สเปรดชีตบนระบบเอ็กซ์วินโดว์

SPREADSHEET ON X WINDOW SYSTEM

โดย

นายโกศล ฉกรรจ์ศิลป์

นายบุญยงค์ แก้วบุดดี

น.ส.ปริญดา เมฆรา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2534

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

007670

ว.188

สเปรดชีตบนระบบเอ็กซ์วินโดว์

Spreadsheet On X Window System

นายโกศล ฉกรรจ์ศิลป์ รหัส 31.1014

Mr. Kosol Chakansin

นายบุญยงค์ แก้วบุดดี รหัส 31.1139

Mr. Boonyong Kaewbuddee

นางสาวปริญดา เมฆรา รหัส 31.1147

Ms. Parinda Mekara

อาจารย์ที่ปรึกษา

ดร. บุญธีร์ เครือตราชู

Advisor

Dr. Boontee Kruatrachue

ปริญญานิพนธ์เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2534

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Spreadsheet On X Window System

Mr. Kosol Chakansin

Mr. Boonyong Kaewbuddee

Ms. Parinda Mekara

Advisor

Dr. Boontee Kruatrachue

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE BACHELOR OF ENGINEERING

ON ELECTRICAL ENGINEERING

GRADUATE SCHOOL

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

1991

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	สเปรดชีตบนระบบเอ็กซ์วินโดว์
นักศึกษา	นายโกศล ฉกรรจ์ศิลป์ 31.1014
	นายบุญยงค์ แก้วบุดดี 31.1139
	นางสาวปริญดา เมฆรา 31.1147
อาจารย์ผู้ควบคุมปริญญานิพนธ์	ดร. บุญธีร์ เครือตราชู
ระดับการศึกษา	วิศวกรรมศาสตร์บัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา	2534

บทคัดย่อ

โครงงานสเปรดชีตบนวินโดว์ มีวัตถุประสงค์เพื่อที่จะพัฒนาซอฟต์แวร์เพราะว่าในอนาคตจะมีลิขสิทธิ์ซอฟต์แวร์ขึ้น ทั้งยังเป็นการศึกษาหลักการทำงานของระบบเอ็กซ์วินโดว์ การพัฒนาโปรแกรมการใช้งานบนระบบยูนิกซ์ ในการทำครั้งนี้ได้ออกแบบโครงสร้างข้อมูลอยู่สามแบบ แต่ละแบบก็มีข้อดีข้อเสียต่างกัน แบบที่ถูกเลือกมาใช้นี้เป็นแบบที่ทำงานได้ดีที่สุด คือสามารถอ้างเซลล์ได้อย่างรวดเร็ว และประหยัดหน่วยความจำ การติดต่อกับผู้ใช้ก็ช่วยให้ผู้ใช้ทำงานได้ง่ายขึ้นร่วมกับเมาส์และคีย์บอร์ด มีเมนูให้เลือกใช้ฟังก์ชั่นและสามารถคำนวณอย่างง่าย ๆ ได้ ซอฟต์แวร์นี้ได้ออกแบบมาเพื่อที่จะง่ายต่อการพัฒนาในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนำ

ในสภาพการณ์ปัจจุบัน จะเห็นได้ว่าราคาซอฟต์แวร์มีแนวโน้มสูงขึ้นเรื่อย ๆ เมื่อเทียบกับราคาของเครื่องคอมพิวเตอร์ และหลายประเทศก็มีกฎหมายลิขสิทธิ์ซอฟต์แวร์เข้ามาใช้ รวมทั้งประเทศไทยด้วย นับได้ว่ามีผลกระทบต่อวงการซอฟต์แวร์เป็นอย่างมาก การพัฒนาซอฟต์แวร์ใช้เองจึงเป็นแนวทางแก้ไขปัญหาที่ดีและควรได้รับการส่งเสริมอย่างกว้างขวาง ซอฟต์แวร์ประเภทสเปรดชีตเป็นที่นิยมใช้กันอย่างกว้างขวางในวงการธุรกิจ ถ้าหากซอฟต์แวร์ประเภทนี้ได้รับการพัฒนาให้ใช้งานได้สะดวกและอำนวยประโยชน์ต่อผู้ใช้ให้มากยิ่งขึ้นก็นับว่าเป็นสิ่งที่มีประโยชน์ เพื่อตอบสนองต่อแนวความคิดดังกล่าว โครงงานนี้จึงได้พัฒนาโปรแกรมสเปรดชีตบนระบบเอ็กซ์วินโดว์ขึ้นมา การเข้าถึงข้อมูลได้อย่างรวดเร็วและใช้หน่วยความจำอย่างประหยัดเป็นคุณสมบัติที่ดีของซอฟต์แวร์สเปรดชีต โครงงานนี้จึงได้ออกแบบโครงสร้างข้อมูลให้เหมาะสมคือใช้โครงสร้างข้อมูลแบบลิงค์ลิสต์ของแอเรย์ขนาดสิบคูณสิบ ซึ่งส่งผลให้ประสิทธิภาพการทำงานดีกว่า เร็วกว่า และประหยัดหน่วยความจำได้มาก การอ้างอิงเซลล์ทำได้โดยใช้ระบบหลักและแถว ทั้งหมดห้าร้อยหลักและสองพันแถว สามารถอ้างอิงเซลล์ได้ทั้งหมดหนึ่งล้านเซลล์ นอกจากนี้การใช้เมาส์ร่วมกับคีย์บอร์ดยังช่วยในเพิ่มความเร็วและความสะดวกในการทำงานได้อีกด้วย

