดข้อมูล ( type ) สามารถแบ่งได้เป็น 2 แบบ คือ แบบทั่วไป ( public ) และ แบบท้องถิ่น ( private )

- การทำงานแบบทั่วไป สามารถเรียกใช้ได้จากภายนอกคลาส
- การทำงานแบบท้องถิ่น สามารถเรียกใช้ได้เฉพาะภายในคลาส

ออบเจ็ค ในความหมายเชิงวิชาการนิยามว่า คือ ปริมาณหนึ่งในระบบที่ประกอบขึ้นด้วยองค์ประกอบ 2 ส่วนคือ ข้อมูล และ โค้ดโปรแกรม ส่วนข้อมูลใช้เก็บสถานะของตัวเองเรียกว่า ข้อมูล ( data ) และ ส่วนโค้ดโปรแกรม ใช้ในการตอบสนองต่อออบเจ็คตัวอื่นในระบบเดียวกัน เรียกว่า วิธีการ ( method ) [ คอมพิวเตอร์วิวิ, ฉบับที่ 96, 1992 ]



รูปที่ 2.1.2 โครงสร้างความสัมพันธ์ระหว่างออบเจ็ค ข้อมูล เมสเสจ และวิธีการ

ตัวอย่างออบเจ็ค เช่น โปรแกรมไมโครซอฟท์เวิร์ดสำหรับวินโดว ก็เป็นออบเจ็คหนึ่งของวินโดว

### 2.1.2 การทำงานของออบเจ็ค

ออบเจ็คใด ๆ ในระบบ จะสื่อสารกับออบเจ็คอื่นเพื่อให้บรรลุความต้องการของตนเอง การสื่อสารนี้เป็นลักษณะ “ ร้องขอและตอบสนอง ” โดยที่สามารถเรียกใช้วิธีการซึ่งนิยามไว้ในการนิยามคลาส เพื่อให้เกิดการทำงานกับออบเจ็ค การเรียกใช้ทำได้โดยการส่งเมสเสจ ( message ) ไปยังออบเจ็คที่ต้องการ แล้วตัวออบเจ็คจะตีความหมายของเมสเสจออกมาว่าควรจะตอบสนองต่อเมสเสจนั้นด้วยวิธีใด นั่นคือ เมื่อออบเจ็คหนึ่งขอความช่วยเหลือจากอีกออบเจ็คหนึ่ง เรียกว่า กำลังส่งเมสเสจไปยังออบเจ็คอื่น

ตัวอย่างเช่น ในขณะที่กำลังใช้ไมโครซอฟท์วินโดวอยู่นั้น ผู้ใช้คือออบเจ็คหนึ่งในระบบ ในขณะที่วินโดวก็เป็นอีกออบเจ็คหนึ่ง เมื่อผู้ใช้นี้มีการคลิกเมาส์ นั่นคือกำลังส่งเมสเสจให้กับวินโดวเพื่อรอรับผลตอบสนอง เมื่อมีการรับเมสเสจ ออบเจ็ควินโดว

ต้องพิจารณาว่าจะใช้วิธีการ ( method ) ใดในการตอบสนอง และเมื่อวินโดว์ตอบสนอง ก็จะหมายความว่า เป็นการตอบสนองต่อเมสเสจที่ผู้ใช้ร้องขอ ไปด้วยวิธีการที่สร้างขึ้นเพื่อรองรับเมสเสจนั้น และเมื่อมีการตอบสนองเมสเสจครั้งหนึ่ง ก็จะมีผลทำให้สถานะภาพของวินโดว์เปลี่ยนแปลงไป

### 2.1.3 หลักการสำคัญ

ดังที่กล่าวมาแล้วข้างต้นว่า เมื่อมีการนิยามโปรแกรมขึ้นใหม่สำหรับการโปรแกรมเชิงวัตถุ ให้มีนิยามเป็นเซตของออบเจกต์นั้น เมื่อนิยามของการโปรแกรมเปลี่ยนไป โครงสร้างของโปรแกรมเชิงวัตถุต้องเปลี่ยนไปด้วย จึงได้มีการสร้างกฎข้อบังคับต่าง ๆ ขึ้นมาใหม่สำหรับรองรับโครงสร้างของโปรแกรมเชิงวัตถุ โดยกฎต่าง ๆ เหล่านี้จะอยู่ในรูปของหลักการสำคัญต่าง ๆ ของการโปรแกรมเชิงวัตถุที่มีหลายประการด้วยกันดังจะกล่าวถึงคร่าว ๆ ดังต่อไปนี้

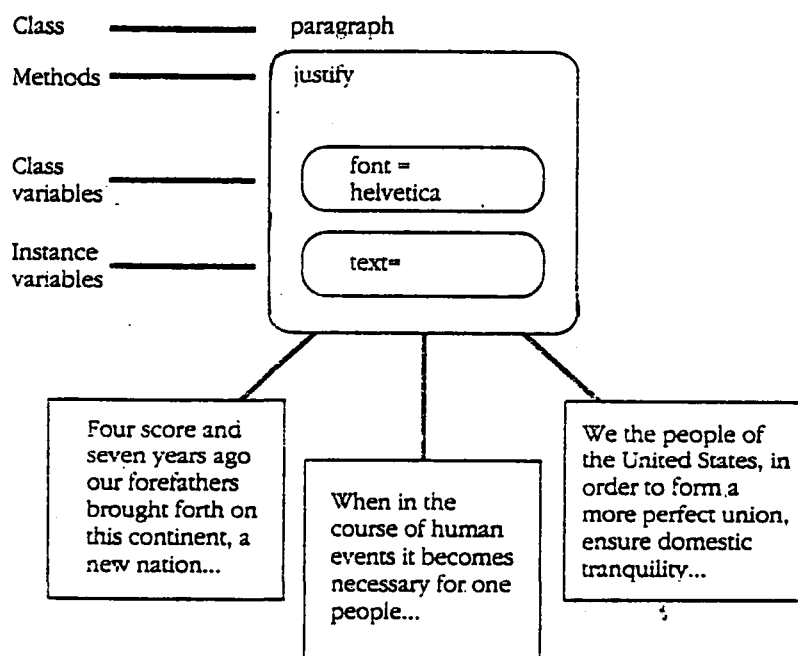
#### 2.1.3.1 คลาสและซึบคลาส

คลาส เป็นเซตของออบเจกต์ที่มีคุณสมบัติ และพฤติกรรมร่วมกัน และอธิบายได้ว่า ออบเจกต์เหล่านี้มีโครงสร้างภายในอย่างไร เช่นคลาสของบุคคล คลาสของพนักงาน ออบเจกต์ทุกออบเจกต์จะต้องอยู่ภายใต้คลาสอย่างน้อยหนึ่งคลาส ในแต่ละคลาสอาจประกอบด้วยออบเจกต์มากกว่าหนึ่งออบเจกต์ ซึ่งออบเจกต์เหล่านี้อาจจะมีลักษณะที่เหมือนกันที่เป็นลักษณะโดยรวมของคลาสนี้ เราเรียกลักษณะโดยรวมของคลาสนี้ว่า ค่าตัวแปรคลาส ( Class Variable ) และสำหรับลักษณะเฉพาะตัวของออบเจกต์ เรียกว่า ตัวแปรตัวอย่างของแต่ละออบเจกต์ในคลาส ( Instance Variable )

จากรูป 2.1.3 สมมติว่าต้องการออกแบบโปรแกรมเวิร์ดโปรเซสเซอร์โดยใช้หลักการของออบเจกต์ ขั้นแรกต้องนิยามคลาส ๆ หนึ่งขึ้นมาที่มีชื่อว่า paragraph คลาส paragraph นี้ เป็นคลาสที่ใช้เก็บคุณสมบัติของข้อมูลหนึ่งย่อหน้าของเอกสาร ประกอบด้วยตัวแปร 2 ตัว คือ font และ text ตัวแปร font เป็นตัวแปรที่ใช้เก็บรูปแบบตัวอักษรที่จะปรากฏให้เห็นหน้าจอ ( เช่น CW ใช้ NORMAL.FON เป็นหลัก ) และตัวแปร text เป็นอาร์เรย์ที่เก็บข้อความเอกสารในย่อหน้านั้น จากการพิจารณาสามารถบอกได้ว่า font นั้นเป็น ตัวแปรคลาส ( class variable ) เพราะเป็นข้อมูลร่วมของคลาส paragraph ในขณะที่ text เป็น ตัวแปรสมาชิกตัวอย่าง ( instance variable ) เพราะเป็นข้อมูล

เฉพาะของตัวอย่าง ( instance ) แต่ละตัว ในย่อหน้าหนึ่งอาจเก็บข้อความ “ การเขียนโปรแกรมเชิงวัตถุ ” แต่อีกย่อหน้าอาจเก็บข้อความว่า “ ฐานข้อมูลเชิงวัตถุ ” ก็ได้

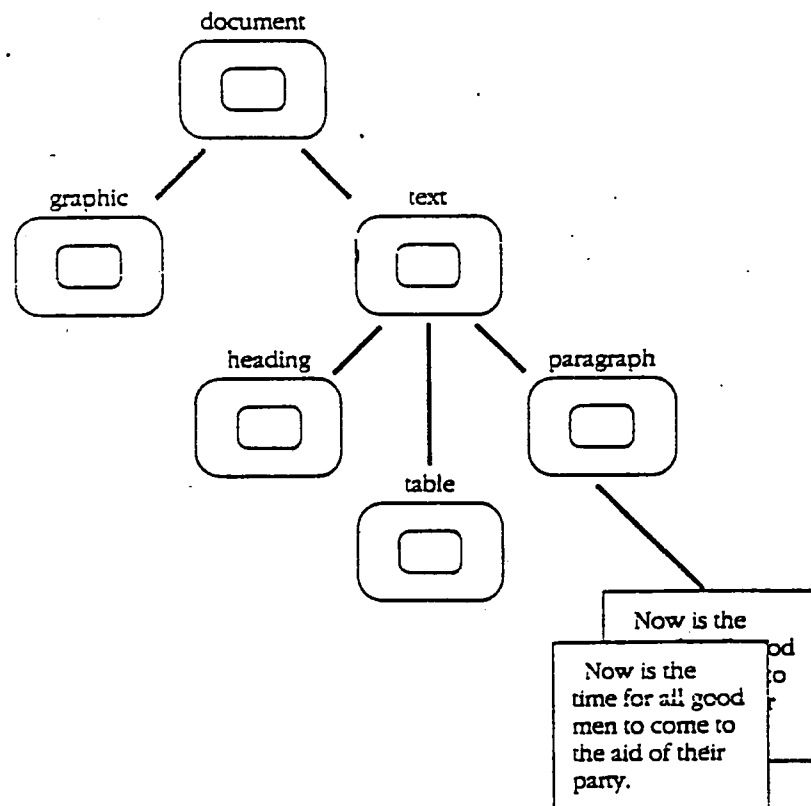
### The Vernacular



รูปที่ 2.1.3 โครงสร้างของคลาส paragraph

สมาชิกตัวอย่าง ( instance ) เป็นออบเจ็กต์ที่สร้างขึ้นจากคลาส โดยที่คลาสจะเป็นตัวบอกถึงโครงสร้างของสมาชิกตัวอย่าง ในขณะที่สถานะปัจจุบันของสมาชิกตัวอย่างถูกนิยามขึ้นโดยคุณสมบัติที่แสดงจากสมาชิกตัวอย่างเอง

เนื่องจากหลักการเชิงวัตถุ เป็นหลักการที่อาศัยการสืบทอดคุณสมบัติเป็นหลัก คลาสแต่ละคลาสในระบบจึงสามารถให้กำเนิดลูกหลานได้ เรียกว่า ชั้นคลาส ( sub-class ) ชั้นคลาสจะสืบทอดคุณสมบัติของคลาสผู้ให้กำเนิด ( parent class ) เช่น จากรูปที่ 2.1.4 เป็นตัวอย่างคลาส document ที่ให้กำเนิดชั้นคลาส text และ graphic โดยที่ชั้นคลาส text ยังสามารถให้กำเนิดชั้นคลาสอื่น ๆ ขึ้นมาได้อีก คือ ชั้นคลาส heading, table และ paragraph ชั้นคลาสนี้บางที่เรียกว่า derived class หรือ คลาสลูก ( child class )



รูปที่ 2.1.4 คลาสและชั้นคลาสของคลาส Document

จากการกำหนดคลาสโดยอาศัยการสืบทอด ทำให้เกิดคลาสขึ้นมามากมายในระบบ ที่เรียกว่า คลาสไลบรารี ( class library ) ขึ้น หรือบางครั้งเรียกว่า คลาสลำดับชั้น ( class hierarchy ) ซึ่งหมายถึงชุดหรือโครงสร้างความสัมพันธ์ของคลาสต่าง ๆ ในระบบแบบเชิงวัตถุ ซึ่งเป็นเสมือนกับไลบรารีที่ใช้เก็บโปรแกรมที่จะนำมาลิงค์ต่อของโปรแกรมที่เขียนขึ้นแบบเก่า แต่แทนที่จะเป็นชุดฟังก์ชัน ก็เป็นโครงข่ายของคลาสในระบบแทน มีข้อดีคือสามารถลดเวลาในการพัฒนาโปรแกรมลงได้ เพราะสามารถทำการเปลี่ยนแปลงแก้ไขในคลาสไลบรารีได้

### 2.1.3.2 เอนแคปซูเลชัน

เอนแคปซูเลชัน คือ การเปลี่ยนแปลงแก้ไข ค่าตัวแปรภายในออบเจกต์จะต้องกระทำผ่านทาง วิธีการ ( Method ) ของออบเจกต์ดังกล่าวเท่านั้น เป็นลักษณะของการป้องกันการแก้ไขค่าตัวแปรสมาชิกภายในออบเจกต์โดยตรง ตัวแปรใด ๆ ภายในออบเจกต์จะมีสถานะค่าปริยาย ( default ) เป็นข้อมูลท้องถิ่น ( Private ) ข้อมูลแบบนี้จะไม่สามารถเข้าถึงได้โดยตรง การจะขอดูหรือเปลี่ยนแปลงค่า ต้องกระทำผ่านโพธิษะเตอร์ที่เป็นวิธีการในออบเจกต์นั้นเท่านั้น ดังตัวอย่างโปรแกรมภาษา C++ ต่อไปนี้

```
class Point {
    int X, Y;
public :
    int GetX () { return X; }
    int GetY () { return Y; }
};
```

จากโปรแกรมตัวอย่างเป็นคลาส Point ของออบเจกต์ที่เก็บข้อมูลของจุด ๆ หนึ่งบนจอภาพ การจะขอค่า X หรือ Y ในออบเจกต์นั้นไม่สามารถทำได้โดยตรง เช่น

```
printf ( "%d", X );
```

แต่ต้องผ่านการเรียกใช้วิธีการ int GetX () เสียก่อน ดังเช่น

```
printf ( "%d", Apoint.Getx() );
```

จากคำสั่งนี้จะเห็นว่า GetX () คืนค่า X มาให้ตัวแปร Apoint.GetX() ก่อนทำการพิมพ์ เป็นต้น

จากวิธีการข้างต้นตามตัวอย่าง ทำให้สามารถเข้าใจโปรแกรมได้ดีขึ้น และทำให้รู้ว่าออบเจ็กต์ตัวไหนในโปรแกรมกำลังต้องการทำอะไร แต่ก็มีข้อเสียคือ ทำให้ความเร็วในการทำงานของโปรแกรมลดลง

การกำหนดเอนแคปซูเลชันซึ่งเป็นคุณสมบัติของออบเจ็กต์ ควรมีลักษณะดังนี้

1. กำหนดขอบเขตที่ชัดเจนให้กับออบเจ็กต์
2. กำหนดส่วนเชื่อมโยง ( interface ) หมายความว่า ออบเจ็กต์นั้นจะติดต่อกับออบเจ็กต์อื่นอย่างไร
3. ส่วนการนำไปใช้งาน ไม่สามารถเข้าถึงได้นอกเหนือขอบเขตของคลาส ที่ผลิตออบเจ็กต์นั้น

### 2.1.3.3 โพลิมอร์ฟิซึม

โพลิมอร์ฟิซึม หมายถึง คุณสมบัติที่ระบบยอมให้การส่งเมสเสจแบบเดียวกันสามารถตอบสนองได้หลายแบบ และไม่จำเป็นต้องได้รับการตอบสนองเหมือนกัน ขึ้นอยู่กับออบเจ็กต์ที่รับเมสเสจเป็นสำคัญ โดยออบเจ็กต์จะตอบสนองได้โดยใช้วิธีการที่มีอยู่ในตัวมันเอง ตัวอย่างเช่น เมื่อออบเจ็กต์ A ส่งเมสเสจ Show ไปให้กับออบเจ็กต์ B ออบเจ็กต์ B จะตอบสนองโดยใช้วิธีการ Show() ที่มีในตัวเอง ถ้าออบเจ็กต์ B เป็นออบเจ็กต์ในคลาส Rectangle ก็จะสามารถรูปสี่เหลี่ยม ถ้าเป็นออบเจ็กต์ในคลาส Circle ก็จะสามารถรูปวงกลม โดยที่ทั้งหมดนี้ใช้เมสเสจเดียวกันคือ Show..

การยินยอมให้มีการตั้งชื่อโพไซเธอร์ หรือวิธีการซ้ำกันได้ในโปรแกรมนั้น เป็นคุณสมบัติขั้นมูลฐานของภาษาแบบเชิงวัตถุโดยทั่ว ๆ ไป แต่ในบางภาษาไม่เรียกส่วนนี้ว่าเป็นโพลิมอร์ฟิซึมแต่จะรวมเข้ากับคุณสมบัติการสืบทอด เพราะว่าในคลาสลูกที่รับเอาวิธีการมาจากคลาสพ่อสามารถทำการแก้ไขวิธีการดังกล่าวได้ทุกกรณี

ในระบบแบบเชิงวัตถุ มีองค์ประกอบที่สำคัญที่สุด คือ คลาสไลบรารี ( class library ) ซึ่งประกอบไปด้วยคลาสต่าง ๆ มากมาย ที่มีโอกาสจะกำหนดชื่อวิธีการของแต่ละคลาสซ้ำกันได้มาก สามารถแก้ไขได้ ด้วยการนำคุณสมบัติโพลี

มอร์ฟิซึมเข้ามาช่วย ตัวอย่างเช่น การ show ของแต่ละคลาสจะมีวิธีการตอบสนองที่ต่างกัน ผู้พัฒนาระบบไม่จำเป็นต้องตรวจดูชื่อของวิธีการในทุก ๆ คลาสเพื่อป้องกันการตั้งชื่อซ้ำก่อนประกาศชื่อวิธีการ ในด้านการใช้งานระบบ ผู้ใช้ก็ไม่จำเป็นต้องจดจำวิธีการส่งเมสเสจหลายวิธี อย่างเช่นวินโดว์ใช้การคลิกเมาส์เพื่อออกคำสั่ง วินโดว์ต่าง ๆ ซึ่งถือเป็นออปเจ็คแต่ละตัวในระบบนั้น ใช้การคลิกเมาส์สองครั้งซ้อนเหมือนกันทั้งหมด ไม่ว่าจะผู้ใช้ต้องการจะเปิดวินโดว์ File manager หรือเปิดวินโดว์เกมส์ก็ใช้วิธีการเดียวกัน การคลิกเมาส์สองครั้งซ้อนเป็นการส่งเมสเสจแบบเดียวกัน แต่การตอบสนองจากระบบขึ้นอยู่กับว่ากำลังต้องการเปิดวินโดว์อะไรอยู่

จุดประสงค์ของโพลีมอร์ฟิซึมมีความต้องการเน้นความคิดที่ว่า

“ ภาษาออปเจ็คโอเรียนเต็ดที่ดีนั้น ควรที่จะสามารถตัดสินใจได้ว่าควรใช้มาตรฐานของออปเจ็คตัวไหนในการตอบสนองเมสเสจที่มีจุดประสงค์คล้าย ๆ กัน ”

#### 2.1.3.4 คุณสมบัติการสืบทอด

การสืบทอดคุณสมบัติของคลาส เป็นวิธีการสร้างคลาสใหม่ โดยอาศัยรูปร่างของคลาสเดิม เป็นคุณสมบัติที่ขาดไม่ได้ของภาษาเชิงวัตถุ เพราะโปรแกรมในนิยามของการโปรแกรมเชิงวัตถุ หมายถึงความสัมพันธ์ของออปเจ็ค ถ้าหากขาดคุณสมบัติการสืบทอดการสร้างชนิดข้อมูลหรือคลาสก็จะทำได้ยากมากขึ้น การที่โปรแกรมเมอร์สามารถกำหนดชื่อวิธีการในคลาสลูก ให้มีชื่อเดียวกันกับวิธีการในคลาสพ่อได้ ก็เพื่อลดความยุ่งยากในการกำหนดชื่อ และเพื่อแสดงกิริยาตอบสนองของออปเจ็คได้ชัดเจนยิ่งขึ้น ดังตัวอย่าง

```
Point = object
```

```
    X, Y : integer;
```

```
    procedure MakePoint ( X,Y : integer );
```

```
    procedure Show;
```

```
end;
```

```
Circle = object (Point)
```

```
    Radius : integer;
```

```
    procedure MakeCircle ( r : integer );
```

procedure Show;

end;

ในตัวอย่างเป็นคลาส Point และ Circle ที่สร้างขึ้นด้วยภาษา PASCAL จะเห็นได้ว่าในคลาส Circle ที่เป็นคลาสลูกของ Point นั้น มีโพรซีเยอร์ที่ชื่อ Show เหมือนกัน แต่ Show ของคลาส Circle เป็นการแสดงภาพวงกลมบนหน้าจอ ในขณะที่ Show ของ Point เป็นการแสดงจุดที่หน้าจอ คลาสลูกคือ Circle จะได้รับการสืบทอดโพรซีเยอร์ Show มาจาก Point และสามารถดัดแปลงแก้ไขได้ตามความต้องการ

ในภาษาแบบเชิงวัตถุเวอร์ชันใหม่นั้น จะมีการนำคุณสมบัติทางการสืบทอดแบบหลายพ่อหลายแม่มาใช้ด้วย เรียกว่า คุณสมบัติการสืบทอดแบบหลายทาง ( Multiple inheritance ) หมายถึง การสืบทอดโดยรับเอาลักษณะของคลาสมากกว่าหนึ่งคลาสมาสร้างคลาสใหม่ ตัวอย่างเช่น โปรแกรมเวิร์ดโปรเซสเซอร์ในปัจจุบัน มักจะรวมความสามารถในการเก็บภาพกราฟฟิกไว้ร่วมกับเอกสารได้ หากสร้างโปรแกรมที่มีคลาสชื่อ WordProcessor ขึ้น ก็จะเป็นคลาสที่สืบทอดมาจากคลาสสองคลาสคือ TextEditor และ ImageProcessor เพราะมีลักษณะของคลาสทั้งสองรวมไว้ในตัว คือสามารถเขียนข้อความได้เหมือนกับเอดิเตอร์ทั่วไป และยังสามารถแสดงและเก็บภาพกราฟฟิกเช่นเดียวกับโปรแกรมจัดการภาพกราฟฟิกได้ด้วย

ข้อดีที่ได้จากคุณสมบัติการสืบทอด

1. การช่วยลดเวลาในการพัฒนาระบบ
2. ลดค่าใช้จ่ายผู้พัฒนา
3. ได้ระบบที่มีโครงสร้างเป็นระเบียบและปรับปรุงเปลี่ยนแปลงได้ง่าย

#### 2.1.4 ข้อดีของภาษาเชิงวัตถุ

เป้าหมายหลักของการพัฒนาซอฟต์แวร์แบบเชิงวัตถุ คือ

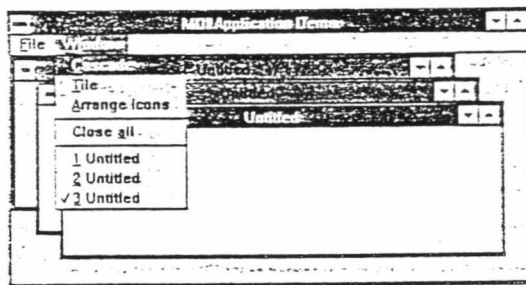
1. ทำให้การพัฒนาซอฟต์แวร์ใช้เวลาสั้นลง และต้นทุนต่ำลง โดยการ  
ใช้คุณสมบัติคลาส ที่สามารถนำกลับมาใช้ได้อีก และสร้างซับคลาสขึ้นมา เพื่อนำมา  
ใช้แก้ปัญหา

2. ทำให้ต้นทุนในการบำรุงรักษาซอฟต์แวร์ต่ำลง เพราะสามารถหาจุดที่ต้องการ  
เปลี่ยนแปลงในซอฟต์แวร์ได้ และการเปลี่ยนแปลงไม่ทำให้เกิดผลกระทบไปยังภายนอก  
คลาสได้

## 2.2 ทฤษฎีการเชื่อมต่อแบบหลายเอกสาร

### 2.2.1 ความหมายของการเชื่อมต่อแบบหลายเอกสาร

การเชื่อมต่อแบบหลายเอกสาร ( Multiple Document Interface ) หรือที่เรียกกันทั่วไปว่า เอ็มดีไอ ( MDI ) เป็นเทคนิคการติดต่อกับผู้ใช้ เพื่อจัดการกับเอกสารได้หลาย ๆ หน้า พร้อม ๆ กันภายใต้สิ่งแวดล้อมเดียวกัน ซึ่งปัจจุบันกำลังได้รับความนิยมอย่างมาก



รูปที่ 2.2.1 ตัวอย่างหน้าจอแบบเอ็มดีไอ

ในการใช้แอปพลิเคชันภายใต้วินโดวส์ กรณีที่แอปพลิเคชันไม่ได้เป็นแบบเอ็มดีไอ การจัดการเอกสารหลาย ๆ หน้า ก็สามารถทำได้ด้วยวิธีการมัลติอินสแตนซ์ ซึ่งวินโดวส์อนุญาตให้รันแอปพลิเคชันตัวเดียวกันได้หลายครั้ง ตัวอย่างเช่นเราใช้โปรแกรม Thai Write ที่ติดมากับวินโดวส์โปรแกรมนี้ไม่สนับสนุนเอ็มดีไอ ดังนั้นการจัดการเอกสารตัวที่สองตัวที่สาม... จำเป็นต้องรันโปรแกรม Write ตามจำนวนที่ต้องการ เมื่อเราต้องการเลือกจัดการกับเอกสารตัวใดตัวหนึ่งเพียงคลิกเมาส์ลงบนวินโดวส์นั้น ๆ หรืออาจกด Alt+Tab เพื่อเปลี่ยนจากตัวหนึ่งไปตัวหนึ่ง

การจัดการเอกสารหลาย ๆ หน้าแบบมัลติอินสแตนซ์ดังกล่าว เมื่อนำมาเทียบกับแบบเอ็มดีไอ นั้นจะพบว่า ความสามารถยังไม่ดีเท่าแบบเอ็มดีไอ ด้วยเหตุผลที่ว่า แอปพลิเคชันเอ็มดีไอ เอกสารแต่ละหน้าอยู่ภายใต้สิ่งแวดล้อมเดียวกัน แต่สามารถใช้งานได้โดยไม่เกิดความสับสนเมื่อเปลี่ยนไปเปลี่ยนมา การจัดการตำแหน่ง ขนาดของ

เอกสาร หรือวินโดว์แต่ละตัวง่ายกว่ารวมทั้งการยุบตัวไปเป็นไอคอนด้วย การเรียกใช้งานเอกสารมีความถูกต้องแม่นยำ และสะดวกกว่า แอปพลิเคชันที่ได้ประโยชน์จากการติดต่อแบบเอ็มดีไอเป็นอย่างมาก ได้แก่ โปรแกรมเวิร์ดโปรเซสซิง, สเปรดชีต เดสก์ท็อปพับลิชซิง เพราะโปรแกรมเหล่านี้อนุญาตให้ผู้ใช้จัดการเอกสารที่ละหลาย ๆ หน้า หรือหลาย ๆ วินโดว์ภายใต้สิ่งแวดล้อมเดียวกัน

ในอดีตการเขียนแอปพลิเคชันเอ็มดีไอ เป็นสิ่งที่ยากลำบากมากเพราะโปรแกรมเมอร์ต้องสร้างสิ่งแวดล้อมพื้นฐาน ของการติดต่อแบบเอ็มดีไอขึ้นมาเองทั้งหมด จึงมีการเขียนต่างกันไปหลาย ๆ รูปแบบ ดังนั้น เมื่อมีวินโดว์เวอร์ชัน 3.0 ไมโครซอฟท์ จึงได้นิยามการติดต่อเอ็มดีไอ และบรรจุโค้ดลงในวินโดว์ สิ่งแวดล้อมพื้นฐานการติดต่อเอ็มดีไอ ประกอบด้วย

- การย่อวินโดว์ลูกให้อยู่ในรูปไอคอน เมื่อถูกย่อขนาดให้เล็กที่สุด ( minimize ) การขยายวินโดว์ลูกให้เต็มไคลเอนต์วินโดว์ เมื่อถูกเพิ่มขนาดให้ใหญ่ที่สุด ( maximize ) และต้องจัดแสดงปุ่มนำกลับคืนที่เดิม ( restore ) และเมนูระบบ ( system menu ) ไว้ที่เฟรมวินโดว์

- การสร้างแถบเลื่อนด้านข้าง ( scroll bar ) ตามแนวนอนและแนวตั้ง ที่ไคลเอนต์วินโดว์ เมื่อส่วนใดส่วนหนึ่งของวินโดว์ลูกออกนอกเนื้อที่ไคลเอนต์ เพื่อการทำหน้าจอสี่เหลี่ยม ( visual screen )

- การคำนวณและจัดเรียงวินโดว์ลูก ให้อยู่ในตำแหน่งและขนาดที่เหมาะสม เมื่อมีการสั่งจัดเรียงแบบต่อเรียง ( cascaded ), การจัดเรียงแสดงให้เห็นทุกหน้าจอ ( tiled ) และจัดเรียงไอคอน ( arrange icon )

- การรับขีตเต็มคีย์บอร์ดเช่น Ctrl+F4 เพื่อปิดวินโดว์ลูก, Ctrl+Tab และ Ctrl+F6 สลับวินโดว์ลูก, Alt+Spacebar เรียกเมนูระบบของเฟรม, Alt+- เรียกเมนูระบบของวินโดว์ลูก

- การจัดเมนูระบบบนวินโดว์ลูก ให้อนุกรมกับเมนูหลักของแอปพลิเคชันด้วย เมื่อมีการเลื่อนเมนูไปทางซ้ายมือหรือขวามือ รวมทั้งการเติมเมนูย่อย เข้ากับเมนูหลัก เพื่อแสดงจำนวนและใช้เลือกวินโดว์ลูกด้วย

คุณสมบัติที่กล่าวมาทั้งหมดเป็นสิ่งแวดล้อมมาตรฐานที่แอปพลิเคชันเอ็มดีไอ มีให้ผู้ใช้ ซึ่งคุณสมบัติดังกล่าวถูกบรรจุอยู่ในวินโดว์คลาส MDICLIENT ในปัจจุบันพบว่าได้มีการพัฒนาการติดต่อผู้ใช้ไปอีกระดับหนึ่ง แอปพลิเคชันใหม่ ๆ จะมีแถบต่าง ๆ เช่น ขูล

บาร์, สเตตัสบาร์, ออนไลน์เมนูเฮลป์ ( On-line menu help ) คุณสมบัติเหล่านี้จะเก็บไว้พิจารณาในตอนท้าย ในตอนนี้จะกล่าวถึงเรื่องเอ็มดีไอก่อน

แอปพลิเคชันเอ็มดีไอประกอบด้วยวินโดว์หลัก ( main window ) ซึ่งผู้ใช้สามารถเปิดได้หลายหน้าต่างเอกสารตามที่ใช้ต้องการ แต่หน้าต่างเอกสารจะปรากฏบนวินโดว์ของตัวเอง ซึ่งก็คือวินโดว์ลูกที่มีเมนูระบบและส่วนประกอบอื่น ๆ

การเขียนแอปพลิเคชันเอ็มดีไอต้องทำตามเงื่อนไขที่เพิ่มขึ้นมา และศึกษาเรื่องใหม่ ๆ เพิ่มเติมบ้างเล็กน้อย ส่วนสำคัญของการเขียนโปรแกรมในแต่ละส่วนแต่ละขั้นตอนของแอปพลิเคชันเอ็มดีไอ เปรียบเหมือนการเขียนโปรแกรมบนวินโดว์ แต่การติดต่อกับผู้ใช้ชั้นพื้นฐาน, ตัววินโดว์ และเมนูจำเป็นต้องมีเครื่องมือพิเศษเฉพาะ เครื่องมือดังกล่าวมีส่วนประกอบพื้นฐานด้วยกัน 3 อย่างที่ต้องมีสำหรับทุก ๆ แอปพลิเคชันเอ็มดีไอ

1. เฟรมวินโดว์ ( frame window ) ทำหน้าที่บริการในส่วนวินโดว์หลักของแอปพลิเคชัน แสดงเมนูบาร์ของโปรแกรม และเป็นตัวสร้างออปเจ็ควินโดว์ลูก เช่น ทูลบาร์ และสเตตัสไลน์ ดังนั้น จะมีเพียงเฟรมวินโดว์เดียวในแอปพลิเคชันเอ็มดีไอเท่านั้น

2. ไคลเอนต์วินโดว์ ( client window ) ถูกสร้างขึ้นมาจากเฟรมวินโดว์ เป็นตัวจัดการโอเปอเรชันทั่ว ๆ ไปทั้งหมดที่ได้มาจากแอปพลิเคชัน และใช้ในการสร้างหน้าต่างเอกสารขึ้นใหม่หรือการเปิดไฟล์เอกสารที่มีอยู่แล้ว

3. หน้าต่างเอกสาร ( document window ) หรือ บางที่เรียก “ เอ็มดีไอไคลเอนต์วินโดว์ ” อาจมีหน้าต่างแสดงผลให้เห็นได้เพียงหนึ่งหน้าต่างหรือมากกว่า ใช้แสดงฐานข้อมูลประเภทไฟล์ หรือลักษณะข้อมูลประเภทอื่น ๆ ไฟล์ที่ถูกเปิดในแต่ละครั้งเป็นที่รู้โดยทั่วกันว่าถือเป็นเอกสารที่ถูกกำหนดขึ้นเพื่อเรียกใช้งานเฉพาะบนหน้าต่างนี้

ส่วนประกอบที่กล่าวมาแล้วข้างต้นจะกล่าวถึงอย่างละเอียดอีกทีในหัวข้อ

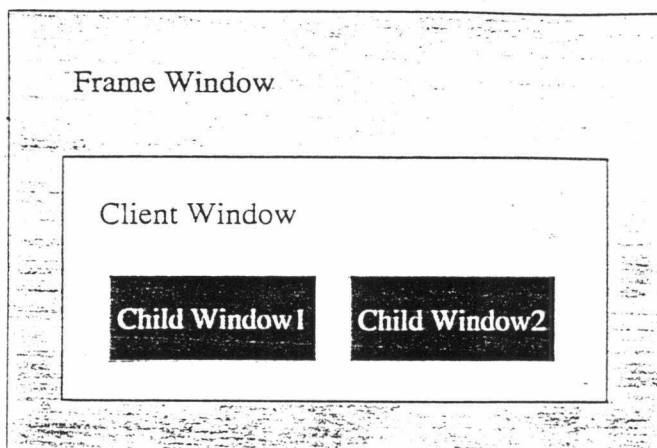
“ ส่วนประกอบพื้นฐานของแอปพลิเคชันเอ็มดีไอ ”

## 2.2.2 โครงสร้างของแอปพลิเคชันเอ็มดีไอ

โครงสร้างของแอปพลิเคชันธรรมดาอย่างง่ายที่สุดประกอบด้วยฟังก์ชัน WinMain ที่ภายในมีเมสเสจลูปซึ่งทำหน้าที่ดึงเมสเสจจากวินโดว์ เพื่อส่งต่อให้กับวินโดว์โพสเซสเซอร์หรือฟังก์ชันที่เขียนขึ้นมาก็เพื่อจัดการเมสเสจที่ต้องการเท่านั้น เช่น การวาดภาพสวอย ๆ บนพื้นที่ไคลเอนต์เมื่อมีเมสเสจ WM\_PAINT การตรวจจับตำแหน่งเมาส์เพื่อใช้ในการวาดภาพ เป็นต้น

การเขียนแอปพลิเคชันแบบเอ็มดีไอ จะต่างไปจากแอปพลิเคชันธรรมดาทั่วไป เพราะว่ามีวินโดว์เพิ่มขึ้นมาอีก 2 วินโดว์ ดังนั้นระเบียบวิธีการเขียนโปรแกรมจะถูกกำหนดขึ้นใหม่เพื่อรองรับสิ่งแวดล้อมและการตอบสนองของผู้ใช้แบบเอ็มดีไอ การเปลี่ยนแปลงนี้สามารถทำได้ตั้งแต่ส่วนของเมสเสจจูป ให้ทำการตรวจสอบคีย์เอ็มดีไอเพิ่มเติมขึ้นมา การสร้างไคลเอนต์วินโดว์ต้องกระทำภายใต้ชื่อคลาส MDIClient ซึ่งมีโพธิ์เซอร์ที่ควบคุมของไคลเอนต์อยู่ภายในวินโดว์ ซึ่งจะรับผิดชอบในการควบคุมวินโดว์ลูก เช่น การจัดเรียง ไอคอน ทำยสุดคือวินโดว์ลูกที่จะต้องสร้างขึ้นมาเพื่อจัดการงานที่ต้องการ เช่น ใช้แสดงกราฟฟิก เป็นต้น

ตามรูปที่ 2.2.2 แสดงถึงกายภาพของแอปพลิเคชันแบบเอ็มดีไอ ที่ประกอบด้วยเฟรมวินโดว์ ไคลเอนต์วินโดว์ และวินโดว์ลูกที่ซ้อนกันอยู่



รูปที่ 2.2.2 โครงสร้างของแอปพลิเคชันแบบเอ็มดีไอ

### 2.2.3 ส่วนประกอบพื้นฐานของแอฟพลิเคชันเอ็มดีไอ

เมื่อพิจารณาถึงลักษณะทางกายภาพของแอฟพลิเคชันเอ็มดีไอ จะพบว่าประกอบไปด้วยเฟรมวินโดว์ ไคลเอนตวินโดว์ และวินโดว์ลูกที่ซ้อนทับกันอยู่ ส่วนประกอบทั้งสามนี้จะต้องมีการทำงานที่สัมพันธ์กันได้ โดยที่จะมีการใช้พื้นที่เหมือนกันเพื่อการส่งเมสเสจ นั่นคือต้องมีเมสเสจลูปเพื่อใช้ในการตรวจสอบคีย์เอ็มดีไอ

#### 2.2.3.1 เมสเสจลูป ( Message Loop )

เมสเสจลูป คือ ลำดับของคำสั่งในโปรแกรมที่ได้รับการปฏิบัติการซ้ำหลาย ๆ ครั้ง เนื่องจากการส่งเมสเสจของออปเจ็คไปยังลูป เพื่อให้ทำการตรวจสอบว่าเป็นไปตามเงื่อนไขใดที่กำหนดไว้ในโปรแกรม โดยทั่วไปวินโดว์ลูกจะมีคีย์ดาวน์ ( accelerator key ) ในเมนูระบบ อย่างเช่น `Alt+F4` เพื่อปิดวินโดว์ลูกเมื่อพร้อมที่จะแก้ไขเอกสารส่วนถัดไป กด `Ctrl+F6` เพื่อสลับวินโดว์ลูก ดังนั้น บริเวณเมสเสจลูปจะทำการตรวจสอบคีย์เอ็มดีไอเหล่านี้เมื่อมีการเรียกใช้

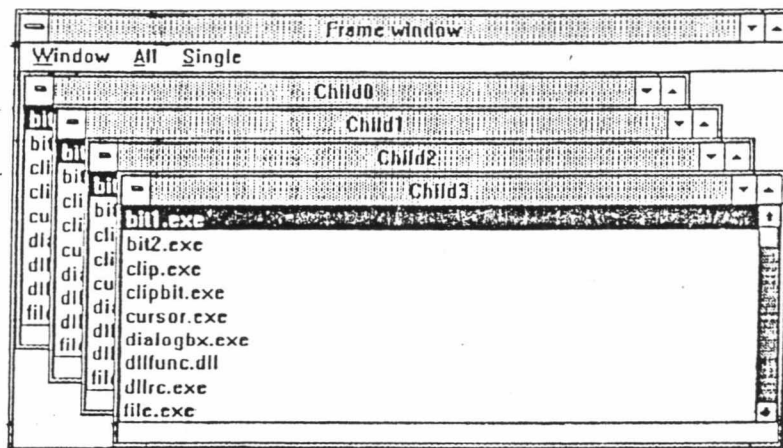
```
while( GetMessage ( &msg, null, 0, 0 ))
{
    if ( !TranslateMDISysaccel ( hWndMDIClient, &msg ) &&
        !TranslateAccelerator( hWndFrame, hAccel, &msg ) )
    {
        TranslateMessage( &msg );
        DispatchMessage( &msg );
    }
}
```

การสร้างเมสเสจลูปสามารถทำได้ดังตัวอย่างโปรแกรมที่แสดงให้เห็นด้านบน โดยมี การเรียกใช้ฟังก์ชัน `TranslateMDISysAccel` เพื่อแปลงระบบการนำข้อมูลเข้าคอมพิวเตอร์ ทางแป้นพิมพ์ที่เป็นของเอ็มดีไอให้เป็น `WM_SYSCOMMAND` เพื่อส่งให้ไคลเอนตวินโดว์ เป็นตัวจัดการ นอกจากนี้ยังมีการ เรียกใช้ฟังก์ชัน `TranstateAccelerator` เพื่อจะแปลงคีย์ดาวน์ ด้วยเช่นกัน ทั้งสองฟังก์ชันส่งค่ากลับเป็น `TRUE` ถ้าสามารถแปลงการกดคีย์ไปเป็นคำสั่งได้

ตามตัวอย่างโปรแกรม ถ้าฟังก์ชันใดฟังก์ชันหนึ่งส่งค่ากลับเป็น TRUE จะถูกกลับค่าเป็นเท็จเมื่อรวมกับโอเปอเรเตอร์ && ที่มีค่าฟังก์ชันข้างใดข้างหนึ่งเป็นเท็จ ดังนั้นผลลัพธ์รวมก็จะได้ค่าเป็นเท็จ ( FALSE ) ทำให้ประโยคเงื่อนไข if ไม่มีการทำงานในส่วนนี้นั้นคือฟังก์ชัน TranslateMessage และDispatchMessage จะไม่ได้รับการทำงานในการวนลูปรอบนั้น

### 2.2.3.2 เฟรมวินโดว์ (FRAME WINDOW)

เฟรมวินโดว์ คือวินโดว์หลักของแอปพลิเคชันเอ็มดีไอ ซึ่งประกอบด้วยเมนูเป็นส่วนใหญ่



รูปที่ 2.2.3 ตัวอย่างหน้าจอแบบเอ็มดีไอที่ประกอบด้วย เฟรมวินโดว์ ไคลเอนตวินโดว์  
วินโดว์ลูก

สำหรับการสร้างเฟรมวินโดว์นั้นจะต้องสร้างให้เสร็จก่อนที่จะเข้าสู่เมสเสจลูป ดังเช่นในฟังก์ชัน Winmain เฟรมวินโดว์ลูก CreateWindow ก่อนที่จะเข้าสู่เมสเสจลูปจริง ๆ ส่วนวินโดว์อื่น ๆ จะถูกสร้างภายหลัง เช่น ไคลเอนตวินโดว์ วินโดว์ลูก เป็นต้น

การที่จะสร้างเฟรมวินโดว์ขึ้นมาได้นั้น ต้องมีการลงทะเบียนคลาสซึ้นใหม่จึงจะสร้างเฟรมวินโดว์ขึ้นมาได้

```

wndclass.style           = 0 ;
wndclass.lpfnWndProc     = FrameWndProc ;
wndclass.cbClsExtra     = 0 ;
wndclass.cbWndExtra     = 0 ;
wndclass.hInstance      = hInstance ;
wndclass.hIcon           = LoadIcon(hInst, cMDIDEMOe);
wndclass.hCursor        = LoadCursor(NULL, IDC_ARROW);
wndclass.hbrBackground  = COLOR_APPWORKSPACE + 1 ;
wndclass.lpszMenuName   = NULL ;
wndclass.lpszClassName  = szFrameClass ;

RegisterClass (&wndclass) ;

```

จากโค้ดด้านบนเป็นตัวอย่งการลงทะเบียนคลาส พิจารณาตัวแปร `wndclass.style` จะถูกกำหนดค่าให้เป็น 0 เพื่อไม่ให้เกิดการกระพริบของหน้าจอ โดยปกติค่าที่ตำแหน่งนี้จะ เป็นค่า `CS_HREDRAW | CS_VREDRAW` ซึ่งเป็นการกำหนดให้วินโดว์ทำการวาดไคลเอนต์ ใหม่ทุกคั้งเมื่อมีการเปลี่ยนแปลงขนาดของเฟรมวินโดว์ และยังมีผลกระทบต่อวินโดว์ลูก ด้วย โดยที่วินโดว์ลูกจะวาดไคลเอนต์ตัวเองใหม่ทุกครั้งด้วยเช่นกัน

ตัวแปร `wndclass.hbrBackground` ถูกกำหนด ให้มีค่าเป็น `COLOR_APPWORKSPACE + 1` เพื่อให้สีพื้นของเฟรมวินโดว์เป็นสีเดียวกับวินโดว์อื่น ๆ ใน ระบบ สำหรับสีพื้นนี้สามารถกำหนดได้จากโปรแกรม Control Panel

### 2.2.3.3 เอ็มดีไอไคลเอนต์วินโดว์ ( MDI Client Window )

สำหรับการสร้างไคลเอนต์วินโดว์ เราสามารถสร้างได้ทันทีโดยไม่ต้องลง ทะเบียนคลาสเพราะว่าเป็นคลาสที่มีอยู่ในวินโดว์อยู่แล้ว หน้าที่หลักของไคลเอนต์วินโดว์ จะคอยบริการ ควบคุมวินโดว์ลูก ๆ ของตัวเอง เช่น การกำเนิดวินโดว์ลูก การจัด ตำแหน่งและขนาดของการจัดเรียงทั้งแบบต่อเรียง ( cascade ) และการจัดเรียงโดยไม่ซ้อน กัน ( tile ) การจัดเรียงไอคอน การสวิตซ์ไปมาระหว่างวินโดว์ลูก ๆ และการต่อ เต็มเมนูแสดงรายการวินโดว์ลูก ๆ การทำหน้าจอสื่อเหมือนจะปรากฏแถบเลื่อนด้านข้าง

เมื่อวินโดว์ลูกเลยขอบเขตของไคลเอนต์ออกไป งานเหล่านี้เป็นบริการพื้นฐานที่ไคลเอนต์วินโดว์มีให้ ถ้าหากเราต้องการจะเพิ่มความสามารถ ของไคลเอนต์ก็สามารถทำได้โดยใช้เทคนิค ชั้นคลาส และ ซูเปอร์คลาส ( เป็นการเชื่อมต่อโค้ดของเราเข้ากับคลาสมาตรฐานที่ติดมากับวินโดว์ ) เช่น ทำรูปถังขยะไว้ปิดงาน เช่นเดียวกับเครื่องแมคอินทอชก็สามารถทำได้

```
clientcreate.hWindowMenu = hMenuInitWindow;
clientcreate.idFirstChild = IDM_FIRSTCHILD;

hwndClient = CreateWindow("MDICLIENT",NULL,
    WS_CHILD | WS_CLIPCHILDREN | WS_VISIBLE,
    0, 0, 0, 0,hwnd, 1, hInst,
    (LPSTR)&clientcreate);
```

โค้ดด้านบนเป็นการสร้างไคลเอนต์วินโดว์ด้วยฟังก์ชัน CreateWindow ที่มีพบเห็นกันบ่อยๆโดยสามารถกำหนดสไตล์ต่าง ๆ ให้กับไคลเอนต์วินโดว์ได้ตามปกติรวมทั้ง WS\_HSCROLL และ WS\_VSCROLL จะทำให้เราได้บริการหน้าจอเสมือนสำหรับไคลเอนต์วินโดว์

#### 2.2.3.4 วินโดว์ลูกเอ็มดีไอ ( MDI child window )

สำหรับวินโดว์ลูกนั้นจะต้องมีการลงทะเบียนคลาสตามปกติ และการสร้างวินโดว์ลูกนั้นไม่สามารถใช้ฟังก์ชัน CreateWindow เพราะว่าวินโดว์ลูกจำเป็นต้องอยู่ภายใต้การควบคุมของไคลเอนต์วินโดว์ ดังนั้นการสร้างวินโดว์ลูกของไคลเอนต์จะต้องส่งให้ไคลเอนต์เป็นผู้สร้างทางเมสเสจ WM\_MDICREATE โดยการส่งเมสเสจนี้จะส่งไปพร้อมตัวแปรโครงสร้างMDICREATESTRUCT ไปยังไคลเอนต์วินโดว์ดังโค้ดด้านล่างนี้

```
mdicreate.szClass = szEllipseClass ;
mdicreate.szTitle = "Ellipses" ;
mdicreate.hOwner = hinst ;
```

```

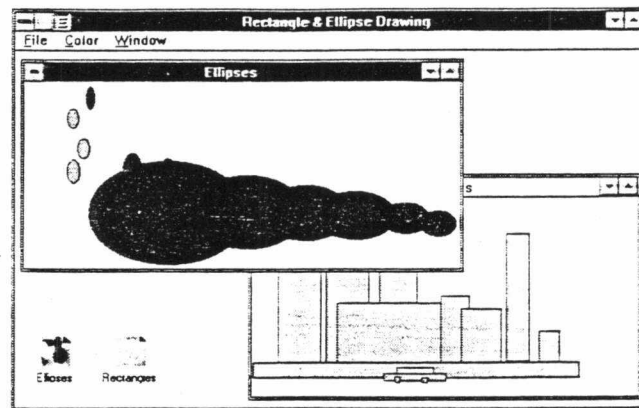
mdicreate.x           = CW_USEDEFAULT ;
mdicreate.y           = CW_USEDEFAULT ;
mdicreate.cx          = CW_USEDEFAULT ;
mdicreate.cy          = CW_USEDEFAULT ;
mdicreate.style       = 0 ;
mdicreate.lParam      = NULL ;
hwndChild = (HWND) SendMessage (hwndClient,WM_MDICREATE, 0,
                                (long)(LPCREATESTRUCT) &mdicreate) ;

```

จากตัวแปรโครงสร้างที่ต้องผ่านให้โคลเอนต์ด้านบน พบว่าพารามิเตอร์ที่จำเป็นต่อการสร้างวินโดว์ลูกมีครบถ้วน ดังเช่นตัวแปร mdicreate.szClass ใช้อ้างถึงชื่อคลาสที่ต้องการใช้เป็นวินโดว์ลูกของโคลเอนต์วินโดว์ ตัวแปร mdicreate.szTitle ใส่ชื่อไตเติลของวินโดว์ลูก mdicreate.hOwner ใช้เก็บอินสแตนซ์ของแอปพลิเคชัน ตัวแปร mdicreate.x mdicreate.y mdicreate.cx และ mdicreate.cy เก็บตำแหน่งและขนาดของวินโดว์ที่ต้องการให้ปรากฏ โดยที่

x	แทนตำแหน่ง x ของมุมบน-ซ้ายของเอ็มดีไอไฮลด์วินโดว์
y	แทนตำแหน่ง y ของมุมบน-ซ้ายของเอ็มดีไอไฮลด์วินโดว์
cx	ค่าความกว้างเริ่มต้นของเอ็มดีไอไฮลด์วินโดว์
cy	ค่าความสูงเริ่มต้นของเอ็มดีไอไฮลด์วินโดว์

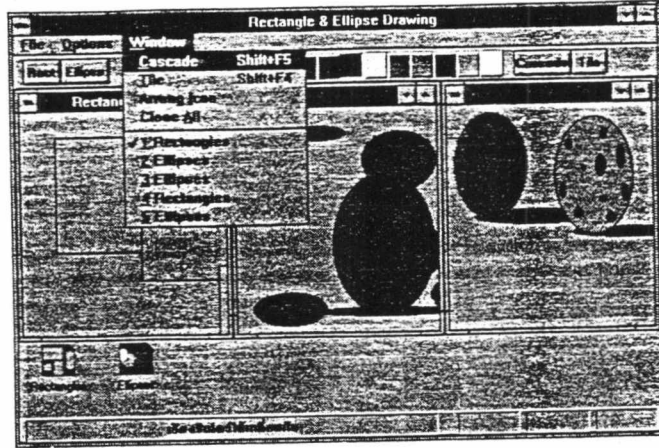
mdicreate.style เป็นรูปแบบของวินโดว์ในเวอร์ชัน 3.1 สามารถเลือกใช้รูปแบบได้เต็มที่ หากต้องการรูปแบบที่กำหนดให้โดยอัตโนมัติให้กำหนด 0 ไปให้ และตัวแปรตัวสุดท้าย mdicreate.lParam เป็นพารามิเตอร์ที่จะผ่านไปให้เมสเสจ ของวินโดว์ลูกทำงาน หากวินโดว์ไม่มีการรับค่าตัวแปรดังกล่าวให้กำหนดค่าเป็น NULL



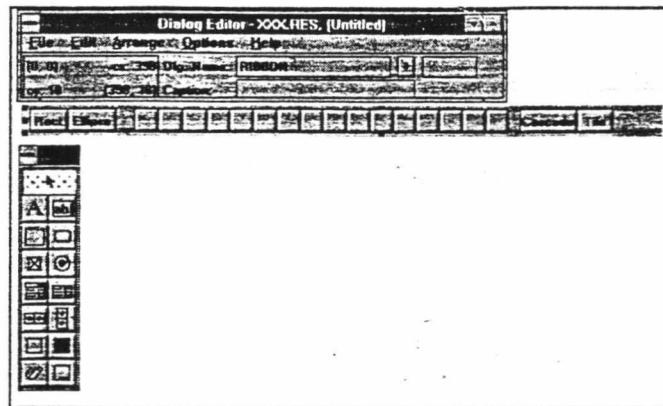
รูปที่ 2.2.4 ตัวอย่างแอปพลิเคชันเอ็มดีไอ

วินโดว์ลูกในแอปพลิเคชันเอ็มดีไอ มีลักษณะคล้ายกับวินโดว์หลักตามปกติทั่วไปมาก ที่ประกอบด้วยระบบย่อขนาดให้เล็กที่สุด, ขยายขนาดให้ใหญ่ที่สุด และสามารถเปลี่ยนแปลงขนาดได้ แต่มีข้อแตกต่างเพียงอย่างเดียว คือสามารถเคลื่อนย้ายวินโดว์ลูกจากเฟรมวินโดว์ได้ นอกจากนี้แต่ละวินโดว์ลูกภายในเฟรมวินโดว์ ยังสามารถแสดงได้ในลักษณะแบบไอคอน (ICON)

ในตอนต้นที่กล่าวมาทั้งหมด เป็นเรื่องของ การเขียนแอปพลิเคชันเอ็มดีไอ ซึ่งได้อธิบายสิ่งแวดล้อมพื้นฐาน อันได้แก่ เฟรมวินโดว์ ซึ่งเป็นวินโดว์หลักของระบบเอ็มดีไอ และไคลเอนต์วินโดว์ที่ถูกสร้างมาจากคลาส MDICLIENT ทำหน้าที่จัดการวินโดว์ลูกในสไตล์แบบเอ็มดีไอ และไคลเอนต์วินโดว์หรือวินโดว์ลูก ที่ต้องเขียนขึ้นมาให้ทำงานภายใต้สิ่งแวดล้อมแบบเอ็มดีไอ ดังนั้นในส่วนต่อไปจะพิจารณาเพิ่มเติมถึงสิ่งแวดล้อมเอ็มดีไอที่สร้างขึ้นใหม่ตามสมัยนิยม ซึ่งได้แก่ ทูลบาร์ สเตตัสบาร์ และเมนูเฮลป์ ดังแสดงให้เห็นในรูปที่ 2.2.5



รูปที่ 2.2.5 แอปพลิเคชันเอ็มดีไอ ที่เพิ่มทูลบาร์ สเตตัสบาร์ และเมนูเฮลป์



รูปที่ 2.2.6 ไดอะล็อกเอดิเตอร์กับการออกแบบทูลบาร์

## 2.2.4 ทูลบาร์

สำหรับแอปพลิเคชันสมัยใหม่ ได้มีการใช้ทูลบาร์กันเป็นจำนวนมาก ซึ่งอาจจะเรียกได้ว่าเป็นมาตรฐาน สำหรับการเชื่อมต่อแบบหลายเอกสารก็ว่าได้ หน้าที่หลักของทูลบาร์คือ เพื่อให้อ่านความสะดวกในด้านความเร็ว สำหรับการสั่งโปรแกรมให้ทำงานเช่นเดียวกับวิธีการใช้คีย์ด่วน ซึ่งจะใช้วิธีการสั่งงานจากรูปภาพ โดยที่รูปภาพสามารถสื่อความหมายได้ด้วย