ระบบเอ็กซ์วินโดว์เป็นระบบการทำงานแบบผู้ใช้หลายคน มีรูปแบบการใช้งานที่สะดวกเพราะระบบอินเตอร์เฟสที่ทันสมัย เป็นที่นิยมใช้กันทั่วไป จึงได้นำมาพิจารณาประยุกต์ใช้ในการพัฒนาสเปรดชีตชุดนี้ มากกว่านี้ โครงงานนี้ยังเป็นการส่งเสริมให้นักศึกษาได้ฝึกทำงานเป็นกลุ่ม เพิ่มระดับความสามารถในการโปรแกรมบนระบบยูนิกซ์และระบบวินโดว์อีกด้วย

คณะผู้จัดทำ

18 มีนาคม 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	3
บทคัดย่อภาษาอังกฤษ	4
คำนำ	5
สารบัญ	6
สารบัญภาพ	11
สารบัญตาราง	12
บทนำ	
จุดประสงค์และโครงสร้างของปริญญานิพนธ์	13
บทที่ 1 ประวัติความเป็นมาของเอ็กซ์วินโดว์	14
บทที่ 2 ความรู้พื้นฐานของระบบเอ็กซ์วินโดว์	
2.1 ความรู้พื้นฐานของระบบเอ็กซ์วินโดว์	17
2.1.1 ดิสเพลย์และสกรีน	17
2.1.2 การทำงานของเซอร์เวอร์และไคลแอนท์	17
2.1.3 การจัดการวินโดว์	19
2.1.4 เหตุการณ์	19
2.2 สถาปัตยกรรมของเอ็กซ์วินโดว์	19
2.3 เอ็กซ์ไลบรารี่	22
บทที่ 3 หลักการทำงานของเอ็กซ์	
3.1 การทำงานของเอ็กซ์ไลบรารี่	24
3.2 เอ็กซ์โปรโตคอล	26
3.2.1 คำขอร้อง	26
3.2.2 คำตอบรับ	26
3.2.3 การแจ้งเหตุ	27
3.2.4 การแจ้งเหตุข้อผิดพลาด	27
3.3 ระบบคำร้องขอในเอ็กซ์ไลบรารี่	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ทรัพยากร 29

3.5 พรอบเพอตี้และอะตอม 29

3.6 วินโดว์แมนนาเจอร์ 30

3.7 วินโดว์คืออะไร 30

3.8 ลักษณะของวินโดว์ 30

3.9 แผนภูมิต้นไม้แบบลำดับชั้น 33

3.10 ระบบพิกัด 35

3.11 ลำดับการเก็บวินโดว์ในสแตก 35

3.12 แมบปิ้งและการปรากฏ 36

3.13 กราฟฟิคของเอ็กซ์ 36

3.13.1 พิกเซลและสี 36

3.13.2 พิกเซลและแพลน 38

3.13.3 พิกแมบและดรอพเอเบิล 39

3.13.4 การวาดและองค์ประกอบ 40

3.13.5 ไทล์และสติปเปิ้ล 40

3.13.6 ลักษณะอื่น ๆ ของวินโดว์ 41

3.14 ลักษณะเฉพาะของวินโดว์ 41

3.15 โครงสร้างของวินโดว์ 43

3.16 คลาสของวินโดว์แบบอินพุทเอาท์พุทและอินพุททางเดียว 43

3.17 เคบและวิชวล 43

3.18 ไอคอน 44

3.19 ลักษณะพิเศษของวินโดว์ราก 45

3.20 อีเวนต์คืออะไร 45

3.21 การเลือกเหตุการณ์มาเก็บไว้ในคิว 45

3.22 คิวของเหตุการณ์ 47

3.23 โครงสร้างของเหตุการณ์ 47

3.24 ลูปของเหตุการณ์ 48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 โครงสร้างข้อมูล

4.1 แบบที่ 1 49

4.2 แบบที่ 2 50

4.3 แบบที่ 3 51

การแทรกแถว,การแทรกหลัก,การลบแถว,การลบหลัก 52

การเติมข้อมูล,การคัดลอกข้อมูล,การย้ายข้อมูล,การเก็บข้อมูล,

การดึงข้อมูลจากดิสก์ 53

บทที่ 5 อินเตอร์เฟส

5.1 เอ็กซ์วิว 54

5.2 ระบบเอกซ์วิวและเอกซ์วินโดว์ 54

5.2.1 พัฒนาการของเอกซ์วิวทูลคิท(Xview toolkit) 54

5.2.2 เอ็กซ์วิวจัดให้มีชุดของวินโดว์ต่อไปนี้ 55

5.3 แบบจำลองการโปรแกรมกับเอกซ์วิว 55

5.3.1 การโปรแกรมเชิงวัตถุ 55

5.3.1.1 คลาสอ็อปเจ็คแบบลำดับชั้น 56

5.3.1.2 การรับส่งอ็อปเจ็ค 57

5.4 ฟังก์ชั่นคุณลักษณะเฉพาะพื้นฐาน 58

5.4.1 การสร้างและการใช้งานอ็อปเจ็ค 58

5.4.2 ชนิดของลักษณะเฉพาะ 59

5.4.3 ชนิดของอ็อปเจ็ค 59

5.5 การสร้างโปรแกรมประยุกต์ในระบบเอกซ์วิว 60

5.5.1 การเชื่อมต่อ 60

5.5.2 การเริ่มเอ็กซ์วิว 61

5.5.3 การสร้างและการแก้ไขอ็อปเจ็ค 62

5.6 งานประยุกต์ใช้เอกซ์วิวสร้างสเปรดชีต 64

5.6.1 ส่วนเชื่อมต่อผู้ใช้ 64

5.6.2 เมนู 64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.6.3 เครื่องมือที่นำมาช่วยในการพัฒนาโปรแกรม 73