การรวบรวมรูปภาพนั้น จะมีการออกแบบที่แตกต่างกันไป ขึ้นอยู่กับแต่ละบริษัทที่ผลิต เช่น ไมโครซอฟท์รวบรวมรูปภาพไว้บนบาร์ที่อยู่ใต้เมนูบาร์ลงมา เรียกว่า ทูลบาร์ หรือบางบริษัทเรียกว่า ริบบอน ถ้าเป็นการรวบรวมไว้ในบล็อกสี่เหลี่ยมย้ายไปไหนก็ได้เรียกว่า ทูลบ็อกซ์ สمارตไอคอน ฯลฯ

เทคนิคการสร้างทูลบาร์ที่พบเห็นโดยส่วนมากมีอยู่ 2 แบบด้วยกันคือ

1. ใช้วินโดว์และคอนโทรลมาตรฐาน วิธีนี้ใช้วินโดว์หรือว่าคอนโทรลมาตรฐานขนาดเล็ก ๆ เช่น ปุ่มนูน ( button ) แปะเข้ากับแท่งบาร์ซึ่งเป็นวินโดว์เหมือนกัน เป็นวิธีที่ง่าย เขียนโปรแกรมน้อย สามารถใช้ได้อะลือกเอดิเตอร์เป็นตัวออกแบบได้

ข้อเสีย คือ อิงกับระบบวินโดวส์ ใช้หน่วยความจำมาก ทำงานช้า

2. ควบคุมเองทั้งหมด วิธีนี้เป็นวิธีที่ยากกว่า คือ จะต้องจัดการวาดรูปปุ่มต่าง ๆ ด้วยลายเส้นเอง จัดการรับเมสเสจเม้าส์เองทั้งหมด ต้องเขียนโปรแกรมออกแบบขึ้นมาเองด้วย เช่น โปรแกรมประเภทโลดัส

ข้อดี คือ ใช้หน่วยความจำน้อยกว่า ทำงานเร็ว ไม่อิงกับระบบวินโดวส์ สามารถจัดเป็น DLL มาตรฐานเพื่อใช้ในแอปพลิเคชันตัวถัดไปได้

ตัวอย่างขั้นตอนการสร้างทูลบาร์ โดยใช้คอนโทรลมาตรฐานแปะเข้ากับแท่งบาร์สามารถทำได้ดังนี้

1. ออกแบบทูลบาร์โดยใช้ไดอะลือกเอดิเตอร์ ทำให้ได้ไดอะลือกกรีซอร์สออกมา
2. ส่งไดอะลือกกรีซอร์สให้วินโดวส์เป็นตัวสร้างแท่งบาร์ และแปะคอนโทรลมาตรฐานดังกล่าวลงแท่งบาร์ โดยใช้ความสามารถของไดอะลือกแมนเนเจอร์ ที่ติดมากับวินโดวส์เป็นตัวตีความกรีซอร์ส

3. ควบคุมการทำงานที่จำเป็นบนแท่งบาร์

### 2.2.5 สเตตัสบาร์

จากการพิจารณาโครงสร้างทางกายภาพ ของแอปพลิเคชันเอ็มดีไอประกอบ ด้วยเฟรมวินโดว์ถูกซ้อนทับโดยไคลเอนต์วินโดว์ และวินโดว์ลูกก็จะซ้อนทับไคลเอนต์ วินโดว์อีกที ถ้าหากย่อขนาดไคลเอนต์วินโดว์ ให้มีความกว้างสั้นลงก็จะเห็นพื้นที่เล็ก ๆ ของเฟรมวินโดว์แสดงให้เห็น ซึ่งเป็นบริเวณที่ใช้วาดเป็นสเตตัสบาร์

การทำสเตตัสบาร์ สามารถทำได้เช่นเดียวกับการทำทูลบาร์ คือมีได้หลายวิธี ที่นิยมคือสร้างวินโดว์มารองรับเหมือนกับทูลบาร์

ข้อดี คือ ในขณะที่มีการเลือกเมนู สเตตัสบาร์จะเป็นเครื่องมือที่ช่วยอธิบาย การทำงานของเมนูที่ถูกเลือก ทำให้ทราบว่าขณะนี้เครื่องกำลังทำงานอยู่ที่ส่วนใด หรือกำลัง ใช้คำสั่งใดอยู่

## บทที่ 3

### การออกแบบระบบ

#### 3.1 บทนำ

จุดที่ดอสต้องปรับปรุงหากมองในมุมมองของโปรแกรมเมอร์แล้ว คงจะไม่พ้นสองจุดหลัก คือ ระบบกราฟิกและระบบการติดต่อกับผู้ใช้ ที่ทำให้วินโดวส์เข้ามามีบทบาทมากขึ้น กอปรกับความสามารถที่เพิ่มขึ้นมาอีกหลาย ๆ อย่างทำให้การเขียนโปรแกรมง่ายขึ้นมาก สรุปความสามารถเด่น ๆ ของวินโดวส์ ได้ดังนี้

- ระบบการติดต่อกับผู้ใช้แบบกราฟิก อันประกอบด้วย วินโดวส์, เมนู, กรอบข้อความ ( dialog box ), และคอนโทรล ( controls ) ชนิดต่าง ๆ
- ระบบรับข้อมูลเข้าแบบใหม่ ที่สะดวกกว่าเดิม
- ระบบกราฟิกสำหรับอุปกรณ์ทุก ๆ อย่าง ที่สามารถแสดงกราฟิกได้
- ระบบจัดการงานแบบระบบหลายภารกิจ ( multitasking )
- การเชื่อมโยงและการใช้ข้อมูลร่วมกันระหว่างแอปพลิเคชัน

ในการเขียนโปรแกรมบนดอสโปรแกรมเมอร์ภาษาซี ส่วนมากมักจะติดต่อกับอินพุทหรือเอาพุท (I/O) และ หน่วยความจำ โดยใช้ฟังก์ชันมาตรฐานในไลบรารีภาษาซี ฟังก์ชันเหล่านั้นจะมอง I/O เป็นอุปกรณ์แบบตัวหนังสือเท่านั้น ส่วนการติดต่อกับหน่วยความจำ ก็จะติดต่อโดยตรงอย่างอิสระ การกระทำเหล่านี้ จะทำไม่ได้ในวินโดวส์ ที่ซึ่งแอปพลิเคชันแต่ละแอปพลิเคชัน จะต้องปันส่วนสิ่งต่าง ๆ ใช้ร่วมกัน เช่น คีย์บอร์ด, หน่วยความจำ, พอร์ตต่าง ๆ จอภาพ หรือแม้กระทั่งตัวประมวลผล หรือ ซีพียู ( CPU ) โดยจะถือว่าสิ่งเหล่านี้เป็นทรัพยากร ( resource ) ที่ต้องแบ่งกันใช้

ความแตกต่างของการเขียนโปรแกรมระหว่างดอสกับวินโดวส์ ทำให้โปรแกรมเมอร์ที่คุ้นเคยกับดอส จะต้องเปลี่ยนลักษณะการเขียนโปรแกรมเสียใหม่ สำหรับในระยะแรกก็อาจจะรู้สึกไม่คุ้นเคยบ้าง

ความแตกต่างที่เห็นได้ชัด ระหว่างการเขียนโปรแกรมบนดอสและวินโดวส์ ได้แก่

1. ระบบติดต่อกับผู้ใช้
2. ระบบการรับข้อมูลเข้า
3. ระบบกราฟิก
4. ระบบจัดการงาน

ซึ่งมีรายละเอียดดังนี้

1. ระบบติดต่อกับผู้ใช้ จุดประสงค์หลักอันหนึ่งของการออกแบบวินโดว์ ก็เพื่อให้ผู้ใช้สามารถใช้งานแอปพลิเคชันหลาย ๆ แอปพลิเคชันพร้อมกัน รวมทั้งยังให้ผู้ใช้สามารถมองเห็นการทำงานของทุก ๆ แอปพลิเคชันได้พร้อมกันอีกด้วย ด้วยเหตุที่ว่าใช้งานหลาย ๆ แอปพลิเคชันพร้อมกันนั้น แต่ละแอปพลิเคชันจึงต้องแบ่งปันจอภาพ เพื่อแสดงผลต่อผู้ใช้ สำหรับระบบงานที่คล้ายกันบางระบบ จะเจาะจงจอภาพทั้งจอ ให้แก่แอปพลิเคชันเดียวใช้งาน การแสดงผลของแอปพลิเคชันอื่น ๆ จะถูกซ่อนไว้ข้างหลัง ( หรือ ที่เรียกว่า Background Process ) แต่ในระบบวินโดว์ จะแตกต่างออกไป แอปพลิเคชันที่ทำงานบน วินโดว์ ทุก ๆ ตัว จะได้ใช้จอภาพแสดงผลกันทั่วหน้า ใหญ่บ้าง เล็กบ้าง ตามแต่ผู้ใช้จะต้องการ และสามารถแสดงผลให้แก่ผู้ใช้ได้ตลอดเวลา

แต่ละแอปพลิเคชัน จะใช้จอภาพแสดงผลร่วมกับแอปพลิเคชันอื่น ๆ โดยใช้งานเฉพาะส่วนที่อยู่ในกรอบสี่เหลี่ยมของตน ที่เรียกกันว่า วินโดว์ ( Window ) หากพิจารณาอย่างง่าย ๆ วินโดว์จะเป็นเพียงรูปสี่เหลี่ยมที่มีกรอบธรรมดาเท่านั้น แต่ที่จริงแล้ว วินโดว์ไม่เพียงแต่ จะเป็นอุปกรณ์แสดงผลชั้นพื้นฐานของแอปพลิเคชันเท่านั้น

แต่ยังเป็นสิ่งที่ผู้ใช้สามารถติดต่อกับแอปพลิเคชันได้เป็นอย่างดี โดยผ่านส่วนประกอบต่าง ๆ ของวินโดว์ เช่น เมนู ปุ่มบังคับ สโครลล์บาร์ ( Scroll bar ) เป็นต้น

เนื่องจากวินโดว์ มีวินโดว์ให้แอปพลิเคชันใช้หลายแบบ วินโดว์จึงไม่สร้างวินโดว์ล่วงหน้า ให้กับแอปพลิเคชันใด ๆ แอปพลิเคชันเหล่านั้นจะต้องสร้างวินโดว์ของตน ในแบบที่ตนต้องการเอง ซึ่งต่างกับคอสที่เตรียมจอภาพพื้นฐานให้ใช้ได้ทันที แต่วินโดว์ก็จะมีระบบจัดการพื้นฐาน สำหรับวินโดว์อำนวยความสะดวกสำหรับผู้ใช้อยู่แล้ว เช่น การเคลื่อนย้าย การเปลี่ยนขนาด การลดขนาดให้เป็นไอคอน เป็นต้น

ข้อได้เปรียบของวินโดว์อีกประการ ก็คือ จากการที่แต่ละแอปพลิเคชันมีวินโดว์เป็นของตนเอง จึงสามารถแน่ใจได้เลยว่า แอปพลิเคชันจะไม่แย่งกันใช้งานจอภาพ จนเกิดความสับสนในการใช้งาน

2. ระบบการรับข้อมูลเข้าแบบใหม่ ในระบบคอสธรรมดา เมื่อแอปพลิเคชันจะรับข้อมูลจากผู้ใช้ เช่น ผ่านทางฟังก์ชัน Getchar() ก็จำเป็นต้องรอจนกว่าผู้ใช้จะใส่ข้อมูลมาจริง ๆ เสียก่อน หากในระหว่างนั้นผู้ใช้จะใส่ข้อมูลทางอื่น เช่น

แมส ก็จะไม่สามารถรับข้อมูลเข้ามาได้ แต่ในวินโดว์จะแตกต่างออกไป วินโดว์จะรวบรวมข้อมูลเข้าจากทุก ๆ อุปกรณ์ แล้วเก็บไว้ใน คิวแมสเสจ ( Message Queue ) เมื่อแอปพลิเคชันต้องการอ่านข้อมูลเข้า ก็เพียงแต่อ่านจากแมสเสจคิวเท่านั้น แอปพลิเคชันจะได้รับข้อมูลจากผู้ใช้ โดยไม่มีการสูญหายของข้อมูล

ข้อมูลที่อยู่ในคิวแมสเสจ จะอยู่ในรูปของแมสเสจ ที่มีขนาดในการเก็บเท่ากันและมีรูปแบบการใช้งานคล้ายกัน ในการอ่านค่าแต่ละครั้งการพิจารณาข้อมูลก็จะใช้โปรแกรมลักษณะที่คล้ายกัน ทำให้การเขียนแอปพลิเคชันง่ายและสะดวกขึ้น

การอ่านข้อมูลจากอุปกรณ์ต่าง ๆ ก็เช่นกัน ในระบบวินโดว์ การอ่านค่าจากแมสเสจคิว ใช้ฟังก์ชันเพียงฟังก์ชันเดียวเท่านั้น คือ GetMessage() แต่ในดอสจะยุ่งยากกว่ามาก คือ หากเราต้องการอ่านค่าจากคีย์บอร์ดเป็นตัวหนังสือ ก็ต้องใช้ฟังก์ชัน Getchar() หากต้องการตรวจสอบการกดคีย์บอร์ดธรรมดา ก็ต้องใช้ฟังก์ชัน keypressed() การที่ต้องใช้หลายฟังก์ชันในการทำงานที่มีลักษณะคล้ายกันนั้น เมื่อเทียบกับการใช้ฟังก์ชัน GetMessage() เพียงฟังก์ชันเดียว จะเห็นได้ชัดว่าการเขียนแอปพลิเคชันบน วินโดว์ นั้นเรียบง่ายและสะดวกสบายกว่ามาก

นอกจากนี้ข้อมูลที่เกิดจากผู้เลือกเมนู หรือปุ่มบังคับต่าง ๆ ก็ยังบรรจุเอาไว้ในแมสเสจคิวด้วย ดังนั้นการรับรู้การกระทำทั้งหมดของผู้ใช้ทุกอย่าง จึงใช้เพียงฟังก์ชัน GetMessage() เพียงฟังก์ชันเดียว เท่านั้น

3. ระบบกราฟิก จากระบบดอสที่ไม่เคยมีการสนับสนุนระบบกราฟิกเลย มาสู่วินโดว์ ที่เต็มไปด้วยคำสั่ง และวิธีการอันหลากหลาย ซับซ้อน และมีประสิทธิภาพสำหรับการจัดการกับระบบกราฟิก ไม่ว่าจะเป็เส้นตรง วงกลม รูปเหลี่ยม รูปทึบหรือเส้นโค้ง ไม่เพียงเท่านั้น คำสั่งทุกคำสั่ง ยังสามารถใช้ได้กับอุปกรณ์แสดงผลทุกชนิด ที่สามารถแสดงผลเป็นกราฟิกได้ สามารถใช้คำสั่งเดียวกันรูปแบบเดียวกันวาดรูปต่าง ๆ ได้ทั้งบนจอภาพ เครื่องพิมพ์ ( กราฟิก ) พล็อตเตอร์ ฯลฯ

วินโดว์ได้อาศัยโปรแกรมขับอุปกรณ์ ( device driver ) ในการแสดงผลให้เหมาะสมกับอุปกรณ์ที่ใช้แต่ละแบบ โปรแกรมขับอุปกรณ์ อาจเรียกว่า พื้นผิวอุปกรณ์ ( device context ) และการแสดงผลกราฟิกทุกครั้ง จะต้องกระทำผ่านพื้นผิวอุปกรณ์นี้เสมอ

4. ระบบจัดการงาน ในการทำงานในดอสนั้น จะไม่มีระบบจัดการปันส่วนทรัพยากรใด ๆ เมื่อแอปพลิเคชันหนึ่ง ๆ ทำงาน ก็ถือเสมือนว่าแอปพลิเคชันนั้น

เป็นเจ้าของทรัพยากรทั้งหมด เช่น คีย์บอร์ด จอภาพ หน่วยความจำ ซีพียู แต่สำหรับวินโดวส์ จะมีการบังคับให้แอปพลิเคชัน มีการปันส่วนทรัพยากรต่าง ๆ ร่วมกันเสมอ ซึ่งต่างจากแอปพลิเคชันในดอส ที่แอปพลิเคชันสามารถนำหน่วยความจำที่ไม่ได้ยึดครองโดยดอส นำไปใช้ได้ทั้งหมด โดยไม่ต้องคำนึงถึงแอปพลิเคชันอื่น ๆ

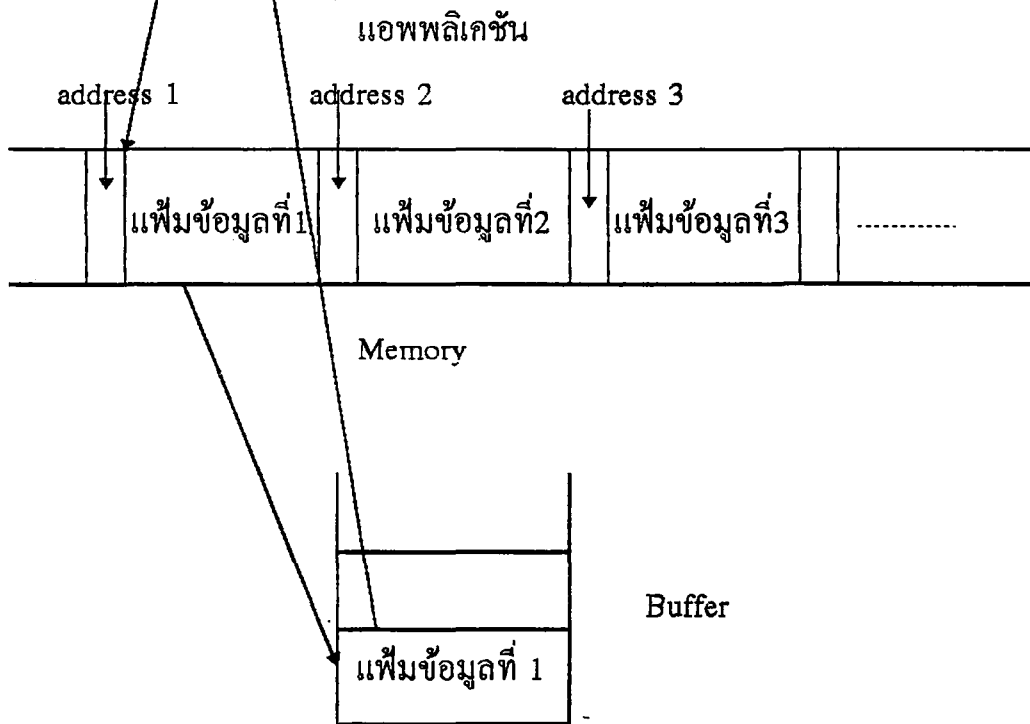
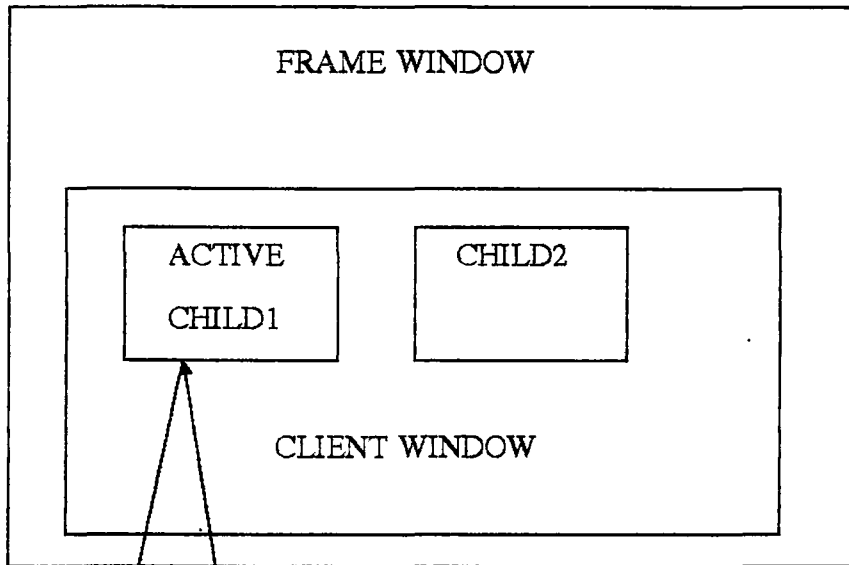
แต่ในวินโดวส์ ซึ่งเป็นระบบหลายภารกิจ (Multitasking) นั้น แตกต่างออกไป พื้นที่หน่วยความจำ ซึ่งเป็นทรัพยากรอันจำกัดนั้น จะต้องปันส่วนกันใช้ หากทุกแอปพลิเคชันต่างใช้อย่างสิ้นเปลือง หน่วยความจำอันจำกัดก็จะหมดไปอย่างรวดเร็ว จึงทำให้แอปพลิเคชันต่าง ๆ ทำงานพร้อมกันได้ไม่มาก เพียงหนึ่งหรือสองแอปพลิเคชัน ฉะนั้นในการทำงานจริง ๆ ของวินโดวส์ จึงมีการเคลื่อนย้ายสลับเปลี่ยน (swap) หรือทิ้ง (discard) ส่วนต่าง ๆ ของหน่วยความจำตลอดเวลา เพื่อให้การทำงานทั้งระบบมีประสิทธิภาพสูงสุด การใช้หน่วยความจำแต่ละครั้ง มักจะพบว่ามีอยู่หลายครั้งที่ข้อมูลที่ใส่เข้าไป จะไม่อยู่ในตำแหน่งแอสเดรส (Address) เดิม

การจัดจอกภาพที่จะต้องแบ่งกันใช้ในทุก ๆ แอปพลิเคชัน ในระบบดอสจะยกจอกภาพทั้งจอ ไปขึ้นอยู่กับการจัดการของแอปพลิเคชัน เพียงแอปพลิเคชันเดียว ไม่ว่าแอปพลิเคชันนั้นจะทำการดัดแปลงจอกภาพอย่างไร เช่น เปลี่ยนความละเอียดในการแสดงผล เปลี่ยนสีที่ใช้ในการแสดงผล ฯลฯ แต่ในระบบวินโดวส์จะเปลี่ยนไป แอปพลิเคชันจะต้องใช้จอกภาพร่วมกัน เพราะฉะนั้นแอปพลิเคชันที่ทำงานบนวินโดวส์ จะต้องไม่มีการดัดแปลงการใช้งานจอกภาพโดยตรง ต้องมีการขอใช้โดยผ่านวินโดวส์เสมอ

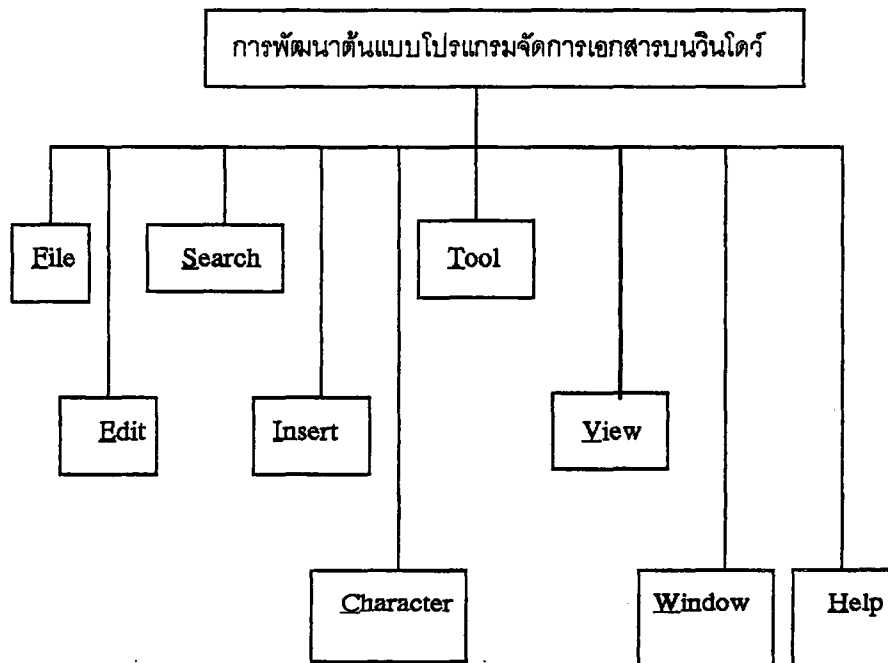
ด้วยเหตุผลเหล่านี้ จึงทำให้ผู้พัฒนาโปรแกรมหันมาสนใจแอปพลิเคชันบนวินโดวส์ โดยในบทนี้ จะกล่าวถึง การออกแบบระบบโดยทั่ว ๆ ไป

### 3.2 โครงสร้างของระบบ

โครงสร้างโปรแกรมของการพัฒนาต้นแบบโปรแกรมจัดการเอกสารบนวินโดวส์ ประกอบด้วยโมดูลที่สำคัญดังแผนโครงสร้างโปรแกรม (Structure Chart) ดังรูป 3.1

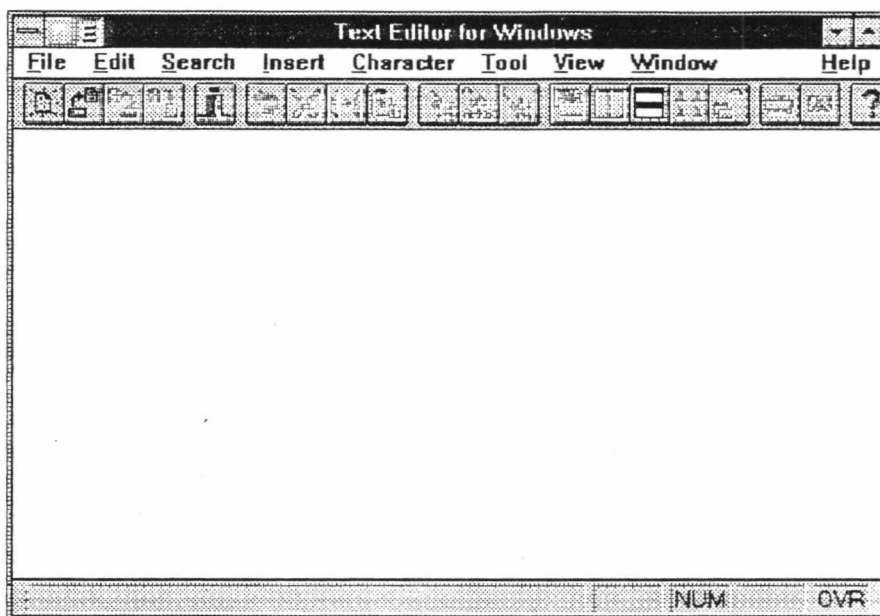


แสดงโครงสร้างภายในข้อมูล



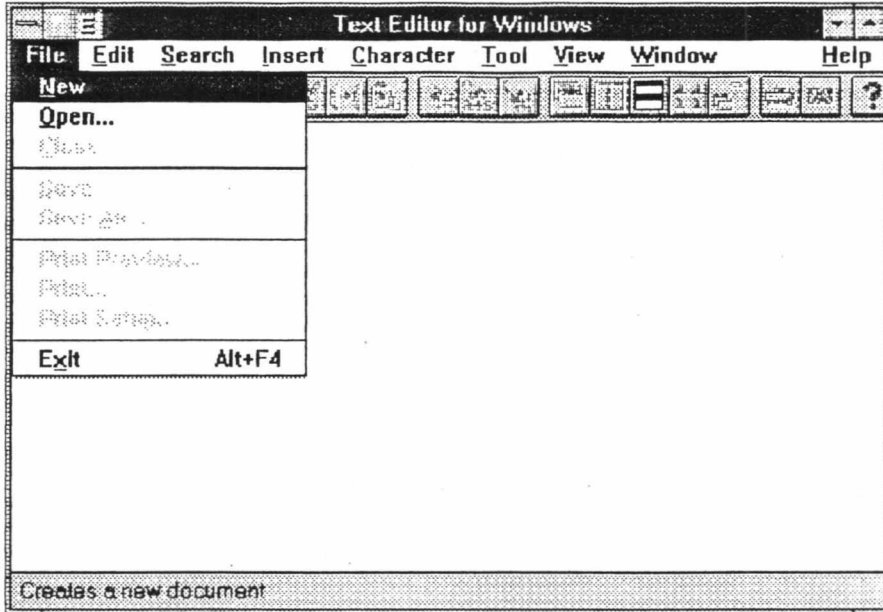
รูปที่ 3.1 แสดงโครงสร้างโปรแกรมการพัฒนาต้นแบบโปรแกรมจัดการเอกสารบนวินโดว

เมื่อทำการรันโปรแกรมจะปรากฏหน้าจอ ดังต่อไปนี้



รูปที่ 3.2 หน้าจอที่ได้จากการรันโปรแกรม WORD.EXE

จากหน้าจอโปรแกรมดังที่แสดงในรูป 3.2 นั้น ประกอบด้วยเมนู ซึ่งในแต่ละส่วนมีหน้าที่แตกต่างกัน ดังจะแสดงต่อไปนี้

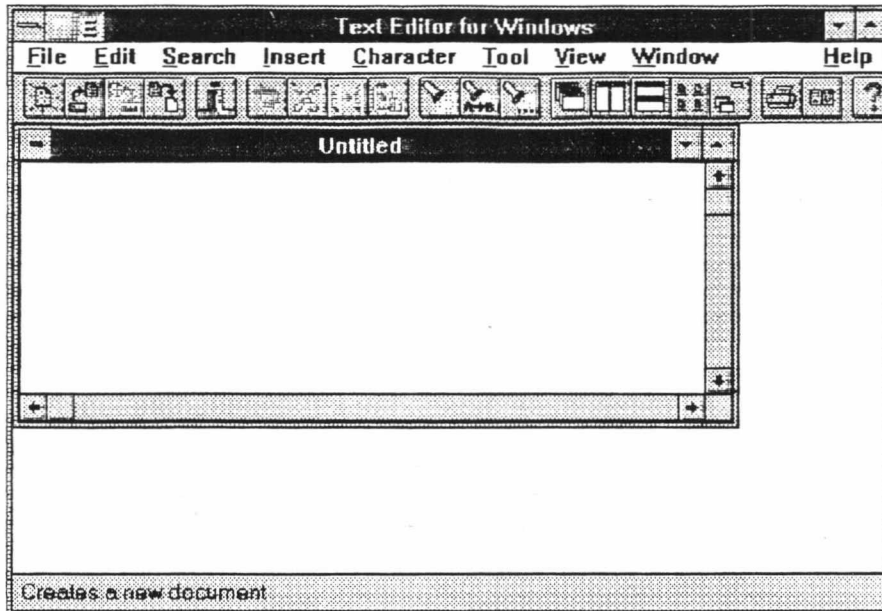


รูปที่ 3.3 ผลที่ได้จากการเลือกเมนูคำสั่ง File หรือ Alt + F

File มีหน้าที่เกี่ยวกับเพิ่มข้อมูล ดังนี้

- New สร้างเอกสารหรือแบบเอกสารใหม่

เมื่อเลือกเมนูย่อย New ( Submenu New ) ในเมนู File จะปรากฏหน้าจอ ดังรูป  
ต่อไปนี้เป็น



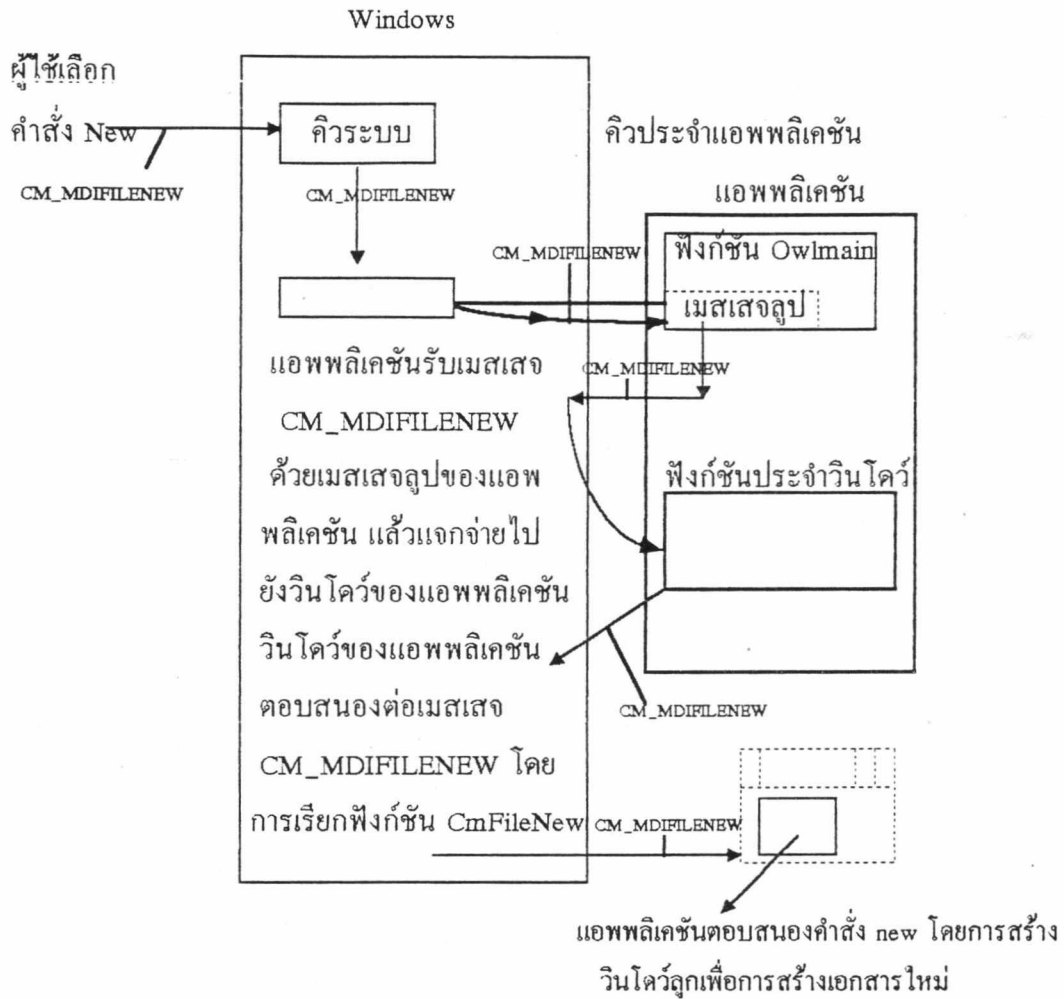
รูปที่ 3.4 ผลที่ได้จากการเลือกเมนู New ในเมนู File

จากคำสั่งนี้ จะทำการสร้างแฟ้มเอกสารใหม่ โดยแฟ้มเอกสารที่สร้างจะได้เป็นแฟ้มเอกสาร .TXT ซึ่งฟังก์ชันนี้มีคุณสมบัติการสืบทอด( inheritance ) มาจาก TDocManager Class ดังนี้

```
virtual void TDocManager :: CmFileNew( );
```

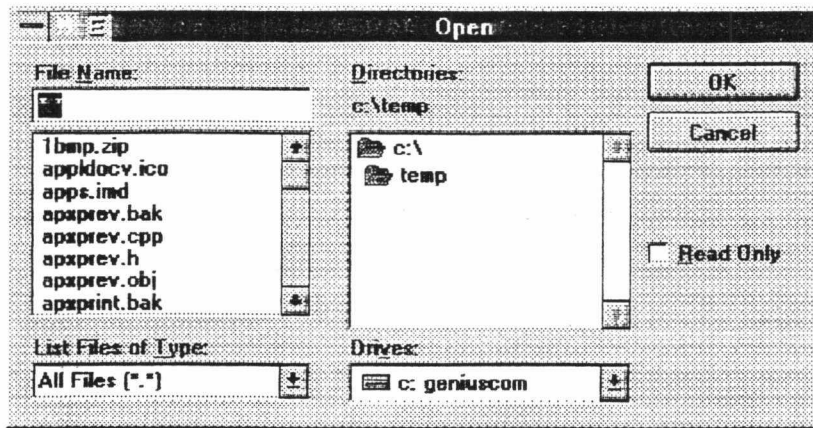
{ เครื่องหมาย ( :: ) เป็นเครื่องหมายแสดงการสืบทอด (scope resolution operator) }

ซึ่งฟังก์ชันนี้จะตอบสนองต่อเมสเสจ CM\_MDIFILENEW ซึ่งถูกส่งผ่านมาทางเมสเสจคิว ดังแสดงโครงสร้างการจัดการเมสเสจของวินโดว์ดังนี้



ภาพแสดงการจัดการเมสเสจของวินโดว์เมื่อผู้ใช้เลือกเมนู New

- Open เปิดเอกสารหรือแบบเอกสารที่มีอยู่แล้ว  
เมื่อเลือกเมนูย่อย Open จะปรากฏหน้าต่างดังรูปต่อไปนี้



รูปที่ 3.5 ผลที่ได้จากการเลือกเมนูคำสั่ง Open ในเมนู File เพื่อทำการเลือก File ที่จะทำให้  
การเปิดไฟล์

จากคำสั่งนี้ จะสามารถทำการเปิดแฟ้มเอกสารขึ้นมาแก้ไข ( Edit ) ได้ โดยเอกสารดังกล่าวต้องมีรูปแบบแฟ้มเอกสาร ( text file program ) เป็นสกุล .TXT ,.CW ซึ่งสร้างจากรหัสภาษาไทย สมอ. หรือเอกสารที่เป็น code program เช่น .C,.CPP,.PAS เป็นต้น ทำให้สามารถนำแฟ้มเอกสารสกุลต่างๆ ดังกล่าวขึ้นมาแก้ไขได้ โดยสามารถเปิดเอกสารขึ้นมาแก้ไขได้หลายเอกสารพร้อมกัน (MDI) โดยไม่สามารถเปิดเอกสารซ้ำชื่อเดียวกัน ถ้าซ้ำจะมีกรอบข้อความแสดงข้อผิดพลาด ซึ่งฟังก์ชันนี้เป็น Public Member Function ของ TDocmanager Class ดังนี้

```
virtual void TDocmanager :: CmFileOpen( );
```

ซึ่งฟังก์ชันนี้จะรองรับเมสเสจ CM\_MDIFILEOPEN ที่ส่งเข้ามาโดยผ่านทางเมสเสจคิวของแอปพลิเคชัน

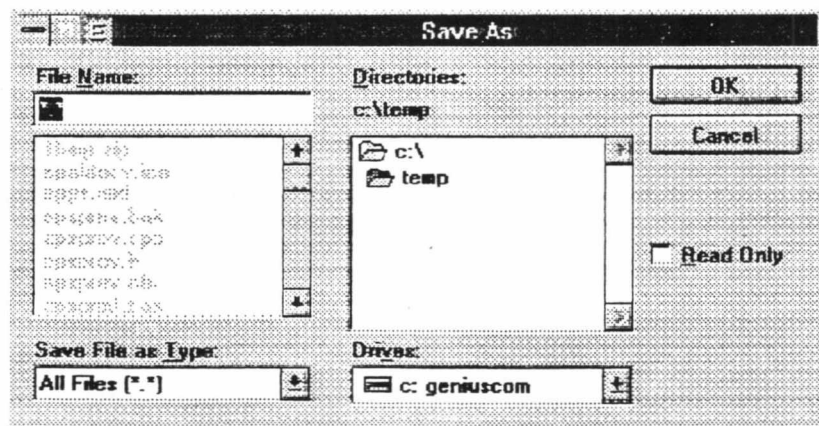
- Close ปิดทุกวินโดว์ของเอกสารปัจจุบัน

จากคำสั่งนี้จะทำการปิดเอกสารปัจจุบันที่เรียกขึ้นมาแก้ไขหรือต้องการยกเลิกการสร้างเอกสาร ซึ่งฟังก์ชันนี้เป็น Public Member Function ของ TDocmanager Class ดังนี้

```
virtual void TDocmanager :: CmFileClose( );
```

ซึ่งฟังก์ชันนี้จะรอรับเมสเสจ CM\_FILECLOSE ที่ส่งเข้ามาโดยผ่านทางเมสเสจคิวของแอปพลิเคชัน ผลที่ได้จากการเลือกคำสั่งนี้จะมีรูปแบบหน้าจอเหมือนดังรูปที่ 3.2

- Save บันทึกเอกสารหรือแบบเอกสารปัจจุบัน  
เมื่อเลือกเมนูย่อยนี้จะปรากฏหน้าต่างดังรูปต่อไปนี้



รูปที่ 3.6 ผลที่ได้จากการเลือกเมนูคำสั่ง Save หรือ Save As เพื่อต้องการให้ผู้ใช้บันทึกเพิ่มข้อมูลลงใน Path ที่ถูกต้อง

ใช้เมื่อต้องการบันทึกข้อมูลในเอกสารทั้งหมดโดยถ้าเป็นการสร้างเอกสารใหม่เมื่อเลือกเมนู Save จะมีการทำงานเหมือน Save As คือให้ใส่ชื่อเพิ่มเอกสารที่ต้องการ Save และถ้าเป็นการเปิดเอกสารขึ้นมาแก้ไขคำสั่งนี้จะทำการ Save ตามชื่อเอกสารเดิม ซึ่งฟังก์ชันนี้เป็น Public Member Function ของ TDocmanager Class ดังนี้

```
virtual void TDocmanager :: CmFileSave( );
```

ซึ่งฟังก์ชันนี้จะรองรับเมสเสจ CM\_FILESAVE ที่ส่งเข้ามาโดยผ่านทางเมสเสจคิวของ แอปพลิเคชัน

- Save As           บันทึกเอกสารสำรองในอีกแฟ้ม

เมื่อเลือกเมนูย่อยนี้จะปรากฏหน้าต่างดังรูปที่ 3.6

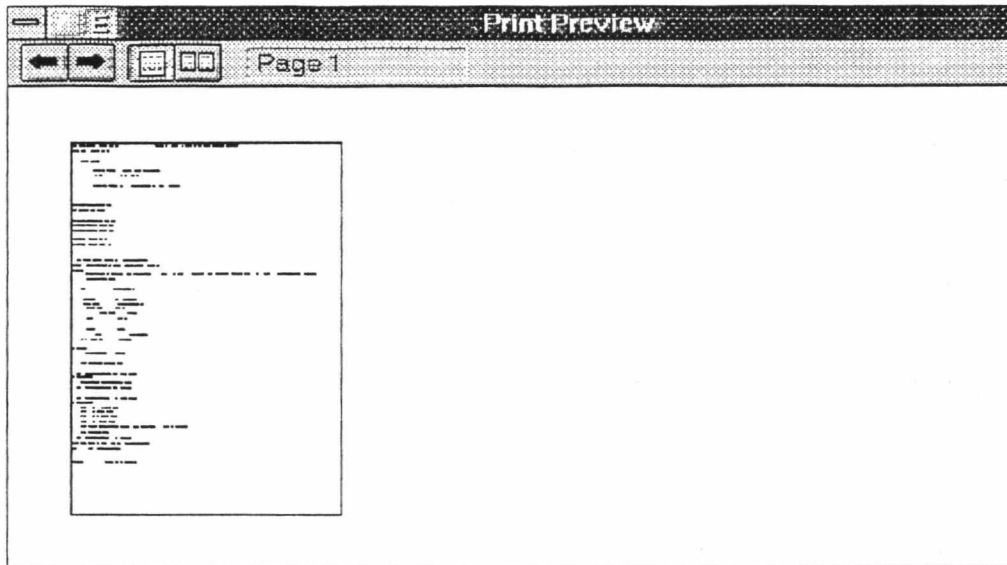
คำสั่งนี้ใช้เมื่อต้องการบันทึกข้อมูลในเอกสารทั้งหมด ตามที่ระบุในกรอบข้อความ Save As ( ดังรูปที่ 3.6 ) โดยถ้ามีการระบุชื่อซ้ำกับเอกสารที่มีอยู่แล้ว จะปรากฏกรอบข้อความแสดงคำเตือนให้ทราบ ก่อนที่จะตัดสินใจ Save เอกสารชื่อนั้นลงไปซึ่งฟังก์ชันนี้เป็น Public Member Function ของ TDocmanager Class ดังนี้

```
virtual void TDocmanager :: CmFileSaveAs( );
```

ซึ่งฟังก์ชันนี้จะรองรับเมสเสจ CM\_FILESAVEAS ที่ส่งเข้ามาโดยผ่านทางเมสเสจคิวของ แอปพลิเคชัน

- Print Preview    แสดงเอกสารเต็มหน้าเหมือนตอนพิมพ์

เมื่อเลือกเมนูย่อยนี้จะปรากฏหน้าต่างดังรูปต่อไปนี้

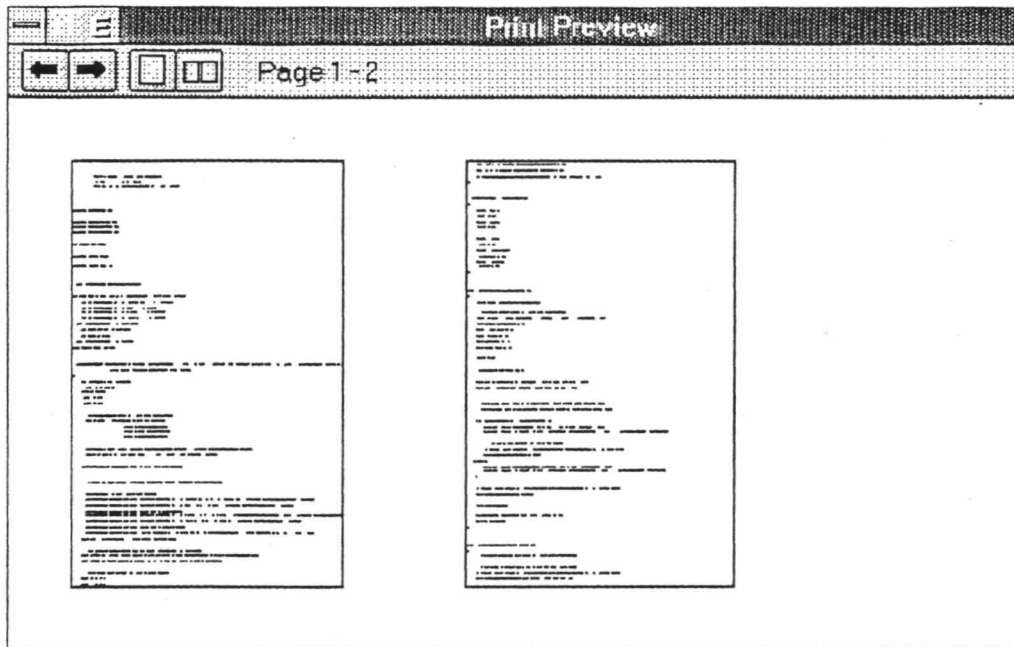


รูปที่ 3.7.1 ผลที่ได้จากคำสั่ง Print Preview แสดงผลหนึ่งหน้าจอ

เมนูนี้ใช้แสดงรูปแบบของเอกสารเต็มหน้าเหมือนตอนที่พิมพ์ สามารถแสดงรูปแบบเอกสารได้ทั้งแบบหน้าเดียวและแบบสองหน้า ทำให้ผู้ใช้สามารถจัดรูปแบบเอกสารได้ตามต้องการ โดยไม่ต้องเสียเวลาลองพิมพ์ออกทางเครื่องพิมพ์จริงๆ ซึ่งฟังก์ชันนี้เป็น Protected Member Function ของ hyperMDIClient Class ดังนี้

```
virtual void hyperMDIClient :: CmFilePrintPreview ( );
```

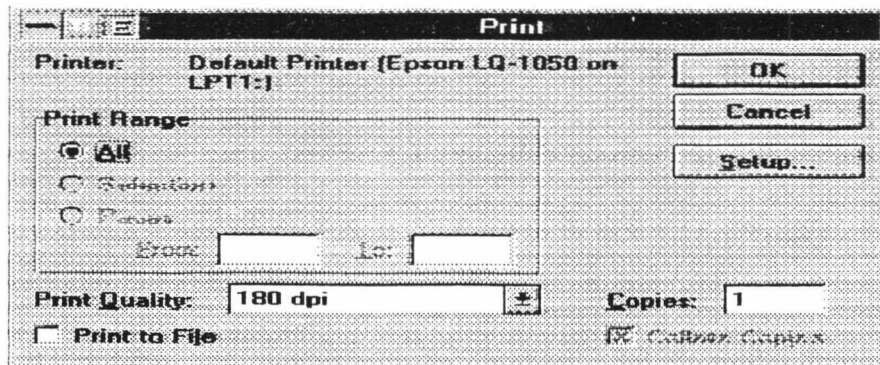
ซึ่งฟังก์ชันนี้จะรอรับเมสเสจ CM\_FILEPRINTPREVIEW ที่ส่งเข้ามาโดยผ่านทางเมสเสจคิวของแอปพลิเคชัน



รูปที่ 3.7.2 ผลที่ได้จากการเลือกเมนู Print Preview แบบแสดงผล 2 หน้าจอ

- Print พิมพ์เอกสารปัจจุบัน

เมื่อเลือกเมนูย่อยนี้จะปรากฏหน้าจอดังรูปต่อไปนี้



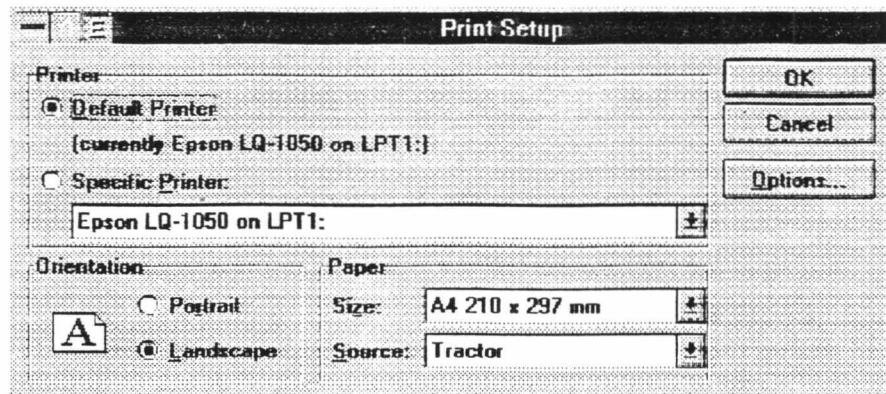
รูปที่ 3.8 ภาพแสดงหน้าจอที่ได้จากการเลือกคำสั่ง Print

เมนูนี้ใช้เมื่อต้องการพิมพ์เอกสารปัจจุบัน โดยเมื่อเลือกเมนูนี้จะปรากฏกรอบข้อความแสดงเครื่องพิมพ์ปัจจุบัน และช่วงของการพิมพ์ (Print Range) ซึ่งฟังก์ชันนี้เป็น Protected Member Function ของ hyperMDIClient Class ดังนี้

```
virtual void hyperMDIClient :: CmFilePrint ( );
```

ซึ่งฟังก์ชันนี้จะรองรับเมสเสจ CM\_FILEPRINT ที่ส่งเข้ามาโดยผ่านทางเมสเสจคิวของแอปพลิเคชัน

- Print Setup      ตั้งค่าเครื่องพิมพ์ปัจจุบัน  
เมื่อเลือกเมนูย่อยนี้จะปรากฏหน้าต่างดังรูปต่อไปนี้



รูปที่ 3.9 ภาพแสดงผลที่ได้จากการเลือกคำสั่ง Print Setup

เมนูนี้ใช้เมื่อต้องการตั้งค่าเครื่องพิมพ์ให้ตรงตามชนิดของเครื่องพิมพ์ที่ต้องการ ซึ่งฟังก์ชันนี้เป็น Protected Member Function ของ hyperMDIClient Class ดังนี้

```
virtual void hyperMDIClient :: CmFilePrintSetup ( );
```

ซึ่งฟังก์ชันนี้จะรอรับเมสเสจ CM\_FILEPRINTERSETUP ที่ส่งเข้ามาโดยผ่านทางเมสเสจคิวของแอปพลิเคชัน

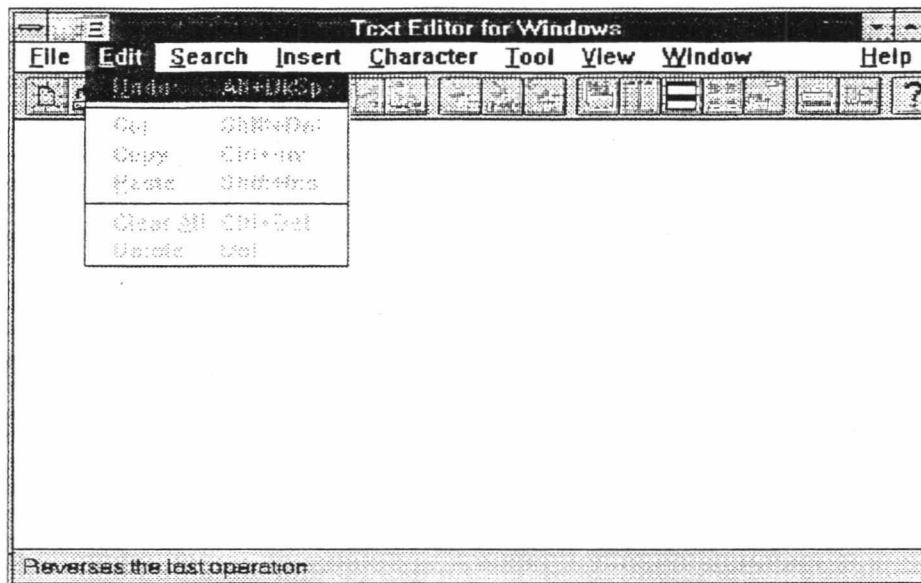
- Exit ยกเลิกการทำงานของระบบ และยืนยันการบันทึกเอกสารที่ยังไม่ได้บันทึกก่อนการยกเลิกการทำงาน ซึ่งฟังก์ชันนี้เป็น Public Member Function ของ TDocmanager Class ดังนี้

```
virtual void TDocmanager :: CmFileExit ( );
```

ฟังก์ชันนี้จะรอรับเมสเสจ CM\_FILEEXIT ที่ส่งเข้ามาโดยผ่านทางเมสเสจคิวของ

แอปพลิเคชัน ดังแสดงโครงสร้างการส่งเมสเสจ CM\_FILEEXIT เพื่อเป็นตัวอย่างให้เห็นถึง การส่งเมสเสจผ่านทางเมสเสจคิวของแอปพลิเคชันบนวินโดว

Edit มีหน้าที่จัดการเกี่ยวกับการปรับปรุงแก้ไขเพิ่มเอกสารดังนี้



รูปที่ 3.10 ผลที่ได้จากการเลือกเมนู Edit

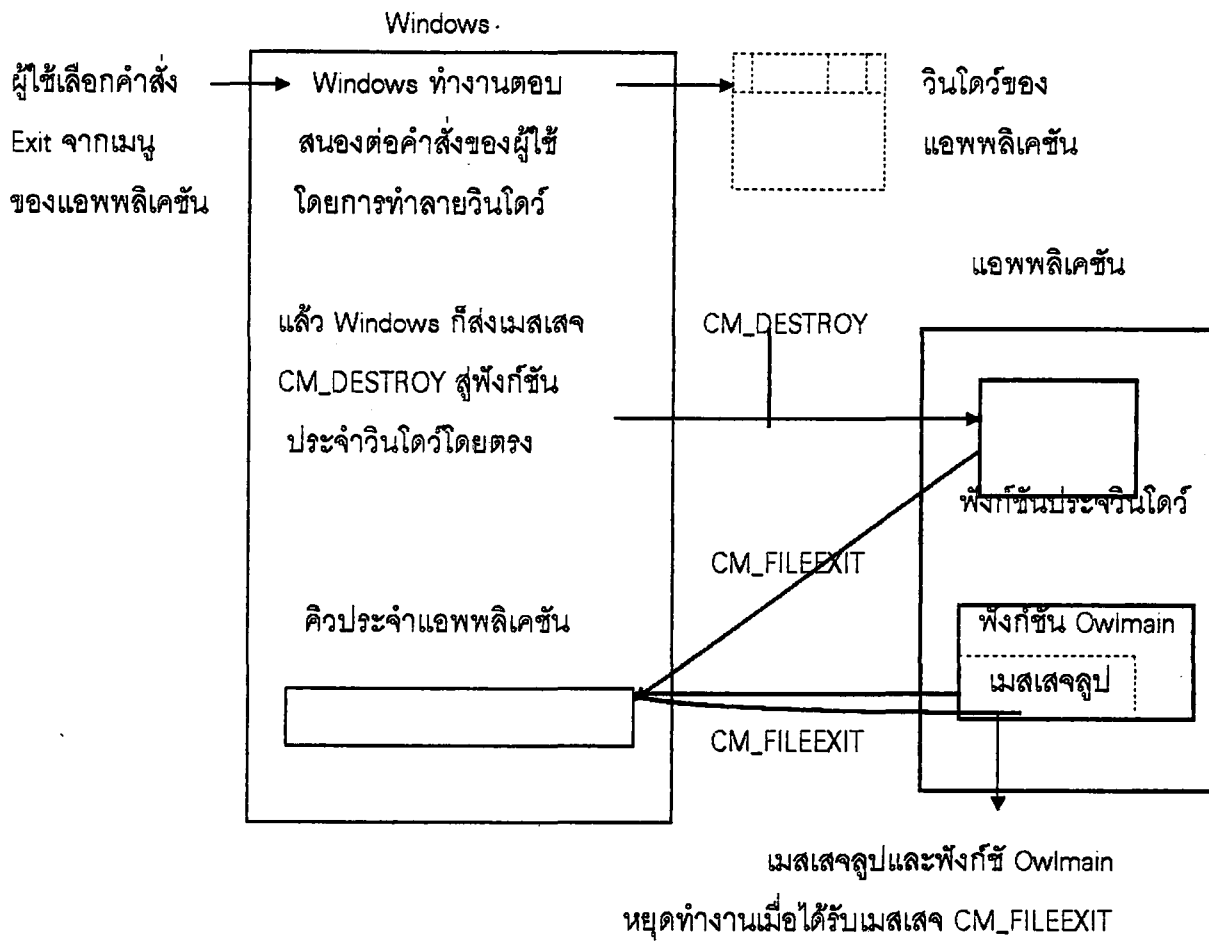
- Undo ทำซ้ำและยกเลิกการกระทำสุดท้าย

เมนูนี้ใช้เมื่อต้องการทำซ้ำคำสั่งเดิมและยกเลิกการกระทำครั้งสุดท้ายที่เคยกระทำมาแล้ว ซึ่งฟังก์ชันนี้เป็น Protected Member Function ของ TEdit Class โดยมีโครงสร้างการสืบทอด จาก TEdit Class ดังนี้

```
void TEdit::CmEditUndo ( );
```