5.6.4 ส่วนของชีดและการแสดงเซลล์ 74

5.6.5 การแสดงสี 75

5.6.6 การใช้สกอลบาร์ 76

5.6.7 การรับอินพุทจากผู้ใช้ 77

5.6.8 การออกจากการทำงาน 78

บทที่ 6 ส่วนรับข้อมูล

6.1 ตัววิเคราะห์คำ 81

6.1.1 การกำหนดโทเคน 81

6.1.2 การทำงานในขั้นตอนการวิเคราะห์คำ 82

6.1.3 รูปแบบของโปรแกรม Lex 82

6.2 ตัววิเคราะห์ไวยากรณ์ 84

6.2.1 รูปแบบโปรแกรม Yacc 84

6.2.2 ส่วนการประกาศ 84

6.2.3 ส่วนการสร้างกฎ 85

6.2.4 ส่วนรูทีนสนับสนุน 86

6.3 การผ่านค่าลักษณะเฉพาะจากตัววิเคราะห์คำ 87

6.3.1 ชนิดของข้อมูล 88

6.3.2 การป้อนข้อมูล 89

6.3.2.1 ขั้นตอนการรับข้อมูล 89

6.3.2.2 ขั้นตอนการแปลความหมาย 90

6.3.2.3 ขั้นตอนการคำนวณค่า 90

6.4 ขั้นตอนการค้นหาโหนดเพื่อทำการคำนวณใหม่ 95

บทที่ 7 สรุปผลโครงงาน 98

หนังสืออ้างอิง 100

กิตติกรรมประกาศ 101

ภาคผนวก ก โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

007670

ภาคผนวก ข คำสั่งยูนิกซ์

ภาคผนวก ค โฟล์ชาร์ต

ภาคผนวก ง โปรแกรมแยกคำและวิเคราะห์คำ

ภาคผนวก จ คู่มือการใช้งานโปรแกรม

---***---

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปประกอบ

รูปที่			หน้า
รูปที่	2.1	แสดงดิสเพลย์ที่ประกอบด้วยหลายสกรีน	17
รูปที่	2.2	แสดงการรันโปรแกรมข้ามเน็ตเวอร์ค	18
รูปที่	2.3	สถาปัตยกรรมของระบบเอ็กซ์วินโดว์	20
รูปที่	3.1	แสดงไคลแอนท์และเซอร์ฟเวอร์	25
รูปที่	3.2	แสดงการจัดคิวคำร้องและคำตอบรับในระบบเอ็กซ์วินโดว์	28
รูปที่	3.3	แสดงโครงสร้างทั่วไปของวินโดว์	32
รูปที่	3.4	แสดงลำดับชั้นของวินโดว์ที่ปรากฏบนจอ	33
รูปที่	3.5	แสดงไดอะแกรมของลำดับชั้นวินโดว์	34
รูปที่	3.6	แสดงระบบพิกัดของเอ็กซ์	35
รูปที่	3.7	แสดงการแปลงค่าพิกเซลไปเป็นตารางสี	37
รูปที่	3.8	แสดงการใช้ไทล์ระบายลงเป็นสีพื้นของวินโดว์	41
รูปที่	3.9	แสดงวินโดว์ชื่อ Xterm ก่อนและหลังการ iconify	44
รูปที่	4.1	แสดงโครงสร้างข้อมูลแบบอะเรย์ของหลักชี้ไปยังอะเรย์ของแถว	49
รูปที่	4.2	แสดงโครงสร้างข้อมูลแบบอะเรย์ของแถวและหลักชี้ไปยังเซลล์เดี่ยวๆ	50
รูปที่	4.3	แสดงโครงสร้างข้อมูลแบบอะเรย์ของแถวและหลักชี้ไปยังกลุ่มเซลล์	51
รูปที่	5.1	ลักษณะของเซลล์ที่ปรากฏบนวินโดว์	75
รูปที่	5.2	แสดงเฟรมคำสั่ง	78
รูปที่	6.1	แสดงการกระจายคำโดยใช้ Lex และ Yacc	80
รูปที่	6.2	แสดงการย้ายข้อมูลในกรณีที่อินพุทเป็นข้อมูลชนิดตัวเลข	91
รูปที่	6.3	แสดงการย้ายข้อมูลในกรณีที่อินพุทเป็นข้อมูลชนิดสูตร	93
รูปที่	6.4	แสดงการย้ายข้อมูลในกรณีที่อินพุทเป็นข้อมูลชนิดเทกซ์	94

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
ตาราง 2.1	แสดงฟังก์ชั่นในเอ็กซ์ไลบรารี่	23
ตาราง 5.1	รายการของเอกซ์วิวอ็อปเจค, เจ้าของอ็อปเจค, แพคเกจที่กำหนด, และรูปแบบของข้อมูล	57
ตาราง 5.2	ชุดฟังก์ชันหลักของเอกซ์วิว	58
ตาราง 5.3	แสดงค่าอุปสรรคที่สงวน	61

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

--------- *** ---------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

โปรแกรมสเปรดชีต

โปรแกรมสเปรดชีตเป็นโปรแกรมที่สำคัญมากที่จะจุดประกายการนำไมโครคอมพิวเตอร์เข้ามาใช้ในวงการธุรกิจ เพราะก่อนหน้าที่จะเกิดโปรแกรมสเปรดชีต การใช้ไมโครคอมพิวเตอร์ยังใช้เพื่อเขียนโปรแกรมประยุกต์ การใช้เวิร์ดโปรเซสเซอร์เก็บเอกสาร และการเล่นเกมส์เท่านั้น แต่เมื่อเกิดโปรแกรมสเปรดชีตที่มีการจำลองการทำงานบนโต๊ะ โดยมีกระดาษขนาดใหญ่วางไว้ให้และมีเครื่องมือคล้ายปากกา ยางลบและเครื่องคำนวณเตรียมให้เสร็จ เราผู้ใช้ก็เพียงสร้างหรือขีดเขียนและคำนวณงานบนกระดาษนี้ ลักษณะของโปรแกรมจะอยู่ที่การคำนวณใหม่ ซึ่งเราเปลี่ยนแปลงข้อมูลบนกระดาษสูตรการคำนวณที่นำข้อมูลนั้นไปใช้จะคำนวณให้ผลลัพธ์ทันที ทำให้โปรแกรมนี้ทำงานในเครื่องมือวิเคราะห์ What-if ได้ดีมาก เพราะในทางธุรกิจ เรามักจะมีคำถามที่เป็น What-if เสมอ เช่น ถ้าเราต้องการกำไรมากขึ้นเป็น 10 เปอร์เซนต์จากเดิม 8 เปอร์เซนต์ ก็จะต้องเปลี่ยนแปลงรายจ่ายด้านต่าง ๆ อย่างไร ถ้าเราจ้างคนมากขึ้นอีกหนึ่งคน จะดีกว่าการให้มีล่วงเวลาอย่างไร เป็นต้น

ข้อดีของโปรแกรมสเปรดชีตจะช่วยงานด้านคำนวณตัวเลขได้ดี เพราะภายในโปรแกรมจะมีสูตรคำนวณด้านต่างๆไว้ให้แล้ว ไม่ว่าจะเป็นฟังก์ชันทางคณิตศาสตร์ สถิติ การเงิน และฐานข้อมูลผู้ใช้สามารถเรียกใช้จากที่มีอยู่และสามารถสร้างสูตรคำนวณขึ้นมาเสริมอีกก็ได้ นอกจากนี้โปรแกรมสเปรดชีตจะช่วยลดข้อผิดพลาดต่างๆให้น้อยลงและสามารถสร้างชิ้นงานที่สะอาดและสวยงาม เพราะเราสามารถจัดแบ่งขนาดของแถวและคอลัมน์ได้เอง โดยไม่ต้องใช้ไม้บรรทัดตีเส้นแบ่งระยะซ้ำแล้วซ้ำอีก

เราสามารถสร้างโมเดลชิ้นงานต่างๆบนสเปรดชีตหรือกระดาษอิเลคทรอนิคส์ได้มากมายตามแต่การประยุกต์ใช้ของเราเอง เช่น การทำระบบบัญชีรับจ่าย บัญชีเงินเดือน และบัญชีที่ซับซ้อนอื่นๆ ดังนั้นโปรแกรมสเปรดชีตจึงได้รับความสนใจนำไปใช้งานต่างๆตั้งแต่สำนักงานจนถึงห้องปฏิบัติการ เพื่อเก็บและวิเคราะห์ข้อมูลมาสร้างกราฟและจัดระบบฐานข้อมูลได้อีกด้วย ในปัจจุบันโปรแกรมสเปรดชีตที่มีขีดความสามารถเด่นขึ้นไปอีก ในด้านการนำภาพและกราฟิกเข้ามารวมกับข้อมูลในสเปรดชีต สามารถปรับเปลี่ยนรูปแบบและแอตทริบิวต์ของตัวอักษรและนำงานออกมาพิมพ์ที่เครื่องพิมพ์อย่างสวยงาม เหมือนงานจากโรงพิมพ์ โปรแกรมสเปรดชีตรุ่นใหม่สนับสนุนการใช้เมาส์ และระบบติดต่อกับผู้ใช้ที่ง่ายและสะดวก ปัจจุบันมักจะเห็นสเปรดชีตปรากฏอยู่บนระบบวินโดว์ต่าง ๆ ซึ่งแนวทางการพัฒนาจะอำนวยความสะดวกต่อผู้ใช้ได้มากขึ้นและสวยงาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์เล่มนี้ได้บรรจุเนื้อหาของสเปรดชีตตั้งแต่หลักการออกแบบและการพัฒนาบนระบบเอ็กซ์วินโดว์ ตลอดจนการประยุกต์ใช้งานสำหรับผู้ใช้ โดยเนื้อหาภายในมีดังนี้

บทที่ 1 ประวัติของเอ็กซ์วินโดว์

บทที่ 2 ความรู้พื้นฐานของระบบเอ็กซ์วินโดว์

-ดิสเพลย์และสกรีน

-การทำงานของเซอร์เวอร์และไคลแอนท์

-การจัดการวินโดว์

-เหตุการณ์

-สถาปัตยกรรมของระบบเอ็กซ์วินโดว์

บทที่ 3 เอ็กซ์คอนเซ็บท์(X Concepts)

-การใช้งานเอ็กซ์ไลบรารี่

-เอ็กซ์โปรโตคอล

-บัฟเฟอริ่ง

-ทรัพยากร

-พรอบเพอตี้และอะตอม

-ผู้จัดการวินโดว์

-ความหมายและลักษณะของวินโดว์

-แผนภูมิต้นไม้แบบลำดับชั้น

-ระบบพิกัด

-ลำดับการเก็บวินโดว์ในสแตก

-การแปลงค่าและการเห็น

-กราฟฟิกของเอ็กซ์

-ลักษณะเฉพาะของวินโดว์

บทที่ 4 โครงสร้างข้อมูลและการออกแบบ(Data structure and design)

บทที่ 5 อินเตอร์เฟส(Interface)

-ส่วนเชื่อมต่อผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-เมนูและเครื่องมือที่นำมาช่วยในการพัฒนาเมนู