ซึ่งฟังก์ชัน CmEditUndo ( ); นี้จะรอรับเมสเสจ CM\_EDITUNDO จากเมสเสจคิวของ แอปพลิเคชัน

- Cut ตัดส่วนที่ทำแถบสี ไว้และวางบนคลิปบอร์ด



### การจัดการเมสเสจของวินโดว์

เมนูนี้ใช้เมื่อต้องการย้ายข้อความที่ทำแถบสว่างไว้ไปยังส่วนอื่นของแฟ้มเอกสาร หรือ ต้องการตัดข้อความส่วนที่ทำแถบสว่างนี้ทิ้งไป โดยการทำงานของคำสั่งนี้ทำผ่านคลิปปอร์ด ( แสดงรายละเอียดไว้ท้ายบท ) ซึ่งฟังก์ชันนี้เป็น Protected Member Function ของ TEdit Class โดยมีโครงสร้างการสืบทอดจาก TEdit Class ดังนี้

```
void TEdit::CmEditCut ( );
```

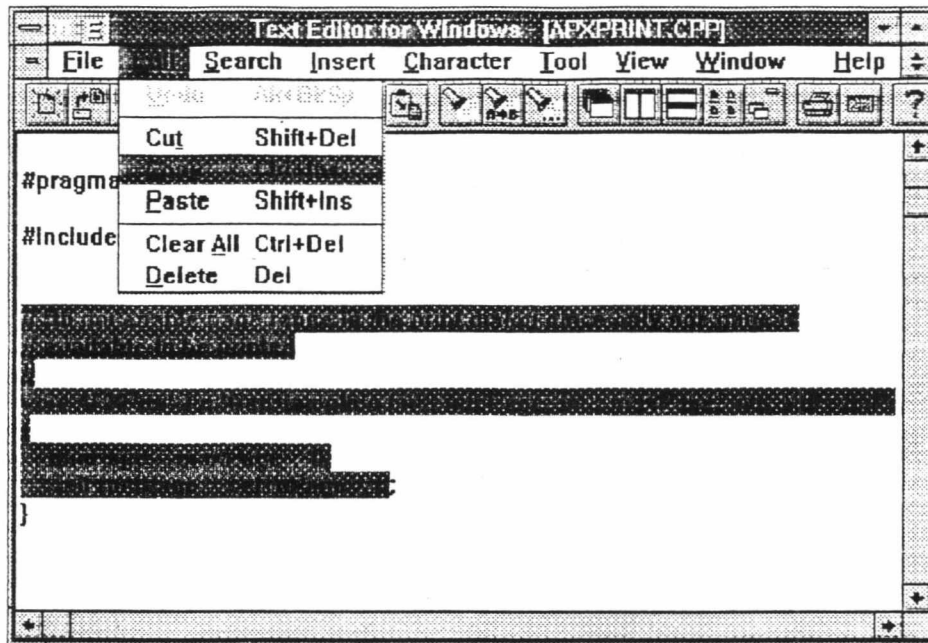
ซึ่งฟังก์ชัน CmEditCut นี้จะรองรับเมสเสจ CM\_EDITCUT จากเมสเสจคิวของ แอปพลิเคชัน

- Paste แทรกเนื้อหาที่อยู่บนคลิปปอร์ดในเอกสารปัจจุบัน  
เมนูนี้ใช้เมื่อต้องการนำเนื้อหาในคลิปปอร์ดมาไว้ในเอกสารตรงตำแหน่ง เคอร์เซอร์ ซึ่งฟังก์ชันนี้เป็น Protected Member Function ของ TEdit Class โดยมีโครงสร้างการ สืบทอดจาก TEdit Class ดังนี้

```
void TEdit::CmEditPaste ( );
```

ซึ่งฟังก์ชัน CmEditPaste นี้จะรองรับเมสเสจ CM\_EDITPASTE จากเมสเสจคิวของ แอปพลิเคชัน

- Copy คัดลอกส่วนที่ทำแถบสว่างไว้และวางไว้บนคลิปปอร์ด



รูปที่ 3.11 ภาพแสดงการทำแถบสว่างเพื่อเลือกข้อความที่ต้องการคัดลอก

เมนูนี้ใช้เมื่อต้องการคัดลอกส่วนที่ทำแถบสว่างไปยังส่วนอื่นของแฟ้มเอกสาร โดยการ  
ทำงานของคำสั่งนี้ทำผ่านคลิบบอร์ด ซึ่งเป็น Protected Member Function ของ TEdit Class  
โดยมีโครงสร้างการสืบทอดจาก TEdit Class ดังนี้

```
void TEdit::CmEditCopy ( );
```

ซึ่งฟังก์ชัน CmEditCopy นี้จะรอรับเมสเสจ CM\_EDITCOPY จากเมสเสจคิวของ  
แอปพลิเคชัน

- Clear All                    ลบเนื้อหาทั้งหมดในเอกสารปัจจุบัน

เมนูนี้ใช้เมื่อต้องการลบข้อความทั้งหมดในเอกสารปัจจุบันที่กำลังสร้าง หรือที่เปิดขึ้นมา  
แก้ไข ซึ่งฟังก์ชันนี้เป็น Protected Member Function ของ TEdit Class โดยมีโครงสร้างการสืบทอด  
จาก TEdit Class ดังนี้

```
void TEdit::CmEditClearAll ( );
```

ซึ่งฟังก์ชัน CmEditClearAll นี้จะรองรับเมสเสจ CM\_EDITCLEARALL จากเมสเสจคิวของ แอปพลิเคชัน

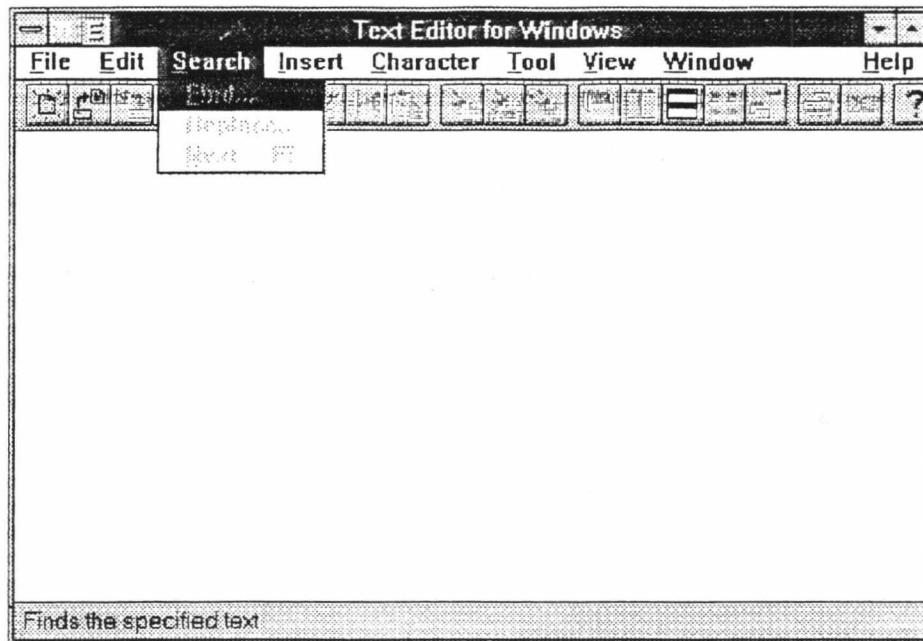
- Delete      ลบส่วนที่ทำแถบสว่างไว้โดยไม่ต้องใส่ลงในคลิปบอร์ด

เมนูนี้ใช้เมื่อต้องการลบข้อความเฉพาะส่วนที่ทำแถบสว่าง ออกจากแฟ้มเอกสาร โดยคำสั่งนี้ไม่ต้องทำผ่านคลิปบอร์ด แต่จะลบข้อมูลออกจากหน่วยความจำโดยตรง ซึ่งเป็น Protected Member Function ของ TEdit Class โดยมีโครงสร้างการสืบทอดจาก TEdit Class ดังนี้

```
void TEdit::CmEditDelete ( );
```

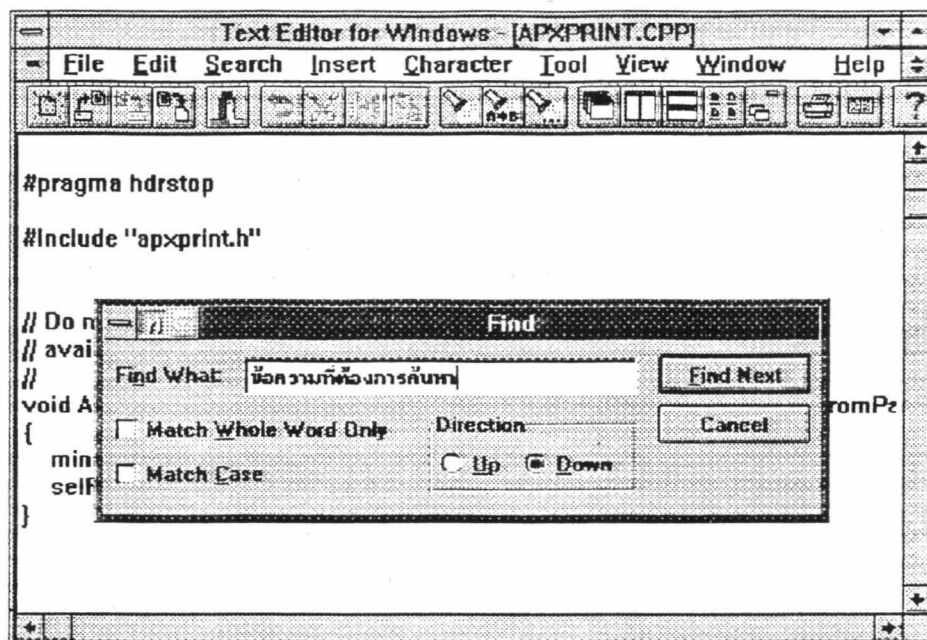
ซึ่งฟังก์ชัน CmEditDelete นี้จะรองรับเมสเสจ CM\_EDITDELETE จากเมสเสจคิวของ แอปพลิเคชัน

Search      มีหน้าที่จัดการเกี่ยวกับแฟ้มเอกสารดังนี้



รูปที่ 3.12 แสดงหน้าจอลำดับคำสั่งทั้งหมดของเมนู Search

- Find                      ค้นหาข้อความที่ระบุ  
 เมื่อเลือกเมนูย่อยนี้จะปรากฏหน้าจอดังรูปต่อไปนี้



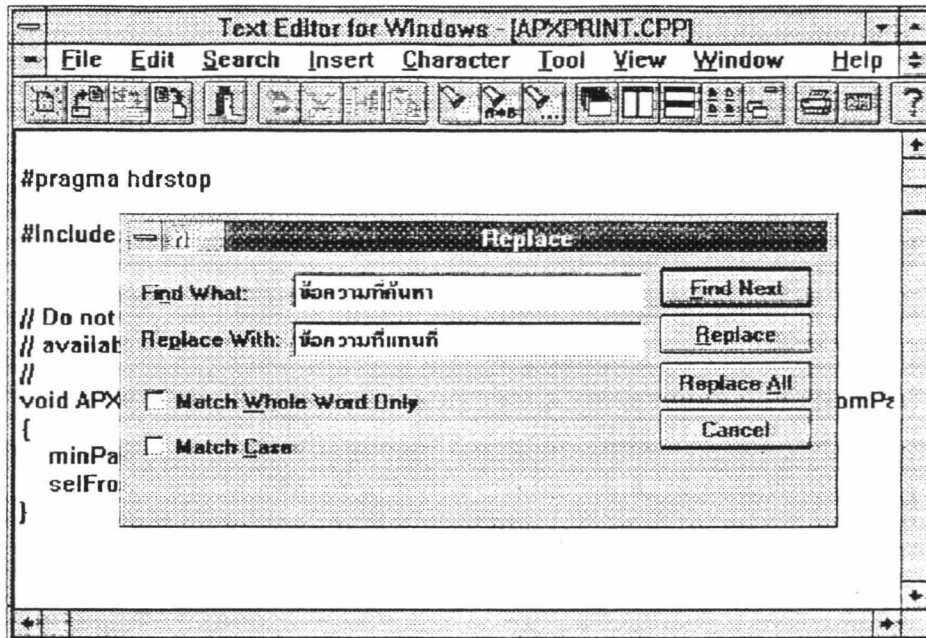
รูปที่ 3.13 ภาพแสดงผลที่ได้จากการเลือกคำสั่ง Find

ใช้เมื่อต้องการค้นหาข้อความที่ระบุในกรอบข้อความ Find โดยจะมีตัวชี้ (pointer) ชี้ไปที่ข้อความที่ระบุ และหยุดอยู่ที่ข้อความที่พบตัวแรก โดยเริ่มการค้นหาจากตำแหน่งของเคอร์เซอร์ ซึ่งฟังก์ชันนี้เป็น Protected Member Function ของ TEditWindow Class โดยมีโครงสร้างการสืบทอดจาก TEditWindow Class ดังนี้

```
void TEditWindow::CmEditFind ( );
```

ซึ่งฟังก์ชัน CmEditFind นี้จะรอรับเมสเสจ CM\_EDITFIND จากเมสเสจคิวของแอปพลิเคชัน

- Replace ค้นหาข้อความที่ระบุและเปลี่ยนเป็นข้อความที่ต้องการเมื่อเลือกเมนูย่อยนี้จะปรากฏหน้าจอดังรูปต่อไปนี้



รูปที่ 3.14 ภาพแสดงผลที่ได้จากการเลือกคำสั่ง Replace

ใช้เมื่อต้องการค้นหาข้อความที่ระบุ และเปลี่ยนเป็นข้อความอื่น โดยเมื่อเลือกคำสั่งนี้จะมีการรอบข้อความให้ใส่ข้อความที่ต้องการเปลี่ยน และข้อความที่จะนำไปแทนที่ โดยจะมีตัวชี้ ๆ ไปที่ข้อความที่ระบุ เมื่อพบจะนำข้อความใหม่ไปแทนที่และหยุดอยู่ที่ข้อความที่พบตัวแรก โดยเริ่มค้นหาและแทนที่จากตำแหน่งของเคอร์เซอร์ ซึ่งฟังก์ชันนี้เป็น Protected Member Function ของ TEditWindow Class โดยมีโครงสร้างการสืบทอดจาก TEditWindow Class ดังนี้

```
void TEditWindow::CmEditReplace ( );
```

ซึ่งฟังก์ชัน CmEditReplace นี้จะรองรับเมสเสจ CM\_EDITREPLACE จากเมสเสจคิวของ แอปพลิเคชัน

- Next ค้นหาข้อความที่ระบุต่อไปจนสิ้นสุดเอกสาร

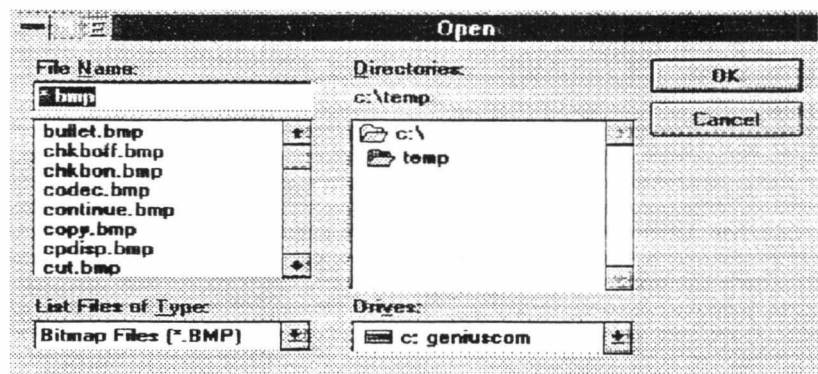
เมนูนี้ใช้เมื่อต้องการค้นหาข้อความที่ระบุไว้ ในกรอบข้อความของเมนู Find หรือ Replace ต่อไปเรื่อย ๆ จนสิ้นสุดเพิ่มเอกสาร โดยจะเปลี่ยนค่าตำแหน่งของตัวชี้ไปเรื่อย ๆ จนสิ้นสุดเอกสาร ตัวชี้จะมีค่าเป็น NULL และจะแสดงกรอบข้อความบอกให้ทราบว่าสิ้นสุดเอกสารแล้ว ซึ่งฟังก์ชันนี้เป็น Protected Member Function ของ TEditWindow Class โดยมีโครงสร้างการสืบทอดจาก TEditWindow Class ดังนี้

```
void TEditWindow::CmEditNext ( );
```

ซึ่งฟังก์ชัน CmEditNext นี้จะรอรับเมสเสจ CM\_EDITNEXT จากเมสเสจคิวของแอปพลิเคชัน

Insert มีหน้าที่จัดการเกี่ยวกับเพิ่มเอกสาร ดังนี้

- Insert Picture แทรกเพิ่มข้อมูลรูปภาพในเอกสารปัจจุบัน  
เมื่อเลือกคำสั่งนี้จะปรากฏหน้าต่างดังนี้



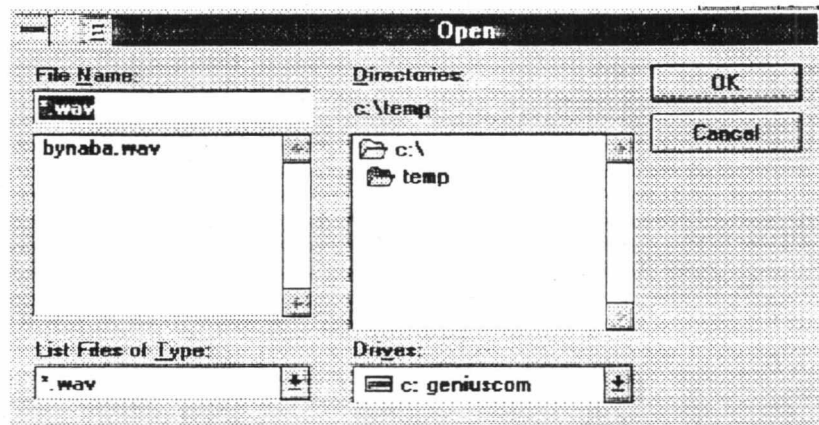
รูปที่ 3.15 แสดงผลที่ได้จากการเลือกคำสั่งของ Insert Picture

เมนูนี้ใช้เมื่อต้องการแทรกเพิ่มข้อมูลรูปภาพ .BMP เข้าไปประกอบในเอกสาร เมื่อเลือกเมนูย่อยนี้จะปรากฏหน้าจอดังรูป 3.15 ซึ่งฟังก์ชันนี้เป็น Protected Member Function ของ hyperMDIChild Class โดยมีโครงสร้างการสืบทอดจาก TMDIChild Class ดังนี้

```
void TMDIChild :: CmlInsertImage ( );
```

ซึ่งฟังก์ชัน CmlInsertImage นี้จะรอรับเมสเสจ CM\_INSERTIMAGE จากเมสเสจคิวของแอปพลิเคชัน

- Insert Sound เปิดเพิ่มข้อมูลเสียงและเล่นเปลี่ยนเมื่อกดปุ่ม OK  
เมนูนี้ใช้เมื่อต้องการเพิ่มเสียงประกอบเอกสาร โดยเพิ่มข้อมูลเสียงที่ใช้คือ .SPM และ .WAV โดยเมื่อเลือกเมนูย่อยนี้จะปรากฏหน้าจอดังรูปต่อไปนี้



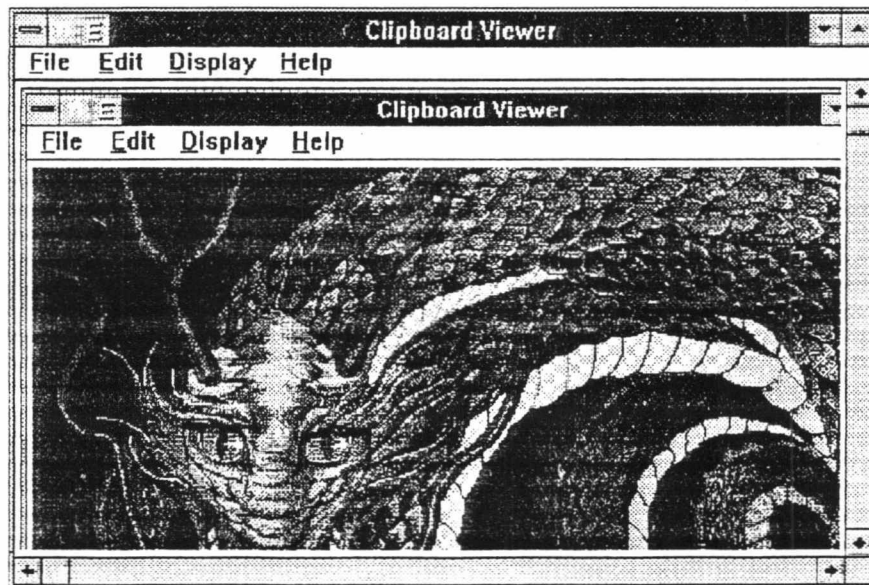
รูปที่ 3.16 แสดงผลที่ได้จากการเลือกคำสั่ง Insert Sound

ซึ่งฟังก์ชันนี้เป็น Protected Member Function ของ HyperApp Class โดยมีโครงสร้างการสืบทอดจาก HyperApp Class ดังนี้

```
void HyperApp :: CmlInsertSound ( );
```

ซึ่งฟังก์ชัน CmlInsertSound นี้จะรอรับเมสเสจ CM\_INSERTSOUND จากเมสเสจคิวของแอปพลิเคชัน

- Clipboard Viewer การแสดงข้อมูลปัจจุบันที่ถูกเก็บไว้ในคลิปบอร์ด เมื่อเลือกเมนูย่อยนี้จะปรากฏหน้าต่างดังรูปต่อไปนี้



รูปที่ 3.17 แสดงรายละเอียดของข้อมูลที่ถูกเก็บอยู่ในคลิปบอร์ด

( เป็นผลที่ได้จากการเลือกคำสั่ง ClipBoard Viewer )

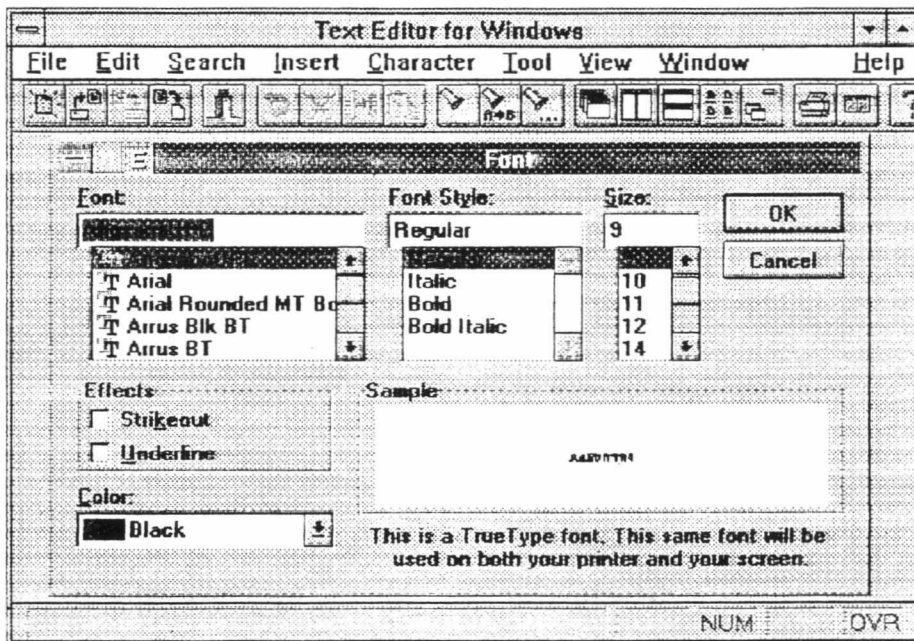
ซึ่งฟังก์ชันนี้เป็น Protected Member Function ของ HyperApp Class โดยมีโครงสร้างการสืบทอดจาก HyperApp Class ดังนี้

```
void HyperApp :: CmlInsertClip ( );
```

ฟังก์ชัน CmClipboardView นี้จะรอรับเมสเสจ CM\_INSERTCLIP จากเมสเสจคิวของ แอปพลิเคชัน

Character มีหน้าที่จัดการเกี่ยวกับรูปแบบตัวอักษรดังนี้

- Fonts เลือกรูปแบบตัวอักษร, ขนาดตัวอักษรและสีของตัวอักษร



รูปที่ 3.18 แสดงผลที่ได้จากการเลือกคำสั่ง fonts

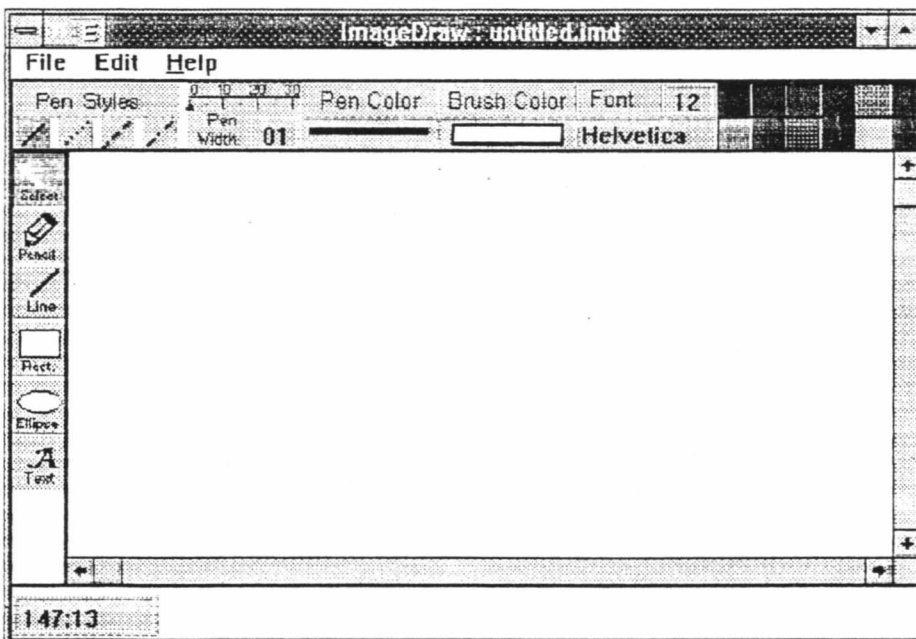
ใช้เมื่อต้องการเปลี่ยนรูปแบบตัวอักษร ขนาดของตัวอักษร และสีของตัวอักษร ซึ่งเมื่อเลือกเมนูนี้จะปรากฏกรอบข้อความให้เลือกรูปแบบต่าง ๆ ของตัวอักษรตามต้องการ ซึ่งฟังก์ชันนี้เป็น Protected Member Function ของ hyperMDIChild Class โดยมีโครงสร้างการสืบทอดจาก TMDIChild Class ดังนี้

```
void hyperMDIChild :: CmCharFont( );
```