-ส่วนของเวอร์คชีตและการแสดงเซลล์

-การแสดงสี

-การใช้สกอลล์บาร์

-การรับข้อมูลคำสั่งและการออกจากการทำงาน

บทที่ 6 ส่วนแปลความหมายของข้อมูล

-ขั้นตอนการกำหนดขอบเขตของข้อมูลที่จะรับ

-แยกและแปลความของข้อมูล

-ส่วนคำนวณทางคณิตศาสตร์และทางตรรกศาสตร์

บทที่ 7 บทสรุปผลโครงงาน

----------***----------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

ประวัติความเป็นมาของ X window

ประวัติของเอ็กซ์วินโดว์ (X window) เริ่มเมื่อปี 1982 เมื่อพอล อเซ็นต์ (Paul Asente) และเบรียน รีด(Brien Reid) แห่งมหาวิทยาลัยสแตนฟอร์ดได้พัฒนาระบบวินโดว์อิ้ง ฟ (Windowing system) ขึ้นมาเป็นครั้งแรก ในรูปของระบบเน็ตเวอร์วินโดว์ที่เป็นอะซิงโครนัส (Asynchronous) ที่วิ่งใต้ระบบงานของสแตนฟอร์ด (Stanford's V operating system) หลังจากนั้นหนึ่งปี อเซ็นต์กับรีดก็ได้แปลงวินโดว์ ฟ ให้ไปอยู่ภายใต้ระบบงานยูนิกซ์ และในเวลาเดียวกันก็ได้มอบหมายวินโดว์อิ้ง ฟ ที่เป็นยูนิกซ์ให้กับ จิม เก็ตตี้ (Jim Gettys) และโรเบิร์ต ไชฟ์เลอร์ (Robert Scheifler) ผู้ซึ่งจะมาเป็นสถาปนิกซอฟต์แวร์ในเอ็กซ์วินโดว์ ในกาลต่อมา ประมาณ 1 ปี ผ่านไป ราวเดือนพฤษภาคม ค.ศ 1983 ทั้งสองได้แปลง ฟ ให้เป็นระบบอะซิงโครนัสที่เรียกว่าระบบเอ็กซ์วินโดว์ เหมือนชื่อที่เรียกในตอนแรก เพราะเก็ตตี้และไชฟ์เลอร์อยากจะตั้งชื่อระบบเสียใหม่ให้ดีกว่านี้ แต่จนแล้วจนรอดก็ไม่มีโอกาสที่จะหาชื่อที่ดีกว่านี้ได้สำเร็จ ในช่วงปี ค.ศ. 1985 วิศวกรของเอ็มไอทีและเด็คได้ร่วมกันพัฒนาเอ็กซ์วินโดว์ และในขณะนั้นเอ็มไอทีก็ได้รับความช่วยเหลือหลายอย่างจากไอบีเอ็มในโครงการ Athena เพื่อการสร้างคอมพิวเตอร์เน็ตเวอร์กให้แผนกต่างๆ ของเอ็มไอทีได้มีโอกาสใช้พีซี และเวอร์กสเตชั่นกันมากขึ้น เนื่องจากว่าที่แล้วๆ มามีเพียงแผนกคอมพิวเตอร์ศาสตร์กับวิศวกรรมไฟฟ้า สองหน่วยงานของมหาวิทยาลัยแห่งนี้ที่มีคอมพิวเตอร์ใช้มากที่สุด จนแผนกอื่นถูกลืมเลือน จุดนี้เองที่เป็นที่มาของโครงการ Athena ที่มีส่วนบันดาลให้เกิดเอ็กซ์วินโดว์ขึ้นมาเป็นผลพลอยได้ จนในเดือนกันยายน 1985 เอ็มไอทีก็ได้ประกาศตัวเอ็กวินโดว์ เวอร์ชัน 9.0 ออกมาให้โลกรู้จัก ซึ่งสามารถใช้กับ VAX station II จากนั้นก็มีรีลีส (Release) อื่นๆ ของเอ็กซ์วินโดว์ประกาศตามกันออกมาเรื่อยๆ จนมาถึงปัจจุบันคือ เอ็กซ์วินโดว์ เวอร์ชัน 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

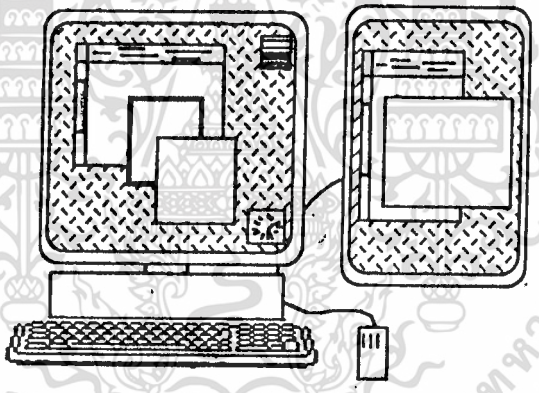
บทที่ 2

ความรู้พื้นฐานของระบบ X Window

2.1 ความรู้พื้นฐานของระบบเอ็กซ์วินโดว์ สามารถแบ่งได้เป็นส่วน ๆ ดังต่อไปนี้

2.1.1 ดีสเพลย์และสกรีน (Display และ Screen)

ดีสเพลย์ ประกอบไปด้วยอุปกรณ์ต่าง ๆ คือ คีย์บอร์ด,เมาส์,และ สกรีน ซึ่ง สกรีนจะมีมากกว่า 1 อันก็ได้ สำหรับดีสเพลย์ที่มีหลายสกรีน สามารถที่จะทำงานร่วมกันได้ โดยมีผู้ใช้เพียงคนเดียว นั่นคือผู้ใช้จะใช้เมาส์ในการควบคุมว่าต้องการจะแสดงผลออกมาที่สกรีนใด เพราะเมาส์สามารถที่จะข้ามจากวินโดว์หนึ่งไปอีกวินโดว์หนึ่งได้



รูปที่ 2.1 แสดงดีสเพลย์ที่ประกอบด้วยหลายสกรีน

2.1.2 การทำงานของเซอร์ฟเวอร์และไคลแอนท์

โปรแกรมที่ใช้ในการควบคุมดีสเพลย์ เรียกว่า เซอร์ฟเวอร์ (Server) ซึ่งเซอร์ฟเวอร์จะเป็นตัวเชื่อมโยงระหว่างโปรแกรมผู้ใช้ (นั่นคือ ไคลแอนท์(Client) หรือ โปรแกรมประยุกต์ (Application)) กับส่วนต่าง ๆ ของเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

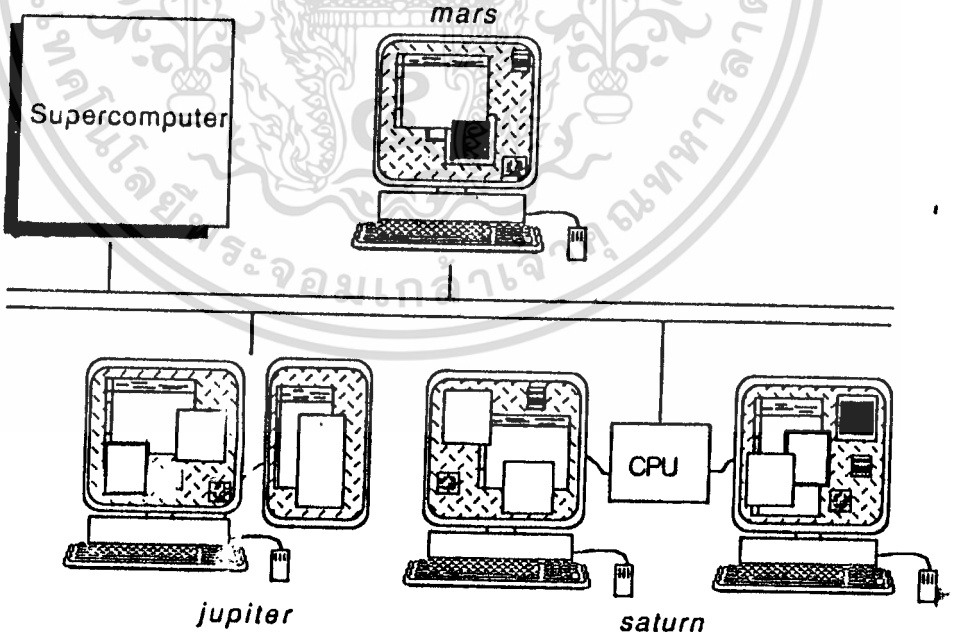
หน้าที่ของเซอร์ฟเวอร์มีดังนี้คือ

-ยอมให้ไคลแอนท์หลาย ๆ ตัว ทำงานบนดิสเพลย์เดียวกัน

-เป็นตัวรับอินเตอร์รัพท์จากไคลแอนท์

-เก็บค่า โคลงสร้างข้อมูล (data structure), วินโดว์ (including window), เคอร์เซอร์ (cursor),ฟอนท์ (font)และ กราฟฟิคคอนเทกซ์ (graphic context) เป็นเสมือนทรัพยากร (resource)ที่สามารถใช้ได้ทั้งไคลแอนท์และทรัพยากรไอดี (resource ID)

จากรูปที่ 2.2 จะเห็นได้ว่าโปรแกรมหลาย ๆ โปรแกรมของผู้ใช้จะติดต่อกันโดยผ่านทางดีสเพลย์ในเน็ตเวอร์ก (network) ถ้าเครื่องที่ผู้ใช้กำลังใช้งานอยู่คือโฮส (Host) ในระบบที่มีโฮสหลายเครื่องและกำลังรันโปรแกรมอยู่ในเครื่อง แต่ละโฮสจะสามารถทำงานได้ภายใต้เซอร์ฟเวอร์หลายตัว ถ้าเซอร์ฟเวอร์ตัวนั้นว่างอยู่ ระบบนี้เรียกว่าการประมวลผลแบบกระจาย (Distribute processing) ระบบนี้สามารถช่วยแก้ปัญหาในเรื่องความไม่สมดุลย์ของงานได้คือในขณะที่โฮสใดโฮสหนึ่งทำงานที่จะต้องทำมากมาย เมื่อเราต้องการจะจะโฮส ก็สามารถที่จะไปใช้โฮสอื่นแทนได้



รูปที่ 2.2 แสดงการรันโปรแกรมข้ามเน็ตเวอร์ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 การจัดการวินโดว์

ในการจัดการเกี่ยวกับตำแหน่งของวินโดว์ หรือขนาดของวินโดว์ที่จะวาดออกหน้าจอ เป็นหน้าที่ของวินโดว์แมนนาเจอร์ (Window manager) วินโดว์แมนนาเจอร์เป็นโปรแกรมอย่างหนึ่งที่เขียนด้วยเอกซ์ไลบรารี่ (X library) เป็นส่วนที่มีความสามารถพิเศษคือสามารถควบคุมโครงร่าง(layout) ของวินโดว์ได้ วินโดว์มนนาเจอร์จะยอมให้ผู้ใช้เปลี่ยนแปลงขนาด หรือย้ายวินโดว์ได้ หรือจะทำเป็นวินโดว์ใหม่ก็ได้เช่นกัน นอกจากนี้ยังทำการควบคุมลำดับของสแตก (stack order) ของวินโดว์ที่แสดงที่จอภาพด้วย โดยทุกอย่างที่วินโดว์เป็นตัวที่คอยกำหนดค่าสูงสุด (Maximum) ที่ยอมให้ทำได้

2.1.4 เหตุการณ์ (Event)

เหตุการณ์ เป็นเหตุการณ์ของอินพุทของยูสเซอร์ ซึ่งมีหลายเหตุการณ์ (การกดคีย์บอร์ด, การกดเมาส์, การเคลื่อนที่ของเมาส์, การเปลี่ยนแปลงของวินโดว์) และผู้เขียนโปรแกรมก็จะเป็นผู้ตองสนองเหตุการณ์นั้น ๆ ด้วยการเขียนโปรแกรมวนลูปคอยตรวจสอบ ว่าในขณะนั้นเกิดเหตุการณ์อะไรบ้าง และเมื่อมีเหตุการณ์ใดเกิดขึ้นจะต้องทำอย่างไรต่อไป ตัวอย่างเช่น เมื่อหน้าต่างเกิดการทับซ้อนกัน หรือเมื่อหน้าต่างถูกเปลี่ยนขนาด จะเกิดเอกซ์โพส (Expose Event) ขึ้น ดังนั้นไคลแอนท์จะต้องไปจัดการวาดหน้าต่างขึ้นมาใหม่