ฟังก์ชัน CmCharFont นี้จะรองรับเมสเสจ CM\_CHARFONT จากเมสเสจคิว  
ของแอปพลิเคชัน

Tools มีหน้าที่จัดการเกี่ยวกับเพิ่มข้อมูลรูปภาพ ดังนี้

- ImageDraw Editor

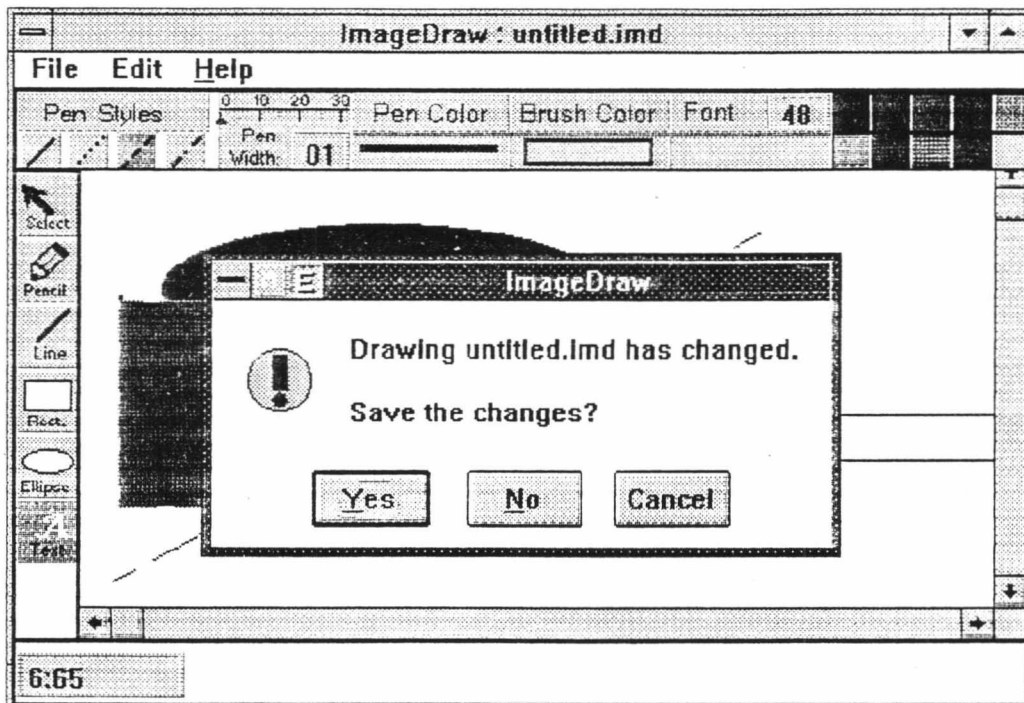


รูปที่ 3.19 แสดงหน้าจอของ ImageDraw Editor

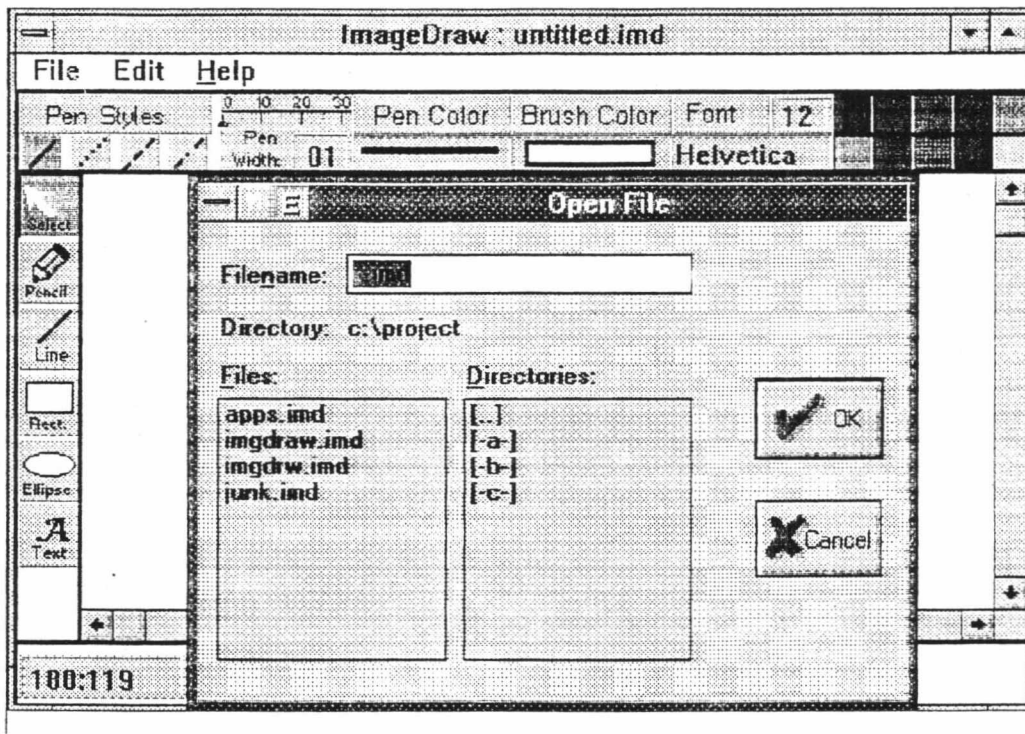
เป็นเครื่องมือที่ใช้ในการสร้างภาพและปรับปรุงแก้ไขภาพ ซึ่งภาพที่สร้างขึ้นนี้จะมีนามสกุลเป็น .imd หรือ .bmp ตามความต้องการ ซึ่งประกอบด้วยเมนู File, Edit และ Help ซึ่งฟังก์ชันนี้เป็น Protected Member Function ของ hyperApp Class โดยมีโครงสร้างการสืบทอดจาก TMDIChild Class ดังนี้

```
void hyperApp :: CmToolDraw ( );
```

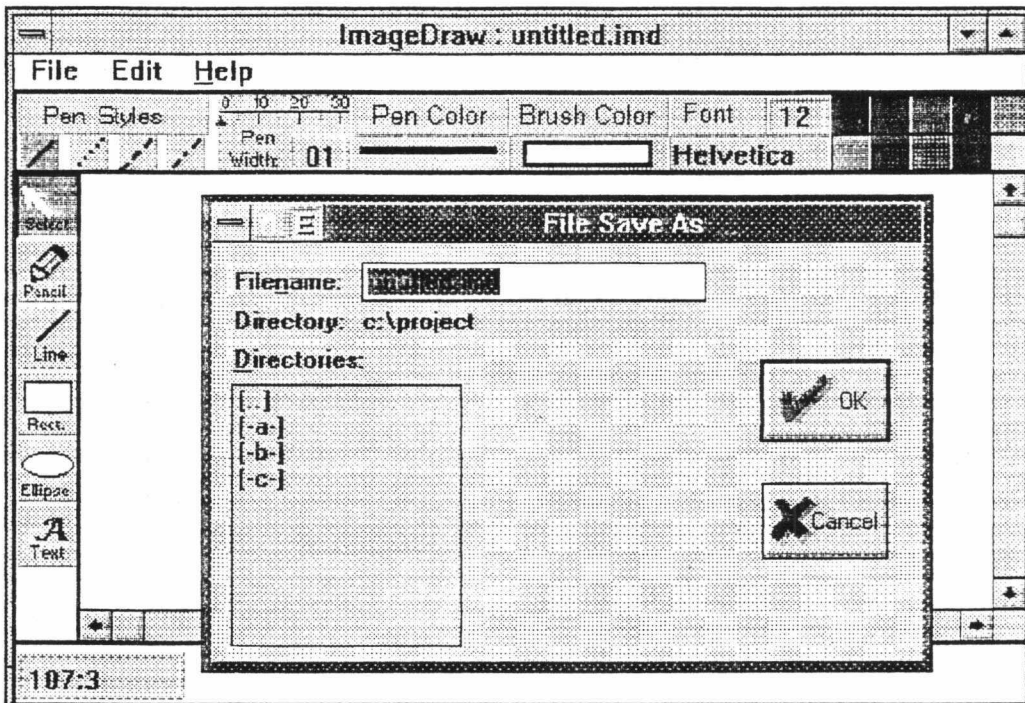
ซึ่งฟังก์ชัน CmToolDraw นี้จะรอรับเมสเสจ CM\_TOOLDRAW จากเมสเสจคิวของแอปพลิเคชัน .



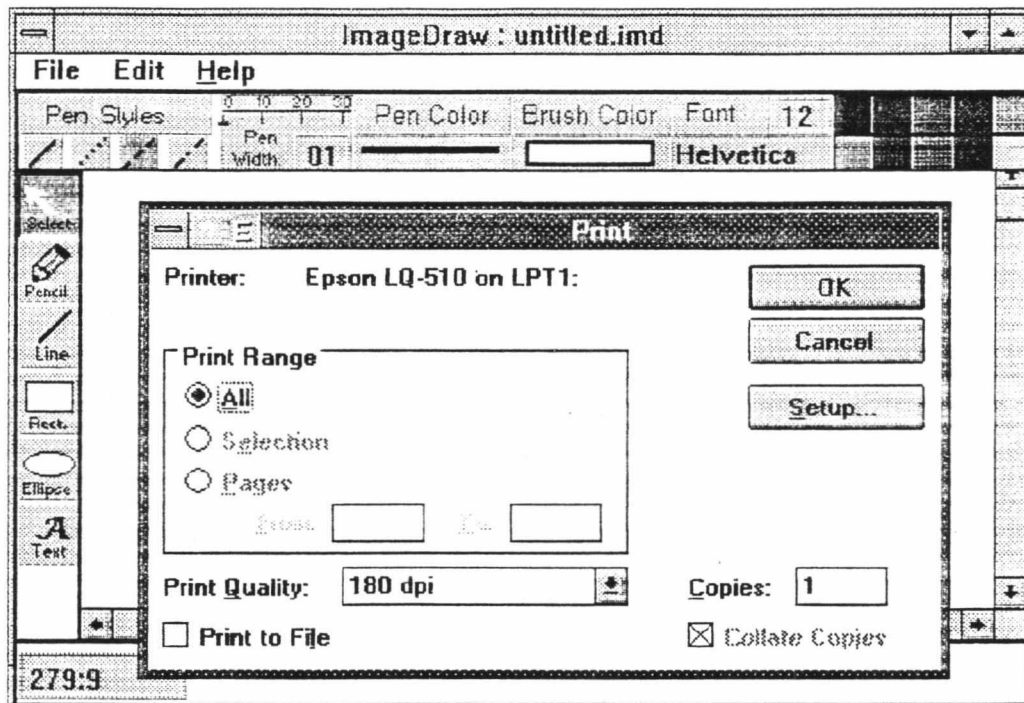
รูปที่ 3.20 ภาพแสดงตัวอย่างรูปที่ได้จากการวาดโดยใช้ ImageDraw Editor  
( โดยการเปิดเมนู File และเลือกคำสั่ง New )



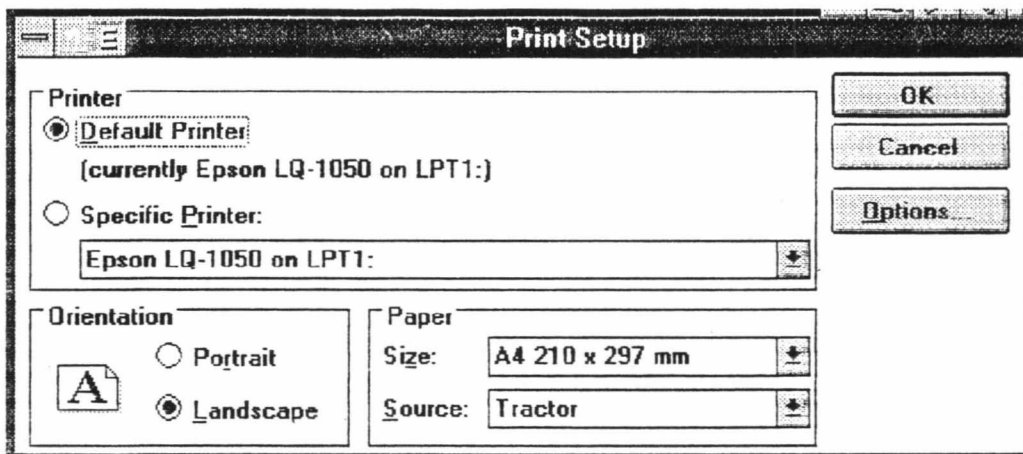
รูปที่ 3.21 ภาพแสดงผลที่ได้จากการเลือกคำสั่ง Open ในเมนู File



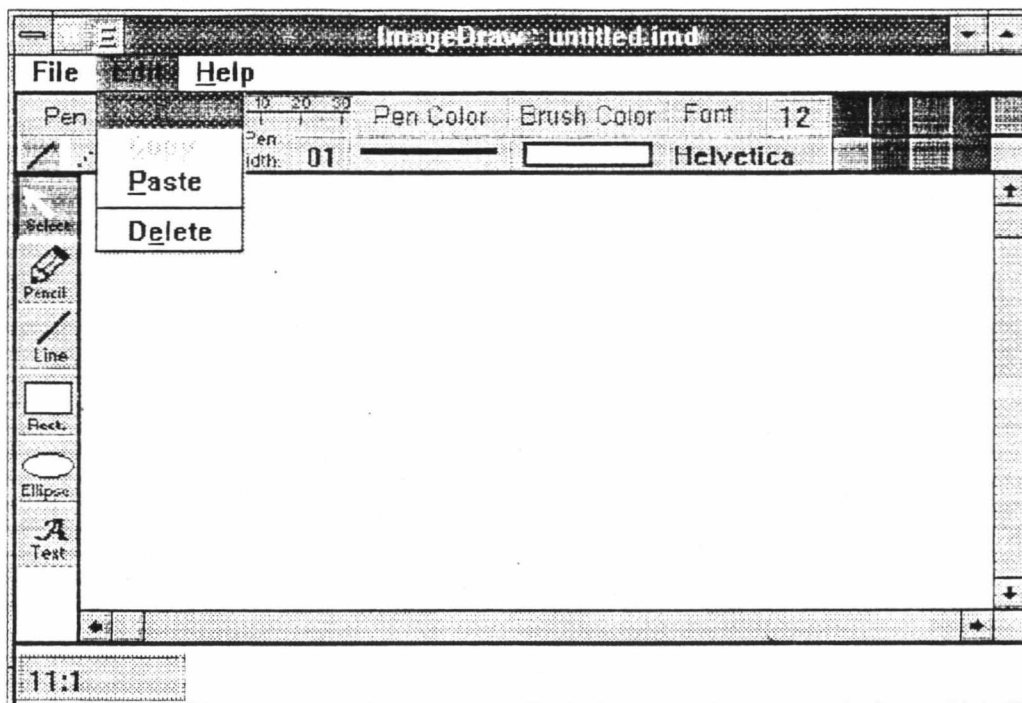
รูปที่ 3.22 ภาพแสดงผลที่ได้จากการเลือกคำสั่ง Save หรือ Save As ในเมนู File



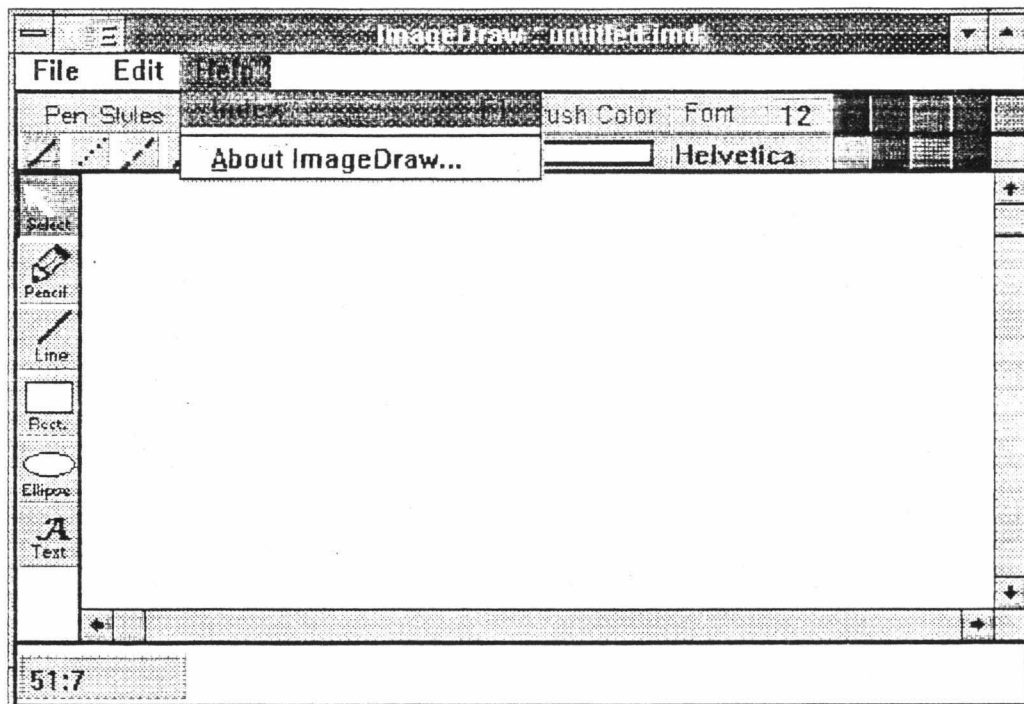
รูปที่ 3.23 ภาพแสดงผลที่ได้จากการเลือกคำสั่ง Print ในเมนู File



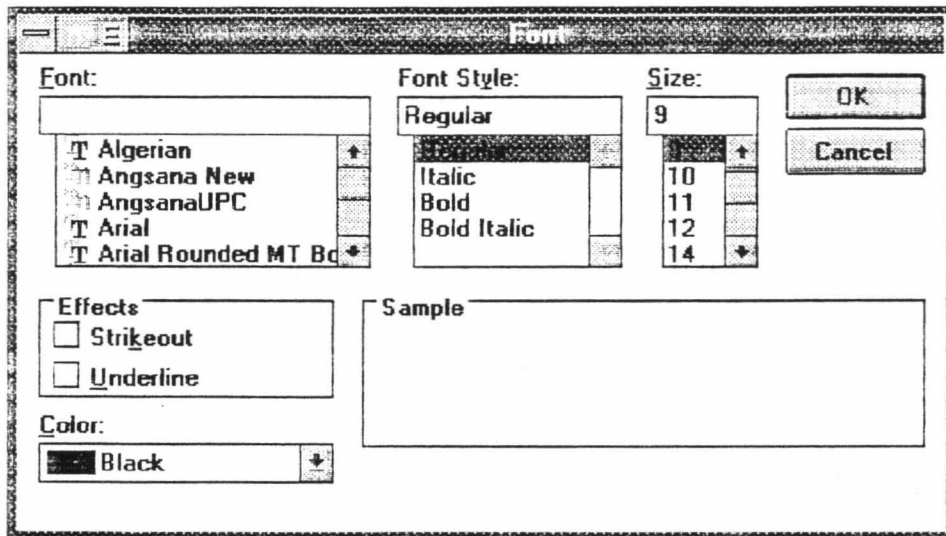
รูปที่ 3.24 ภาพแสดงผลที่ได้จากการเลือกคำสั่ง Print Setup ในเมนู File



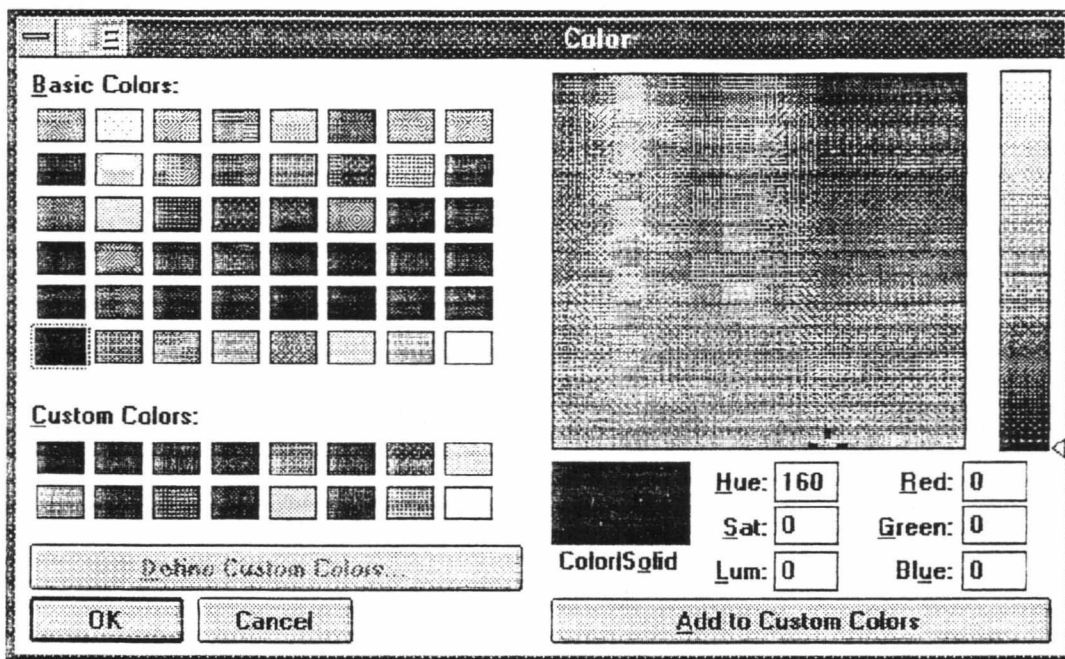
รูปที่ 3.25 ภาพแสดงคำสั่งทั้งหมดในเมนู Edit



รูปที่ 3.26 ภาพแสดงคำสั่งทั้งหมดในเมนู Help

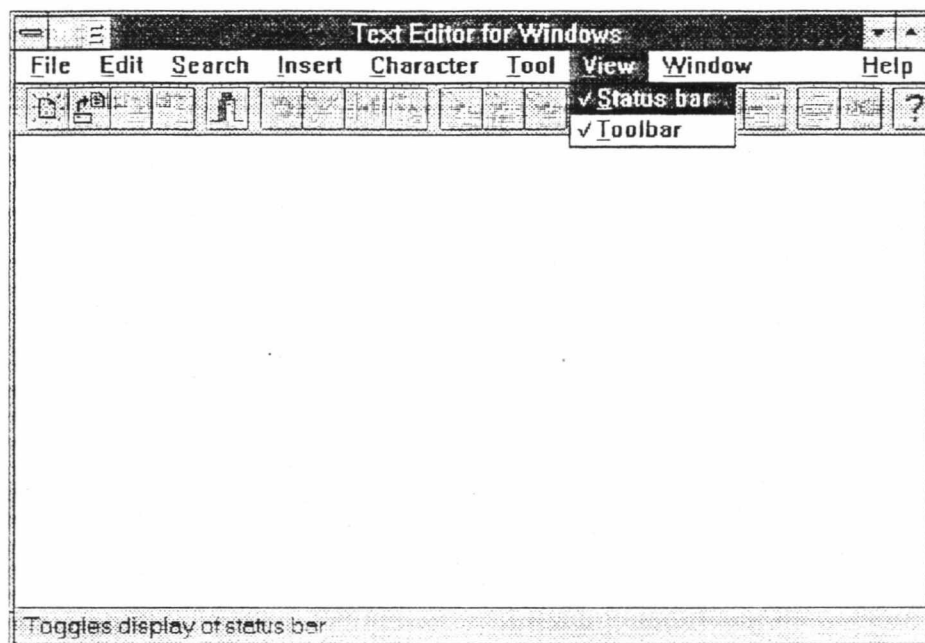


รูปที่ 3.27 ภาพแสดงผลที่ได้จากการเลือกคำสั่ง Font บน Tool Bar ของ  
ImageDraw Editor



รูปที่ 3.28 ภาพแสดงคำสั่งทั้งหมดในเมนู Edit

View มีหน้าที่จัดการเกี่ยวกับการแสดงสถานะของสเตตัสบาร์และทูลบาร์



รูปที่ 3.29 แสดงคำสั่งย่อยของเมนู View

- Status Bar           แสดงสถานะเปิดหรือปิดสเตตัสบาร์

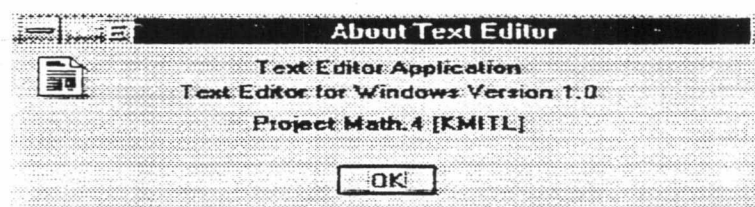
มีลักษณะการทำงานแบบเมนูตรวจสอบ     เพื่อใช้แสดงสถานะของสเตตัสบาร์     คือต้องการจะเปิด หรือปิดสเตตัสบาร์     ถ้าต้องการเปิดให้กด Enter หรือ     คลิกเมาส์ให้ปรากฏเครื่องหมายถูก ( ✓ ) หน้าเมนู     ถ้าต้องการปิดก็ให้กด Enter หรือคลิกเมาส์ซ้ำอีกครั้งหนึ่งเครื่องหมายถูกจะหายไป     ดังภาพ 3.30

- Tool Bar            แสดงสถานะเปิดหรือปิดทูลบาร์

มีลักษณะการทำงานแบบเมนูตรวจสอบ     เพื่อใช้แสดงสถานะของทูลบาร์     คือต้องการจะเปิด หรือปิดทูลบาร์     โดยมีการทำงานเหมือนคำสั่งสเตตัสบาร์     ดังภาพ 3.31

```
void hyperApp :: CmHelpAbout ( );
```

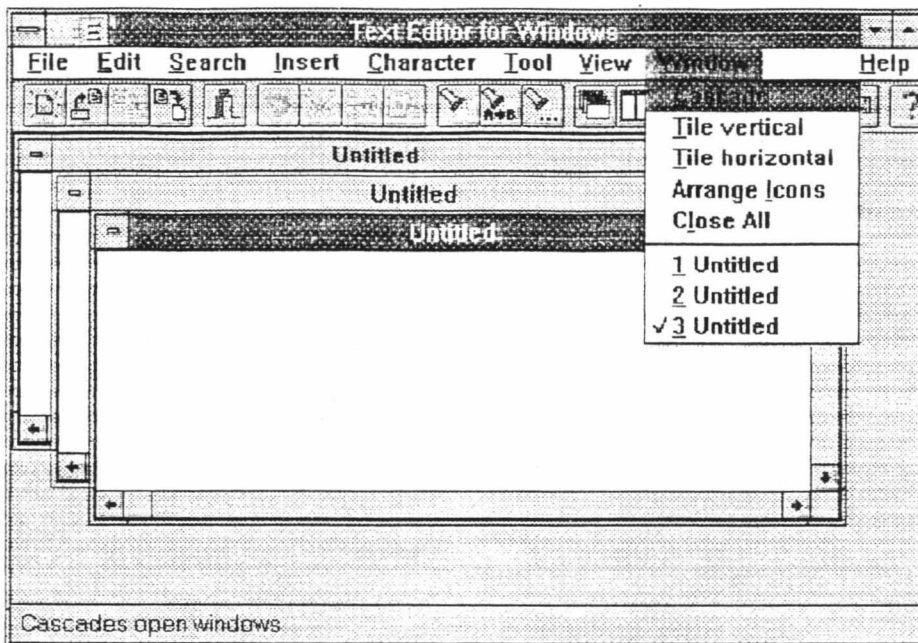
ฟังก์ชัน CmHelpAbout นี้จะรอรับเมลเสจ CM\_HELPABOUT จากเมลเสจคิวของ แอปพลิเคชัน



รูปที่ 3.40 ภาพแสดงผลที่ได้จากการเลือกคำสั่ง About ในเมนู Help

### 3.3 การใช้งานคลิปบอร์ด

การใส่ข้อมูลลงในคลิปบอร์ด ทำได้โดยจัดข้อมูลให้อยู่ในฟอร์มเมตหนึ่ง ( อาจเป็น ฟอร์มเมตมาตรฐานของ windows หรือฟอร์มเมตส่วนตัวก็ได้ ) ซึ่งเกือบทุกฟอร์มเมตจะจอง เนื้อที่หน่วยความจำ แบบไกลบอลมาส่วนหนึ่ง ด้วยฟังก์ชัน GlobalAlloc ก็อบบีข้อมูล ไปยังหน่วยความจำส่วนนั้น แล้วใช้ฟังก์ชัน SetClipboardData เพื่อส่งค่าแชนเดิลของหน่วย ความจำ ( ที่เก็บข้อมูล ) แก่คลิปบอร์ด ข้อมูลก็จะเข้าไปอยู่ในคลิปบอร์ด



รูปที่ 3.32 แสดงการจัดหน้าจอ เมื่อใช้คำสั่ง Cascade

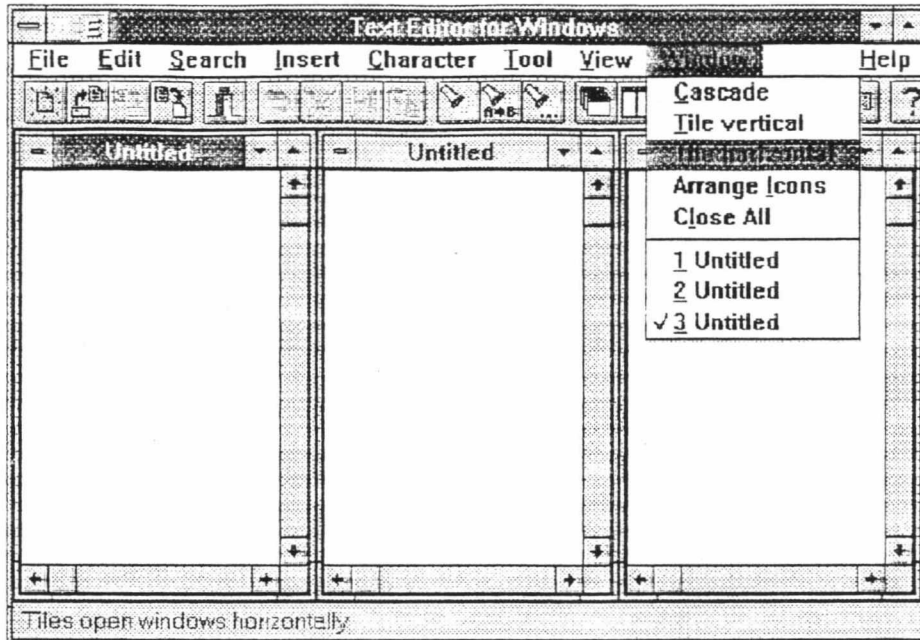
ใช้เมื่อต้องการแสดงแฟ้มเอกสารแบบเรียงต่อกัน และซ้อนกัน เพื่อสะดวกในการค้นหาแฟ้มเอกสารที่ต้องการ ซึ่งฟังก์ชันนี้เป็น Protected Member Function ของ TEditWindow class

โดยมีโครงสร้างการสืบทอดจาก TEditWindow Class ดังนี้

```
void TEditWindow :: CmCascade ( );
```

ฟังก์ชัน CmCascade นี้จะรองรับเมสเสจ CM\_CASCADE จากเมสเสจคิวของแอปพลิเคชัน

- Tile vertical      จัดแจงวินโดว์เป็นการเรียงโดยไม่ซ้อนกันในแนวตั้ง  
เมื่อเลือกเมนูนี้จะปรากฏหน้าจอดังรูปต่อไปนี้



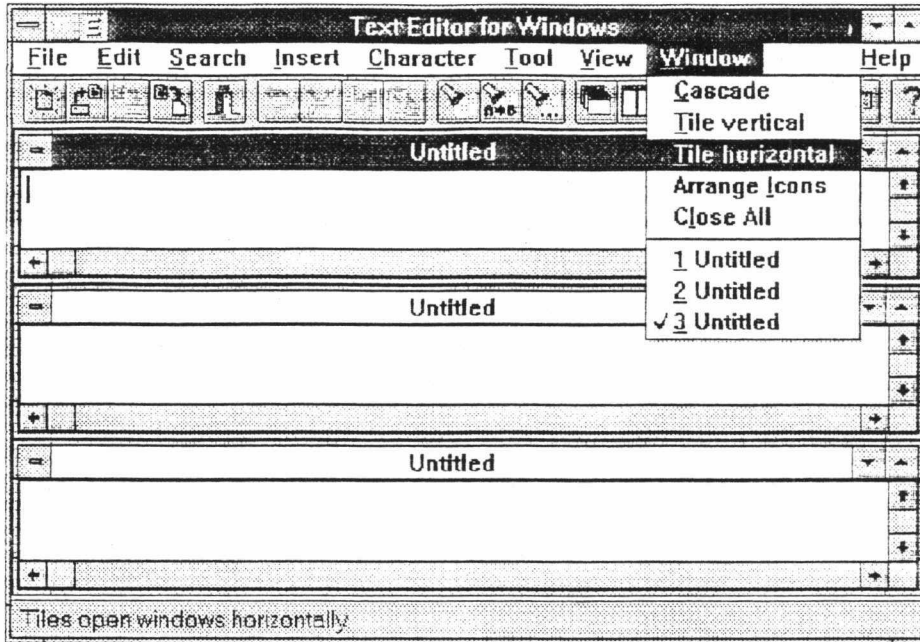
รูปที่ 3.33 แสดงหน้าจอเมื่อเลือกคำสั่ง Tile Vertical

ใช้เมื่อต้องการแสดงแฟ้มเอกสารแบบเรียงต่อ ๆ กัน และไม่ซ้อนกันตามแนวตั้ง เพื่อสะดวกในการค้นหาแฟ้มเอกสารที่ต้องการ และสะดวกในการจัดการเอกสารข้ามแฟ้มเอกสาร ซึ่งเป็น Protected Member Function ของ TEditWindow class โดยมีโครงสร้างการสืบทอดจาก TEditWindow Class ดังนี้

```
void TEditWindow :: CmTileVertical ( );
```

ฟังก์ชัน CmTileVertical นี้จะรองรับเมสเสจ CM\_TILEVERTICAL จากเมสเสจคิวของ แอปพลิเคชัน

- Tile Horizontal จัดแจงวินโดว์เป็นการเรียงโดยไม่ซ้อนกันในแนวนอน เมื่อเลือกเมนูนี้จะปรากฏหน้าจอดังรูปต่อไปนี้



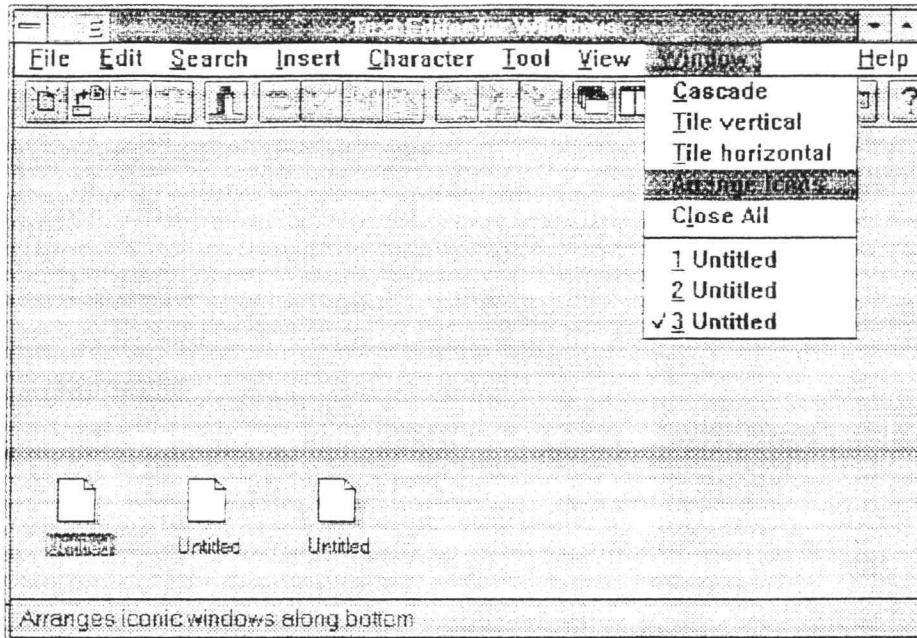
รูปที่ 3.34 แสดงหน้าจอเมื่อเลือกคำสั่ง Tile Horizontal

ใช้เมื่อต้องการแสดงแฟ้มเอกสารแบบเรียงต่อ ๆ กัน และไม่ซ้อนกันตามแนวนอน เพื่อสะดวกในการค้นหาแฟ้มเอกสารที่ต้องการ และสะดวกในการจัดการเอกสารข้ามแฟ้มเอกสาร ซึ่งเป็น Protected Member Function ของ TEditWindow class โดยมีโครงสร้างการสืบทอดจาก TEditWindow Class ดังนี้

```
void TEditWindow :: CmTileHorizontal ( );
```

ฟังก์ชัน CmTileHorizontal นี้จะรอรับเมสเสจ CM\_TILEHORIZONTAL จากเมสเสจคิวของ แอปพลิเคชัน

- Arrange Icon จัดเรียงรูปภาพที่ใช้แทนฟังก์ชันการทำงานของโปรแกรม ( icon ) ทางด้านล่างของวินโดว์



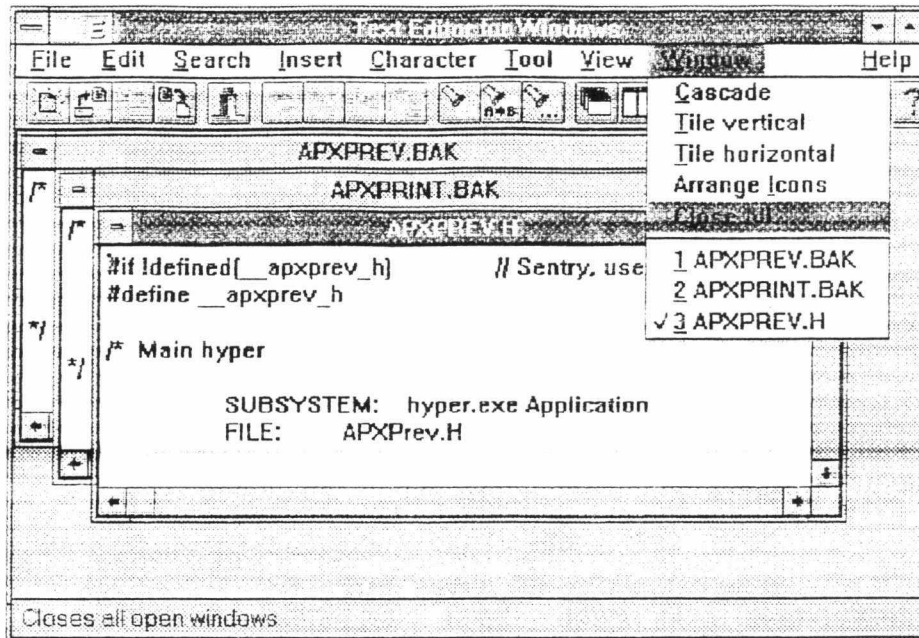
รูปที่ 3.35 แสดงหน้าจอเมื่อเลือกคำสั่ง Arrange Icon

ใช้เมื่อต้องการจัดเรียงไอคอนของเอกสารที่เปิดไว้ชั่วคราว ให้เรียงอย่างเป็นระเบียบ ตรง  
 ส่วนกลางของวินโดว ซึ่งเป็น Protected Member Function ของ TEditWindow class โดยมีโครง  
 สร้างการสืบทอดจาก TEditWindow Class ดังนี้

```
void TEditWindow ... CmArrangeIcon ( );
```

ฟังก์ชัน CmArrangeIcon นี้จะรองรับเมสเสจ CM\_ARRANGEICON จากเมสเสจคิวของแอป  
 พลิเคชัน

- Close All ปิดทุกวินโดวที่กำลังทำงานอยู่  
 เมื่อเลือกเมนูนี้จะปรากฏหน้าจอดังรูปต่อไป



รูปที่ 3.36 แสดงหน้าจอเมื่อเลือกคำสั่ง Close All

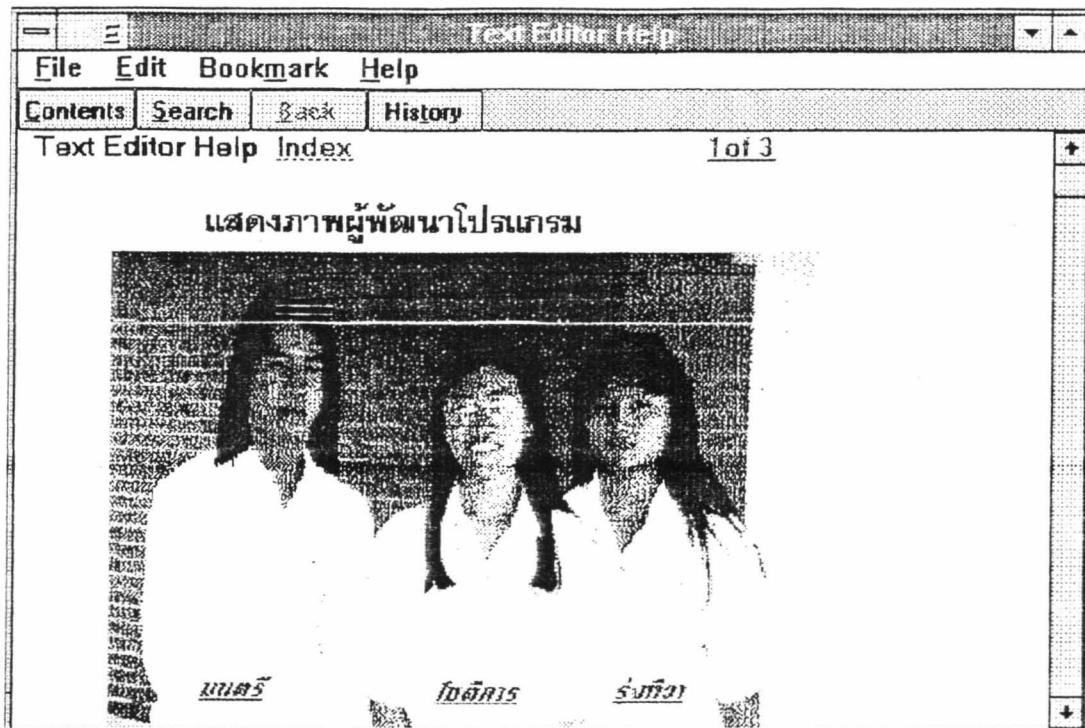
ใช้เมื่อต้องการปิดเอกสารทั้งหมดที่เปิดขึ้นมาแก้ไข ซึ่งเป็น Protected Member Function ของ TEditWindow class โดยมีโครงสร้างการสืบทอดจาก TEditWindow Class ดังนี้

```
void TEditWindow::CmCloseAll() {
```

ฟังก์ชัน CmCloseAll นี้จะรอรับเมสเสจ CM\_CLOSEALL จากเมสเสจคิวของแอปพลิเคชัน ผลที่ได้จากการเลือกคำสั่งนี้จากรูปที่ 3.27 เมื่อกด Enter หรือคลิกเมาส์ ผลที่ได้จะเหมือนกับรูปที่ 3.2

Help มีหน้าที่จัดการเกี่ยวกับระบบช่วยเหลือดังนี้

- Contents แสดงสารบัญของระบบช่วยเหลือ

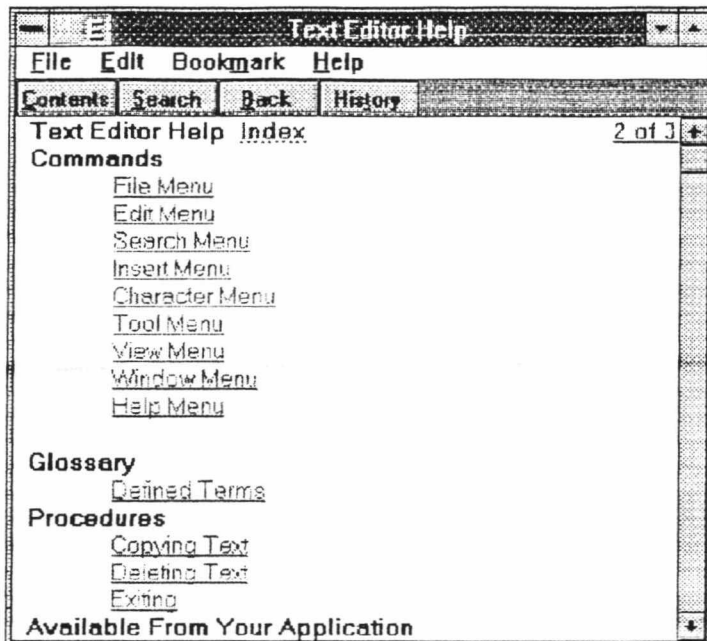


รูปที่ 3.37.1 แสดงหน้าจอเมื่อเลือกคำสั่ง Contents จากเมนู Help

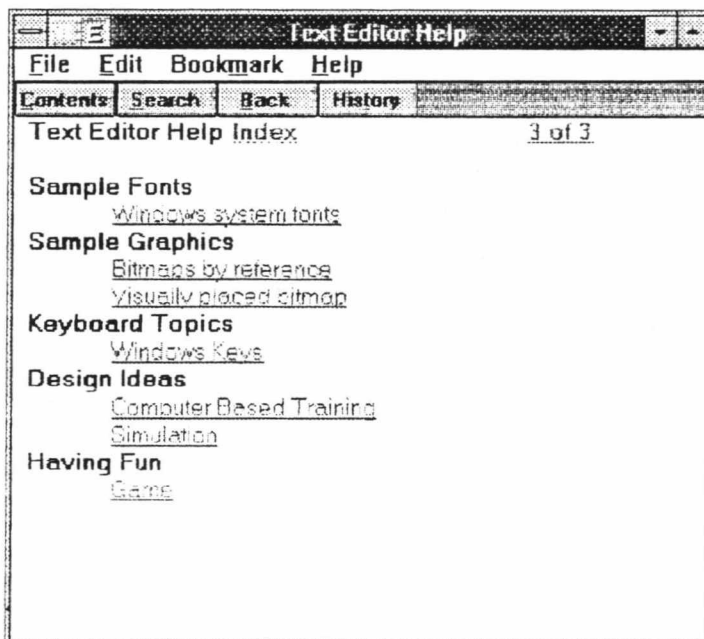
แสดงรายละเอียดของหัวข้อทั้งหมดที่มีอยู่ในระบบช่วยเหลือ โดยมีรูปแบบเป็นไฮเปอร์เท็กซ์ ช่วยเพิ่มความสะดวกสบายให้แก่ผู้ใช้ ซึ่งเป็น Protected Member Function ของ hyperApp class โดยมีโครงสร้างการสืบทอดจาก hyperApp Class ดังนี้

```
void hyperApp :: CmContent ( );
```

ฟังก์ชัน CmContent นี้จะรองรับเมสเสจ CM\_HELPCONTENT จากเมสเสจคิวของแอปพลิเคชัน

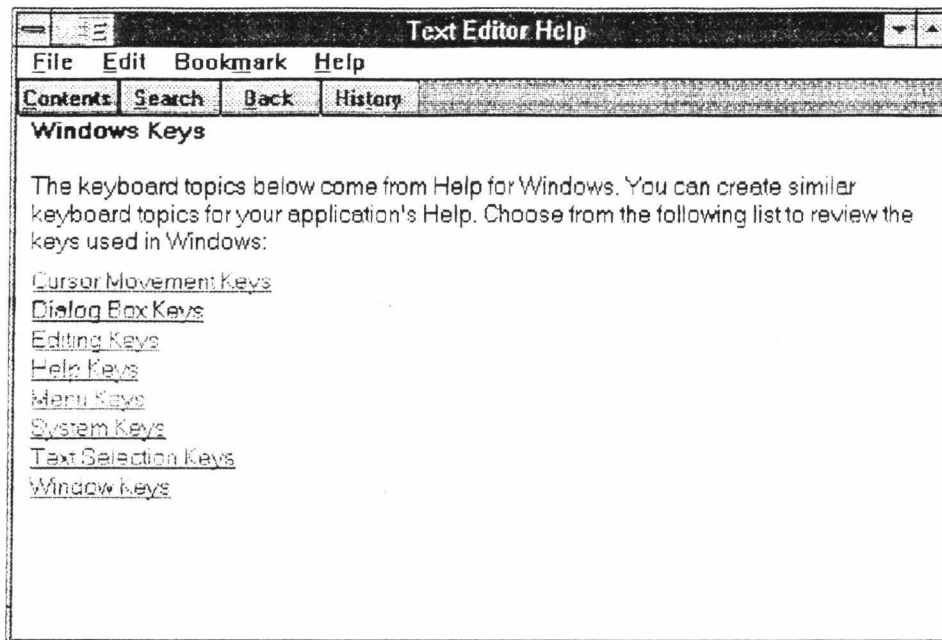


รูปที่ 3.37.2 รูปแสดงรายละเอียดต่อจากรูปที่ 3.37.1



รูปที่ 3.37.3 รูปแสดงรายละเอียดต่อจากรูปที่ 3.37.2

- Keyboard แสดงรายละเอียดของคีย์บอร์ดที่ใช้ในระบบ



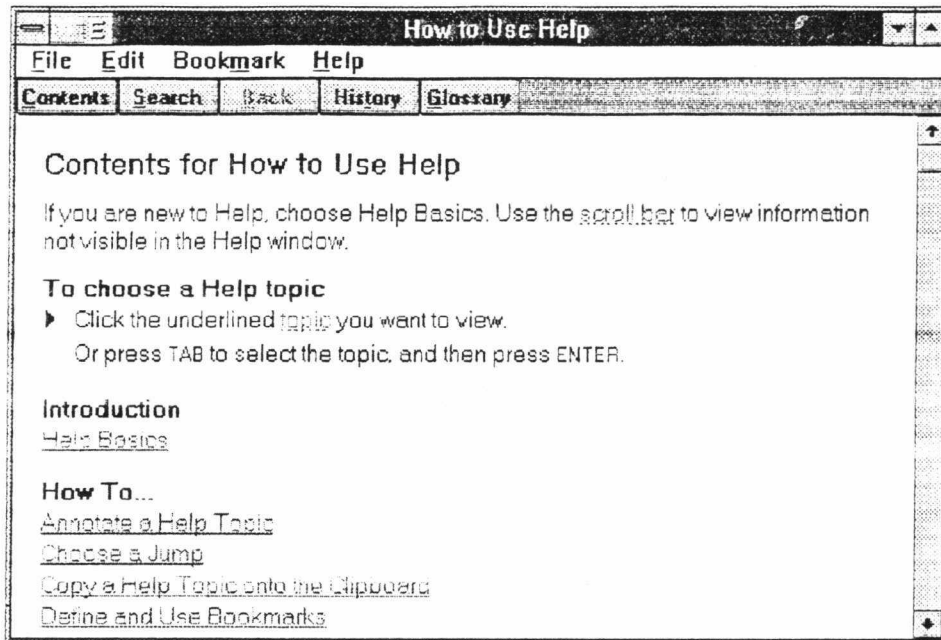
รูปที่ 3.38 แสดงหน้าจอเมื่อเลือกคำสั่ง Keyboard ในเมนู Help

แสดงรายละเอียดของคีย์ ที่ใช้ในระบบและมีรูปแบบเป็นไฮเปอร์เท็กซ์ ช่วยอำนวยความสะดวกให้ผู้ใช้งานขึ้น ซึ่งเป็น Protected Member Function ของ hyperApp class โดยมีโครงสร้างการสืบทอดจาก hyperApp Class ดังนี้

```
void hyperApp :: CmHelpkeyboard ( );
```

ฟังก์ชัน CmHelpkeyboard นี้จะรอรับเมสเสจ CM\_HELPKEYBOARD จากเมสเสจคิวของแอปพลิเคชัน

- Using Help แสดงรายละเอียดเกี่ยวกับการใช้ระบบช่วยเหลือ  
เมื่อเลือกเมนูย่อยนี้จะปรากฏหน้าต่างดังรูปต่อไปนี้



รูปที่ 3.39 แสดงหน้าจอเมื่อเลือกคำสั่ง Using Help ในเมนู Help

เมนูนี้จะบอกถึงรายละเอียดในการใช้ ระบบช่วยเหลือ มีรูปแบบเป็นไฮเปอร์เท็กซ์ ช่วยอำนวยความสะดวกให้ผู้ใช้มากขึ้น ซึ่งเป็น Protected Member Function ของ hyperApp class โดยมีโครงสร้างการสืบทอดจาก hyperApp Class ดังนี้

```
void hyperApp :: CmHelpHelp ( );
```

ฟังก์ชัน CmHelpHelp นี้จะรอรับเมสเสจ CM\_HELPHELP จากเมสเสจคิวของ แอปพลิเคชัน

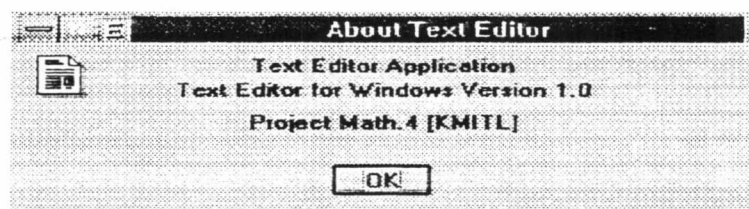
- About เกี่ยวกับโปรแกรมต้นแบบจัดการเอกสารบนวินโดวส์

( Text edit for windows )

เมนูนี้จะปรากฏเมสเสจบ็อกซ์ บอกถึงรายละเอียดของโปรแกรมได้แก่ ชื่อแอปพลิเคชัน , เวอร์ชัน เป็นต้น ซึ่งเป็น Protected Member Function ของ hyperApp class โดยมีโครงสร้างการสืบทอดจาก hyperApp Class ดังนี้

```
void hyperApp :: CmHelpAbout ( );
```

ฟังก์ชัน CmHelpAbout นี้จะรองรับเมสเสจ CM\_HELPABOUT จากเมสเสจคิวของ  
แอปพลิเคชัน



รูปที่ 3.40 ภาพแสดงผลที่ได้จากการเลือกคำสั่ง About ในเมนู Help

### 3.3 การใช้งานคลิปบอร์ด

การใส่ข้อมูลลงในคลิปบอร์ด ทำได้โดยจัดข้อมูลให้อยู่ในฟอร์แมตหนึ่ง ( อาจเป็นฟอร์แมตมาตรฐานของ windows หรือฟอร์แมตส่วนตัวก็ได้ ) ซึ่งเกือบทุกฟอร์แมตจะของเนื้อที่หน่วยความจำ แบบไกลบอลมาส่วนหนึ่ง ด้วยฟังก์ชัน GlobalAlloc ก็อบบีข้อมูลไปยังหน่วยความจำส่วนนั้น แล้วใช้ฟังก์ชัน SetClipboardData เพื่อส่งค่าแชนเดิลของหน่วยความจำ ( ที่เก็บข้อมูล ) แก่คลิปบอร์ด ข้อมูลก็จะเข้าไปอยู่ในคลิปบอร์ด

โดยส่วนมากแล้ว แอปพลิเคชันมาตรฐานที่มีมากับ Windows จะมีความสามารถนี้ให้ผู้ใช้สั่งงานได้ในเมนู Edit ( สั่งด้วยคำสั่ง Cut หรือ Copy ) และ แอปพลิเคชันที่ได้พัฒนาขึ้นมานี้ก็ไต่ยัดแนวเดียวกันนี้ด้วย เพื่อให้ผู้ใช้คุ้นเคยและสะดวกในการใช้งาน

ฟอร์แมตข้อมูลมาตรฐานของ Windows ที่สามารถใช้clipboardส่งผ่านกันได้ มีดังนี้

ฟอร์แมต	หมายถึง
CF_TEXT	เป็นข้อความที่ปิดด้วย Null
CF_OEMTEXT	เป็นข้อความปิดด้วย Null ที่เป็นชุดอักษร OEM
CF_METAFILEPICT	รูปภาพที่เป็น Metafile
CF_BITMAP	รูปบิตแมพที่ขึ้นกับอุปกรณ์
CF_DIB	รูปบิตแมพที่ไม่ขึ้นกับอุปกรณ์
CF_SYLK	มาตรฐานการเก็บข้อมูลแบบ SYLK
CF_DIF	มาตรฐานการเก็บข้อมูลแบบ DIF
CF_TIFF	มาตรฐานการเก็บข้อมูลแบบ TIFF

### 3.3.1 การก๊อปปี้ข้อมูลลงclipboard

เมื่อผู้ใช้เลือกคำสั่ง Copy จากเมนู Edit แอปพลิเคชันก็จะก๊อปปี้ข้อมูลลงสู่ clipboard ขั้นตอนการก๊อปปี้ข้อมูลลงclipboard ทำได้ดังนี้

1. ก๊อปปี้ข้อความที่ต้องการลงในหน่วยความจำ global ที่จองไว้ก่อนแล้ว
2. เปิดclipboard
3. เคลียร์clipboard
4. ส่งแผ่นแอนเดิลของหน่วยความจำ global นั้นแก่clipboard
5. ปิดclipboard

การจองหน่วยความจำ จะต้องคอยตรวจสอบค่าที่ได้รับกลับมาเสมอ เพราะในกรณีที่ windows ไม่สามารถหาหน่วยความจำมาให้ได้ ก็จะส่งค่ากลับมาเป็น NULL

ในแต่ละครั้งที่มีการก๊อปปี้ข้อมูลลงclipboard ก็จะมีการจองหน่วยความจำขึ้นใหม่ในทุก ๆ ครั้ง ( โปรดดูจากโปรแกรม ) เพราะเมื่อส่งแอนเดิลของข้อมูลให้แก่clipboard ก็จะ

ถือว่าคลิปบอร์ดเป็นเจ้าของ นั้นหมายถึงไม่สามารถใช้แฮนเดิลนั้นได้อีก และยังห้าม free ทิ้งอีกด้วย