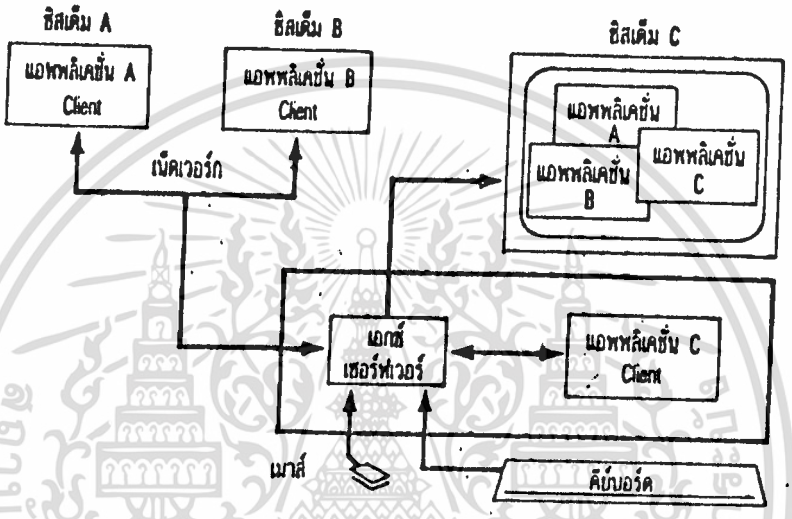
เหตุการณ์มีหลายชนิด และจะเกิดเรียงกันไป เหตุการณ์ที่เกิดขึ้นจะถูกนำไปเก็บไว้ในคิวตามลำดับ โดยไคลแอนท์จะใช้ฟังก์ชั่นในโปรแกรมเอกซ์(X program)เลือกว่าต้องการเหตุการณ์ชนิดใดบ้างมาเก็บไว้ในคิว และจัดการกับเหตุการณ์นั้นโดยโปรแกรมที่เขียนขึ้นมา

2.2 สถาปัตยกรรมของเอ็กซ์วินโดว์

เอ็กซ์วินโดว์แอพพลิเคชั่น(X Window application) ซึ่งวิ่งในรูปของกระบวนการไคลแอนท์-เซอร์ฟเวอร์(Client-Server) แต่ถ้าจะมองกลับเข้ามาภายในเอ็กซเน็ตเวอร์กแล้ว ความสัมพันธ์ของไคลแอนท์และเซอร์ฟเวอร์จะกลับกันกับที่เคยเป็นอยู่ในโมเดลทั่ว ๆ ไป คือไคลแอนท์-แมชชีน(Client - machine) จะเป็นที่สำหรับจัดการกับภาระต่าง ๆ ที่เกี่ยวกับแอพพลิเคชั่น แมเนจเมนต์ (application manager) และเป็นระบบใด ๆ ก็ได้ที่วิ่งบนเอกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไลบรารีที่สามารถเข้าถึงเอ็กซ์เน็ตเวอร์กโปรโตคอลซอฟต์แวร์ซึ่งหมายความว่าผู้ใช้สามารถปฏิบัติตามคำสั่งในโปรแกรม (execute) แอพพลิเคชั่น (application) อันหนึ่งบนระบบหนึ่งแล้วดิสเพลย์ แอพพลิเคชั่นอีกอันอีกระบบหนึ่ง หรือหลาย ๆ ระบบพร้อม ๆ กัน ดังในรูปที่ 2.3



รูปที่ 2.3 สถาปัตยกรรมของระบบเอ็กซ์วินโดว์

เซอร์เวอร์ดังในรูปที่ 2.3 จะเป็นระบบที่ประกอบด้วย คีย์บอร์ด กับมอนิเตอร์(monitor) ที่รับคำสั่งจากไคลแอนท์ รวมทั้งคำสั่งในการเริ่มตอน (Section) หนึ่ง ๆ ส่วนการเชื่อมโยงนั้นทำได้โดยไคลแอนท์เป็ผู้ออกชื่อของซอฟต์แวร์ พร้อมด้วยหมายเลขของเน็ตเวอร์ก หมายเลขประจำเครื่องโฮส และหมายเลขประจำเทอร์มินัล/สกรีน (terminal/screen) ซึ่งจะต้องทำเช่นนี้ให้ครบขั้นตอนทั้งหมด แม้กระทั่งว่าไคลแอนท์กับเซอร์เวอร์จะอยู่บนเน็ตเวอร์กสเตชันตัวเดียวกันก็ตาม ซึ่งวิธีการทั้งหมดนี้จะจัดการโดยสตรีม-เบส (stream-base) อินเตอร์เฟสคอมมิวนิเคชั่น (interface communication) ที่ยอมให้ไคลแอนท์ส่งสัญญาณแสดงข้อต้องการต่าง ๆ ออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปยังเซอร์ฟเวอร์โดยไม่ต้องมีคำตอบสำหรับข้อต้องการต่าง ๆ ที่ส่งออกไปแล้ว หลังจากที่ไคลแอนท์ได้กำหนดการเริ่มต้นสำหรับแอพพลิเคชั่นอันหนึ่งออกไปแล้ว ก็จะกลับเข้าสู่ลูป แล้วรอจนว่าตัวเซอร์ฟเวอร์จะแจ้งเหตุการณ์อย่างใดอย่างหนึ่งให้ทราบ