### 3.3.2 การปะข้อความจากคลิปบอร์ด

เมื่อจะปะข้อความขึ้นหนึ่งจากคลิปบอร์ด นั่นคือ รับแฮนเดิลของข้อมูลที่มีอยู่ในคลิปบอร์ดนั่นเอง โดยมีขั้นตอนดังนี้ คือ

1. เปิดคลิปบอร์ด
2. รับค่าแฮนเดิลของข้อความกลับมา ( โดยใช้ CF\_TEXT หรือ CF\_OEMTEXT )
3. ปิดคลิปบอร์ดและนำค่าแฮนเดิลไปใช้

แต่ก่อนที่จะปะข้อความกลับมา ก็ต้องตรวจสอบเสียก่อนว่า ในคลิปบอร์ดนั้นมีข้อมูลที่เป็นข้อความตัวอักษรอยู่จริง โดยจะตรวจสอบเมื่อได้รับเมนูเสจ ( CM\_INITMENU ( ซึ่งวินโดว์จะได้รับก่อนที่จะแสดงเมนู ) ) คือถ้าในคลิปบอร์ดไม่มีข้อความตัวอักษรอยู่ก็สั่งบัง ( grayed ) คำสั่ง Paste ในเมนูเสียก่อนที่ผู้ใช้จะได้เลือก

สามารถตรวจสอบหรือก๊อปปี้ข้อมูลที่ได้จากคลิปบอร์ด แต่ จะต้องไม่แก้ไขข้อมูลที่ได้แฮนเดิลมา เมื่อต้องการใช้ข้อมูลดังกล่าว จะล็อกด้วยแฮนเดิลที่ได้มา และจะต้องปลดล็อกทันทีที่ใช้เสร็จ สามารถใช้แฮนเดิลข้อมูลที่ได้จากคลิปบอร์ด เพียงชั่วคราวเท่านั้น เพราะแฮนเดิลเป็นของคลิปบอร์ด มิใช่เป็นของแอปพลิเคชันที่ขอรับข้อมูล การจะใช้ข้อมูลจากคลิปบอร์ดที่เหมาะสมที่สุดคือ ก๊อปปี้ข้อมูลจากคลิปบอร์ด แล้วปิดทันทีโดยไม่เปลี่ยนแปลงข้อมูลที่อยู่ในคลิปบอร์ดอย่างเด็ดขาด และก่อนที่จะปิดคลิปบอร์ดนั้นด้วยฟังก์ชัน CloseClipboard จะต้องแน่ใจว่า ปลดล็อกแฮนเดิลของข้อมูลแล้วด้วย

## บทที่ 4

### การประเมินผล

ผลที่ได้จากการพัฒนาแบบโปรแกรมจัดการเอกสารบนวินโดว์ประเมินในแต่ละด้านได้ดังนี้

#### 4.1 ความสะดวกในการจัดการเอกสาร

การจัดการเอกสารของโปรแกรม Thai Write นั้นไม่สามารถจัดการกับเอกสารได้หลาย ๆ หน้าพร้อม ๆ กัน แต่จะจัดการได้เพียงครั้งละหน้าเดียว เราสามารถแก้ไขได้ด้วยการใช้วิธีการ มัลติอินสแตนซ์ คือ วินโดว์ยอมให้รันแอปพลิเคชันตัวเดียวกันได้หลายครั้ง ถ้าเราต้องการจัดการเอกสารตัวที่สองตัวที่สาม... ก็จำเป็นต้องรันโปรแกรม Thai Write ตามจำนวนที่ต้องการ เมื่อต้องการเลือกจัดการกับเอกสารตัวใดตัวหนึ่ง เพียงคลิกเมาส์ลงบนวินโดว์นั้น ๆ หรืออาจกด Alt+Tab เพื่อเปลี่ยนจากตัวหนึ่งไปยังอีกตัวหนึ่งจากข้อบกพร่องนี้ จึงมีการคิดพัฒนาโปรแกรมนี้ขึ้น โดยการใช้เทคนิคการเชื่อมต่อแบบหลายเอกสาร ( MDI ) ซึ่งเป็นเทคนิคการติดต่อกับผู้ใช้ เพื่อจัดการกับเอกสารหลาย ๆ หน้า หลาย ๆ วินโดว์ได้พร้อมกัน ซึ่งปัจจุบันได้รับความนิยมอย่างมาก เมื่อเปรียบเทียบระหว่างการจัดการเอกสารแบบ มัลติอินสแตนซ์ กับแบบเอ็มดีไอ จะเห็นว่าเอ็มดีไอ มีประสิทธิภาพดีกว่าอย่างแน่นอน เนื่องจาก แอปพลิเคชันเอ็มดีไอนี้เอกสารแต่ละหน้าแม้จะอยู่ภายใต้สิ่งแวดล้อมเดียวกัน แต่ก็ไม่ก่อให้เกิดความสับสน เมื่อเปลี่ยนไปเปลี่ยนมา การจัดการตำแหน่งและขนาดของเอกสาร หรือ วินโดว์แต่ละตัวง่ายกว่า รวมทั้งการยุบตัวไปเป็นไอคอน การเรียกใช้งานเอกสารที่ถูกต้องแม่นยำและง่ายดายนกว่า ทั้งยังสามารถทำการเปรียบเทียบเพิ่มข้อมูลได้มากกว่าสองหน้าจอดีขึ้นไป

#### 4.2 ความสะดวกในการใช้งานของผู้ใช้

สำหรับการใช้งานโปรแกรมนั้น นอกจากความสามารถในการจัดการกับเอกสารหลาย ๆ หน้าพร้อม ๆ กันได้แล้ว ยังมีส่วนประกอบสำคัญ ที่เป็นที่ยอมรับอีกมากด้วยกัน เช่น ทูลบาร์ และ สเตตัสบาร์ สำหรับโปรแกรม Thai Write เมื่อเราพิจารณาจะพบว่า โปรแกรมนี้ไม่มีทูลบาร์ จึงทำให้การใช้งานแอปพลิเคชัน ไม่สะดวกเท่าที่ควร และไม่มี สเตตัสบาร์ เพื่อช่วยอธิบายการทำงาน ในขณะที่ทำการเลือก

เมนู ซึ่งสแตตัสบาร์นี้ จะอยู่ส่วนล่างของวินโดว์ ปัจจุบันเราจะพบเห็นการใช้ทุลบาร์กันอย่างแพร่หลาย โดยหน้าที่หลักมักจะเป็นทางลัดในการสั่งโปรแกรมให้ทำงาน เช่นเดียวกับวิธีการกดคีย์ค้อน

สำหรับการสั่งโปรแกรมให้ทำงานใน Thai Write เราสามารถสั่งงานได้ทั้งจากเมนูหรือ คีย์ค้อน ซึ่งเป็นมาตรฐานที่ติดมากับวินโดว์เมื่อพิจารณาการใช้งานของทั้ง 2 แบบจะพบว่าไม่สะดวกต่อการใช้งานมากนัก สำหรับเมนูถูกออกแบบมาเพื่อให้ผู้ใช้สามารถเรียนรู้คำสั่งต่าง ๆ ได้ง่าย เหมาะกับผู้ที่เพิ่งหัดใช้หรือยังไม่มี ความชำนาญ การใช้งานจะต้องกดคีย์อยู่หลายครั้ง เช่น Alt+F-o เพื่อเปิดไฟล์ Alt+F-s เพื่อบันทึกข้อมูล เป็นต้น หรือการเลือกเมนูด้วยเมาส์ ก็ต้องลากหลายครั้ง จะเห็นว่าในช่วงแรกของการใช้งานจะยังมีความสนใจต้องการที่จะใช้ แต่พอนาน ๆ ไปจะรู้สึกว่ามันน่าเบื่ออยากหาวิธีการที่จะสั่งงานได้ครั้งเดียวแล้วโปรแกรมทำงาน ซึ่งอาจทำได้โดยใช้วิธีการกดคีย์ค้อนที่ถูกออกแบบมาสำหรับผู้ใช้ที่ต้องการความรวดเร็ว แต่ในความเป็นจริง การทำงานบางอย่างอาจต้องใช้โปรแกรมมากกว่าหนึ่งตัวขึ้นไป รูปแบบการกดคีย์ค้อนก็แตกต่างกันออกไป เมื่อมีการใช้แอปพลิเคชันหลายตัวก็ต้องมีการจำรูปแบบการตรวจคำสั่งหลาย ๆ แบบ ซึ่งต้องอาศัยความคุ้นเคยอย่างมากจึงจะสามารถจำได้ และบางครั้งอาจเกิดความสับสนก่อให้เกิดความผิดพลาดได้ ดังนั้นจะเห็นว่าการใช้งานของทั้งเมนูและคีย์ค้อนมีช่องว่างอยู่มาก เราจึงมีความคิดที่จะศึกษาถึงเครื่องมือที่จะสามารถอำนวยความสะดวกให้มากขึ้น ซึ่งนักออกแบบยูเซอร์อินเทอร์เฟซได้เสนอขึ้นมา

เป็นวิธีการสั่งงานแบบรูปภาพ ทำให้ไม่ต้องจำ ไม่ต้องลากเมาส์หลายครั้ง เพราะรูปภาพสามารถสื่อความหมายได้อย่างดี เพียงคลิกเมาส์ที่เดียวสามารถใช้งานได้เลย เครื่องมือชนิดนี้มีชื่อเรียกแตกต่างกันไป ตามแต่บริษัทผู้พัฒนาโปรแกรม เช่น ไมโครซอฟท์ รวบรวมรูปภาพไว้บนบาร์ที่อยู่ใต้เมนูบาร์ลงมาเรียกว่า ทุลบาร์ หรือบางบริษัทเรียกว่า ริบบอน (Ribbon) หรือ บางครั้งมีการรวบรวมไว้ในกรอบสี่เหลี่ยมย้ายไปที่ไหนก็ได้เรียกว่า ทุลบ็อกซ์ (Tool Box) เป็นต้น แต่สำหรับการศึกษาค้นคว้า เพื่อพัฒนาโปรแกรมต้นแบบ ในครั้งนี้เราจะศึกษาในรูปแบบของ ทุลบาร์

นอกจากนี้ โปรแกรม Thai Write ไม่มีเครื่องมือที่ช่วยอธิบายการทำงาน ในขณะที่มีการเลือกเมนู ซึ่งจะปรากฏอยู่เบื้องล่างของวินโดว์ ที่เรียกว่า สแตตัสบาร์ ทำให้บางครั้งเราไม่สามารถทราบว่าจะทำอะไร เครื่องกวดังทำงานอยู่ที่ส่วนใด หรือกำลังใช้คำสั่งใดอยู่ และคำสั่งนั้นมีการทำงานอย่างไร

### 4.3 ความสามารถในการคัดลอกแฟ้มข้อมูล

ความสามารถในการคัดลอกข้อความข้ามไฟล์ไม่สะดวก และใช้งานได้ไม่ดีเท่าที่ควร เนื่องจากโปรแกรม Thai Write ไม่สามารถจัดการเอกสารทีละใบได้ หลาย ๆ หน้า พร้อม ๆ กัน แบบวินโดว์หน้าจอละเอียด (single window) แต่แสดงผลได้เพียงแค่นำหน้าเดียว ดังนั้น ในการคัดลอกข้อความข้ามไฟล์ จะทำได้ก็ต่อเมื่อ มีการเก็บข้อความ ที่ต้องการคัดลอก ลงบนคลิปปอร์ดก่อน แล้วจึงปิดหน้าต่างการทำงานนั้น และเปิดหน้าต่างที่เราต้องการจะคัดลอกไปไว้ขึ้นมาใหม่ทับบนหน้าต่างนั้น ซึ่งก่อนทำการแปะข้อความลงไป เมื่อคัดลอกเสร็จแล้ว ยังคงต้องการใช้งานไฟล์เก่าอยู่ก็ต้องทำการเรียกขึ้นมาใหม่ อีก โดยทับหน้าต่างที่เพิ่งทำการคัดลอกเสร็จไป จะเห็นว่าการทำงานที่สามารถเปิดได้เพียงหน้าต่างเดียว ต้องมีการเรียกไฟล์ขึ้นมาใหม่ทุกครั้งที่ต้องการแต่เพียงแคคลิกเมาส์ เลื่อนไปยังหน้าต่างที่เราต้องการ ซึ่งมีการเปิดไฟล์ข้อความนั้นไว้แล้วก็สามารถทำการคัดลอกได้หรือวางข้อความที่ต้องการลงไปได้โดยสะดวก และสามารถย้อนกลับมาดูข้อความเก่าได้โดยไม่ต้องปิดหน้าต่างข้อความอื่น ๆ

### 4.4 การจัดหน้าเอกสารก่อนพิมพ์

แอปพลิเคชัน Thai Write ไม่มีส่วนของ Print Preview ในกรณีที่เรากำลังจะพิมพ์งานออกทางเครื่องพิมพ์ บางครั้งต้องมีการจัดหน้าเอกสารให้มีความเหมาะสม สำหรับ การเว้นช่องหน้ากระดาษจากขอบกระดาษ และเราสามารถตรวจสอบความสวยงามนี้ได้จากแอปพลิเคชันในส่วนของ Print Preview ซึ่งจะแสดงหน้ากระดาษของเอกสารให้เราเห็นทั้งหน้าทีละหน้า ในแต่ละหน้ามีการจัดช่องหน้ากระดาษเหมาะสมเพียงใดและเราสามารถตกแต่งเอกสารได้จากการใช้ส่วนของ Print Preview นี้ด้วย ซึ่งเป็นการอำนวยความสะดวกให้ ทำให้ไม่ต้องเสียเวลาพิมพ์ใหม่บ่อย ๆ เพราะเอกสารที่ได้ออกมา มีการจัดหน้าไม่สวยงาม สำหรับอุปกรณ์ที่ใช้ในการพิมพ์สามารถใช้กับเครื่องพิมพ์ได้หลายรุ่น เนื่องจากมีการกำหนดให้มีการติดตั้งค่าของประเภทเครื่องพิมพ์ได้

### 4.5 ประเภทของเอกสารที่เรียกขึ้นมาใช้งาน

สามารถเรียกเอกสารที่มีนามสกุลเป็น TXT ที่มีรูปแบบอักขระเป็นแบบ Systemfont เอกสารที่เรียกนั้นจะต้องมีรูปแบบการเก็บแฟ้มข้อมูลเป็นแบบเท็กซ์โหมดเท่านั้น ดังนั้นจึงสามารถเรียกเอกสารที่เป็นรหัสโปรแกรมเดิม (Source Code) ขึ้นมาแก้ไขได้ทุกนามสกุล

สำหรับเอกสารที่เป็น .CW นั้นจำเป็นต้องเป็นประเภทที่พิมพ์ด้วยรหัส สมอ. เท่านั้น เพราะ System font ที่เป็นรหัสภาษาไทยเป็นแบบ สมอ.

#### 4.6 ด้านเครื่องมือในแอปพลิเคชัน

มีเครื่องมือที่สร้างขึ้นมาเพื่อใช้จัดการเกี่ยวกับภาพ โดยสามารถคือข้อความลงไปบนหน้าจอเกี่ยวกับภาพได้ โดยมีพื้นที่จำกัดและตกแต่งวางตำแหน่งให้เหมาะสมได้ สามารถเลื่อนภาพและข้อความได้ นอกจากนี้ยังสามารถโหลดเพิ่มข้อมูลภาพที่มีอยู่แล้วขึ้นมาได้ การบันทึกเพิ่มข้อมูลที่เป็นภาพนี้มีการเก็บเป็นเพิ่มข้อมูลแบบกราฟฟิก โดยที่โครงสร้างการเก็บเพิ่มข้อมูลจะแยกเป็นส่วนสำหรับเก็บภาพและส่วนเก็บข้อความ

#### 4.7 โปรแกรมช่วยเหลือ

เมนูช่วย ( HELP ) ช่วยอธิบายการทำงาน ของแต่ละคำสั่งว่าทำงานอย่างไร ใช้สัญลักษณ์อะไร ซึ่งเมนูช่วยนี้สร้างขึ้นโดยมีลักษณะแบบไฮเปอร์เท็กซ์คือสามารถคลิกเมาส์ที่คำสั่งเป็นคีย์เวิร์ด เพื่อเชื่อมโยงไปยังส่วนอธิบายให้แสดงเนื้อหาขึ้นมา พร้อมทั้งรูปภาพอธิบายประกอบ

#### 4.8 ผลที่ได้จากการทำงานของระบบ

สามารถใช้งานโปรแกรมได้โดยที่โปรแกรมไม่หยุดการทำงานอย่างกะทันหัน และหลังจากออกจากโปรแกรมแล้วยังสามารถใช้งานโปรแกรมอื่นต่อได้อีก โดยที่โปรแกรมยังสามารถทำงานภายใต้วินโดว และไม่ทำให้เสียระบบ

## บทที่ 5

### สรุปผลและเสนอแนะ

#### 5.1 สรุปผล

ผลที่ได้จากการพัฒนาต้นแบบโปรแกรมจัดเอกสารบนวินโดวส์ โดยการนำ ทฤษฎีเชิงวัตถุ และทฤษฎีการเชื่อมต่อแบบหลายเอกสารมาใช้ ทำให้ได้เครื่องมือที่ช่วยให้ผู้ใช้สามารถสร้างและจัดการกับเอกสารได้ ซึ่งมีลักษณะเด่นที่สำคัญคือการนำเอาคุณสมบัติทางด้านออปเจ็ค มาใช้ในการสร้างการเชื่อมต่อแบบหลายเอกสาร ( Multirle Document Interface ) ทำให้สามารถเปิดเอกสารได้หลายหน้าจอ นอกจากนี้ยังสามารถทำการเปรียบเทียบเอกสารได้มากกว่าหนึ่งหน้าจอขึ้นไป ( tile ) มีการสร้างแถบเครื่องมือ โดยการใช้รูปภาพสื่อความหมายของคำสั่งทำให้ง่าย และสะดวกต่อการใช้งาน มีแถบเลื่อนหน้าจอ ( scroll bar ) ทำให้สามารถเลื่อนหน้าเอกสารที่ต้องการแสดงได้รวดเร็วขึ้น

ในการพิมพ์เอกสารสามารถติดตั้งประเภทของเครื่องพิมพ์ให้มีความเหมาะสม กับเครื่องพิมพ์ที่ใช้ และมีส่วนแสดงเอกสารแบบเต็มรูปแบบก่อนทำการพิมพ์ สำหรับผู้ใช้ที่ไม่มีความคุ้นเคยในการใช้โปรแกรม มีส่วนของเมนูช่วย ( HELP ) ช่วยอธิบายการทำงานของแต่ละคำสั่งว่าทำงานอย่างไร ใช้สัญลักษณ์อะไร ซึ่งเมนูช่วยนี้สร้างขึ้นโดยมีลักษณะแบบไฮเปอร์เท็กซ์คือสามารถคลิกเมาส์ที่คำซึ่งเป็นคีย์เวิร์ด เพื่อเชื่อมโยงไปยังส่วนอธิบายให้แสดงเนื้อหาขึ้นมา พร้อมทั้งรูปภาพอธิบายประกอบ นอกจากนี้ยังมีความสามารถอื่น ๆ ตามที่แสดงบนเมนูดังที่กล่าวมาแล้วข้างต้น

ท้ายสุดคือทำให้เข้าใจถึงหลักการของการเขียนโปรแกรมเชิงวัตถุ เข้าใจทฤษฎีเชิงวัตถุและทฤษฎีการเชื่อมต่อแบบหลายเอกสารได้มากขึ้น สามารถใช้เป็นแนวทางสำหรับผู้สนใจทำงานวิจัยในด้านที่เกี่ยวข้องได้ อย่างเช่น ไฮเปอร์มีเดีย

ข้อจำกัดของโปรแกรม เนื่องจากการพัฒนาโปรแกรมนี้เป็นการพัฒนาทางด้านเท็กซ์ โหมดเท่านั้น เอกสารที่จะนำมาจัดการนั้นจึงมีข้อจำกัดทางด้านรูปแบบอักขระอยู่มาก จึงทำให้ไม่สะดวกในบางครั้งสำหรับเอกสารที่มีรูปแบบอักขระแตกต่างจากที่กำหนดไว้

## 5.2 แนวทางการวิจัยและข้อเสนอแนะ

เนื่องจากปัญหาพิเศษในหัวข้อ การพัฒนาต้นแบบโปรแกรมจัดเอกสารบนวินโดวส์เป็นการศึกษาเพียงส่วนหลัก ๆ บางส่วนเท่านั้น เนื่องจากระยะเวลาในการทำปัญหาพิเศษค่อนข้างจำกัด ดังนั้นจึงขอเสนอเพื่อให้ผู้ที่มีความสนใจในปัญหาพิเศษหัวข้อนี้ได้ทำการศึกษาและพัฒนาต่อไปดังนี้

### 5.2.1 ด้านรูปแบบตัวอักษรและภาพประกอบเอกสาร

เนื่องจากโปรแกรมที่พัฒนาขึ้นมาเน้นจัดทำเฉพาะทางด้านเท็กซ์โหมด ทำให้ไม่สามารถเปลี่ยนรูปแบบตัวอักษร หรือโหลดภาพขึ้นมาแทรกบนเอกสารที่เป็นข้อความได้ หรือเปิดไฟล์ข้อความที่มีรูปแบบการเก็บข้อมูลแบบกราฟฟิกโหมด เนื่องจากการเปลี่ยนรูปแบบตัวอักษร การเปิดเพิ่มข้อมูลแบบกราฟฟิก และการโหลดภาพนั้นต้องอาศัยคุณสมบัติทางด้านกราฟฟิกโหมด ซึ่งมีรูปแบบการบันทึกเพิ่มข้อมูล และโครงสร้างการเขียนโปรแกรมแตกต่างจากเท็กซ์โหมดมาก เพราะต้องมีการกำหนดรูปแบบตัวแปรในโปรแกรมใหม่ให้เป็นแบบเรคคอร์ด และมีการจองบัพเฟอร์โดยที่ต้องแบ่งบัพเฟอร์นั้นเป็นแต่ละส่วนโดยเฉพาะ สำหรับการบันทึกข้อมูลแต่ละเพิ่มข้อมูล ทำให้ต้องมีการศึกษาเพิ่มเติมอีกเป็นจำนวนมาก เมื่อได้ทำการศึกษาแล้วยังไม่สามารถทำได้สมบูรณ์ จึงได้เสนอเป็นข้อเสนอแนะ สำหรับผู้ที่สนใจในการนำความสามารถทางด้านกราฟฟิกโหมดมาช่วยสนับสนุนโปรแกรมได้ ถ้าหากนำความสามารถทางด้านกราฟฟิกโหมดมาช่วยสนับสนุนโปรแกรมนี้ได้ ก็จะทำให้สามารถเปลี่ยนรูปแบบตัวอักษรได้มากขึ้น เปิดเพิ่มข้อมูลที่มีรูปแบบการเก็บเพิ่มแบบกราฟฟิกโหมด และสามารถโหลดภาพเข้ามาเพิ่มได้ด้วย

### 5.2.2 ด้านของการประมวลผล

เนื่องจากโปรแกรมนี้พัฒนาภายใต้โปรแกรมภาษาบอร์แลนด์ C++ เวอร์ชัน 4.0 ซึ่งเป็นโปรแกรมที่มีขนาดใหญ่ ทำให้ใช้เวลานานในการโหลดแต่ละครั้ง นอกจากนี้ยังมีการใช้คลาสไลบรารีของ OWL ซึ่งต้องมีการเรียกใช้ไฟล์หลัก (Header file) จึงทำให้ใช้เวลาในการประมวลผลนานมากกว่าการพัฒนาภายใต้โปรแกรมภาษาอื่น ถ้าหากสามารถนำไปประยุกต์ใช้กับโปรแกรมภาษาอื่นที่สามารถอำนวยความสะดวกได้ในด้านเวลา

และมีเครื่องมือที่ใช้ในการสนับสนุนการเขียนโปรแกรมได้ดีกว่าโปรแกรมภาษาบอร์แลนด์ C++ เวอร์ชัน 4.0 ก็จะสามารถทำให้พัฒนาโปรแกรมได้ดียิ่งขึ้น

#### 5.2.4 ด้านทฤษฎี

สามารถนำคุณสมบัติการสืบทอดลูกหลานของทฤษฎีเชิงวัตถุ ไปใช้ในการสร้างเครื่องมือ หรือแอปพลิเคชันบนวินโดวส์ ที่มีความสามารถพิเศษอื่น ๆ นอกเหนือจากที่เคยพบเห็นได้ หรือใช้เป็นแนวทางในการศึกษาเพิ่มเติม

#### 5.2.3 ด้านการเปิดเพิ่มข้อมูลประเภทต่าง ๆ

ในการเปิดเพิ่มข้อมูลที่มีรูปแบบการเก็บเพิ่มต่างกัันนั้น ต้องมีการแปลงรูปแบบข้อมูลให้เหมือนกับรูปแบบของโปรแกรมที่ใช้อยู่ ดังนั้น สำหรับการเปิดเพิ่มข้อมูลแบบกราฟฟิกใหม่ โดยใช้โปรแกรมการพัฒนาต้นแบบการจัดเอกสารบนวินโดวส์ จึงยังไม่สามารถทำได้ เพราะโปรแกรมนี้มีรูปแบบการเก็บเพิ่มข้อมูลเป็นแบบเท็กซ์ใหม่ เราจึงต้องมีการสร้างส่วนกราฟฟิกใหม่ขึ้นมาเพิ่มเติม เพื่อให้สามารถรองรับเพิ่มข้อมูลที่มีรูปแบบต่างออกไปได้ เช่น ทำให้สามารถเปิดเพิ่มข้อมูลที่มีนามสกุล .DOC ซึ่งสร้างจากไมโครซอฟท์เวิร์ดได้

ภาคผนวก

## ภาคผนวก

### การติดตั้งระบบ

#### HARDWARE.

- เครื่องคอมพิวเตอร์ที่มี CPU ตั้งแต่ 386 ขึ้นไป
- มี Hard disk
- มีหน่วยความจำหลัก (Ram) 4 เมกะไบต์ขึ้นไป
- มีเครื่องพิมพ์

#### SOFTWARE.

- Microsoft DOS Version 5 ขึ้นไป
- Microsoft Windows Thai Edition Version 3.1 ขึ้นไป

### ไฟล์ที่ใช้

#### ไฟล์หลักที่ใช้

- Word.exe
- Playsnd.exe
- ImgDraw.exe
- Clipbrd.exe

#### ไฟล์รูปภาพที่ใช้ในโปรแกรมเครื่องมือวาดภาพ ( ImgDraw.exe )

- |               |              |
|---------------|--------------|
| - dash.bmp    | - rectdn.bmp |
| - selpcol.bmp | - test.bmp   |
| - txtup.bm    | - dashd.bmp  |
| - dots.bm     | - lineup.bmp |

- penwidth.bmp
- openf.bmp
- selup.bmp
- dashds.bmp
- pencildn.bmp
- selbcol.bmp
- tpb.bmp
- pencilup.bmp
- seldn.bmp
- txtdisp.bmp
- dot.bmp
- penstyle.bmp
- selfont.bmp
- txttn.bmp
- elpsdn.bmp
- rectup.bmp
- test2.bmp
- elpsup.bmp
- pwmrk.bmp
- showcolr.bmp
- dashes.bmp
- pwmrkm.bmp
- solid.bmp
- cpdisp.bmp
- linedn.bmp
- quit.bmp
- solids.bmp

#### ไฟล์ที่ใช้ในเมนูช่วยเหลือ ( Help File )

- help.hlp
- clipbrd.hlp
- imdrwhlp.hlp

## บรรณานุกรม

Herbert Schildt , TURBO C++ for Windows inside&out : Osborne McGraw-Hill

2600 Tenth Street Berkeley, California 94710 U.S.A.

Lee Adams , High-Performance C Graphics Programming for Windows

: Windcrest / McGraw-Hill

Tom Swan , Mastering Windows Programming with Borland C++ 4

Microsoft Press, Microsoft Windows 3.1 Guide to Programming

ขจร โรจนเมธินทร์, เทคนิคการเขียนโปรแกรม Turbo C++ สำหรับวินโดวส์

: บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน)

จිරพัฒน์ จันทรเจดศักดิ์, วีระ นพนิราพาธ, เขียนโปรแกรมบน Microsoft

Windows : บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน)

คอมพิวเตอร์วิวด ฉบับที่ 96 ประจำเดือน สิงหาคม 2535

หน้า 164 ถึง 193

คอมพิวเตอร์วิวด ฉบับที่ 115 ประจำเดือน เมษายน 2537

หน้า 201 ถึง 213

คอมพิวเตอร์วิวด ฉบับที่ 116 ประจำเดือน เมษายน 2537

หน้า 205 ถึง 225