คราวนี้เมื่อไคลแอนท์ได้รับแจ้งเหตุการณ์ใด ๆ แล้ว มันจะเล็กชิคิวโค้ดกลุ่มหนึ่ง ที่โปรแกรมเมอร์(Programmer)ได้เขียนขึ้นมา สำหรับเหตุการณ์นั้น ๆ ซึ่งหลังจากที่ได้เล็กชิคิวโค้ดดังกล่าวแล้ว เอ็กซ์วินโดว์จะคืนการควบคุมไปให้ลูปหลัก(main loop) เพื่อรอเหตุการณ์ถัดต่อไป เหตุการณ์ในเอกซ์วินโดว์นั้นจะหมายถึงการเคลื่อนไหวของเมาส์,เคอร์เซอร์ นอกจากนั้นยังหมายถึงเหตุการณ์ที่เกิดขึ้นครั้งสุดท้ายของโปรแกรมหนึ่ง ๆ

ส่วนวิธีที่เอ็กซ์วินโดว์ ทำการประมวลผลนั้นจะอาศัยแบบจำลองเชิงวัตถุ(object-oriented model) ต่าง ๆ เข้าช่วยในแง่ของผู้ใช้โดยทั่วไปนั้น กระบวนการ ไคลแอนท์-เซอร์ฟเวอร์จะเป็นนามธรรม คือถ้าระบบอันใดอันหนึ่งมาพร้อมกับเดสค์ท้อปแมนนาเจอร์(Desktop manager) ซึ่งจะดิสเพลย์แอพพลิเคชั่นต่าง ๆ ตลอดจนไฟล์ข้อมูล ในรูปของไอคอน (icon) ในกรณีนี้ผู้ใช้ก็สามารถเริ่มแอพพลิเคชั่นอันหนึ่งโดยใช้เมาท์เป็นตัวคลิ๊ก (click) บนไอคอนที่แทนแอพพลิเคชั่นอันใดอันหนึ่ง ในกรณีที่ไม่มีเดสท้อปแมนนาเจอร์ผู้ใช้ก็จะต้องพิมพ์ชื่อของไฟล์สำหรับแอพพลิเคชันในวินโดว์ที่วิ่งอยู่บนยูนิกซ์เชลล์(Unix shell)นั่นคือ ผู้ใช้ไม่จำเป็นจะต้องรู้เรื่องเน็ตเวอร์กคอนเนกชัน (Network connection) หรือระบบดำเนินงานที่ไคลแอนท์กำลังวิ่งอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 เอ็กซ์ไลบรารี่ (overview of Xlib)

จากตารางที่ 2.1เป็นรูทีน (routine) ของ Xlib ตามกลุ่มฟังค์ชันหลัก ๆ

ฟังก์ชั่น(Function)	คำอธิบาย
Color	รูทีนที่ใช้ในการเปลี่ยนสี ที่ใช้ในการวาดแอพพลิเคชัน
Cursor	รูทีนที่ใช้ในการเปลี่ยนแปลงลักษณะรูปร่างของเคอร์เซอร์
Data Management	เครื่องมือที่ใช้ในการจัดการเกี่ยวกับข้อมูลของวินโดว์
Display Connection	รูทีนที่เชื่อมโยง และไม่เชื่อมโยงกับแอพพลิเคชัน
Display and Server-specification	มาโครและฟังชัน ที่จัดการข้อมูลเกี่ยวกับเซอร์ฟเวอร์และติดต่อกับฮาร์ดแวร์ของดีสเพลย์
Drawing	รูทีนที่ใช้วาดจุด, เส้น, รูปสี่เหลี่ยม, วงกลม ฯลฯ
Error	รูทีนที่ใช้ในการกำหนดฟังชั่นเรียกใช้(function call) เมื่อเกิดข้องผิดพลาด
Event	รูทีนที่รับค่าอินพุทจากยูสเซอร์, แอพพลิเคชัน, เซอร์ฟเวอร์
Extension	รูทีนที่ใช้ในการหาส่วนที่ว่างในเซอร์เวอร์และใส่ข้อมูลเกี่ยวกับการใช้
Font	รูทีนที่ใช้แสดงรายละเอียดของฟอนท์
Geometry	รูทีนที่ใช้ส่งผ่านค่าของเส้น, วงกลม, ฯลฯ
Grachics Context	รูทีนที่ใช้ส่งผ่านค่าของสิ่งที่ต้องการจะวาด
Host Acess	รูทีนที่ใช้ควบคุมการทำงานจากส่วนต่าง ๆ กับเน็ตเวอร์ก
Image	รูทีนที่รับค่า แสดงค่า หรือส่งค่าที่ต้องการจะให้ออกที่จอภาพ
Interclient - communication	รูทีนที่ใช้ในการส่งผ่านข้อมูลระหว่างไคลแอนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ต่อ)

ฟังก์ชั่น(Function)	คำอธิบาย
Keyboard	รูทีนที่ใช้ในการแปลงค่าอินพุทจากคีย์บอร์ด และเก็บค่าที่แมปแล้ว
Pointer	ฟังค์ชันที่ใช้จัดการและเปลี่ยนแปลงตัวชี้(pointer) ของอินพุท
Region	รูทีนที่ใช้คำนวณพื้นที่ที่ใช้ทำงาน
Resource management	รูทีนที่ใช้ในการจัดการเกี่ยวกับการใช้ทรัพยากรของผู้ใช้
Screen server	รูทีนที่ใช้ในการกำหนดการทำงานของช่องว่าง(blank) ว่าต้องการ จะให้มีช่วงห่างเท่าไร เมื่อมีการเคลื่อนเคอร์เซอร์ไปแต่ละครั้ง
Text	รูทีนที่ใช้ในการวาดอักษร และหาขนาดของตัวอักษรที่จะวาด
User preference	รูทีนที่ใช้กำหนด หรือรับคลิ๊กของคีย์บอร์ด และการกำหนดให้ทำงานแบบอัตโนมัติ
Window attribute	รูทีนที่ใช้กำหนดและรับค่า ลักษณะของวินโดว์
Window life	รูทีนที่ใช้สร้างและลบวินโดว์
Window management	รูทีนมีการอนุญาตให้วินโดว์เคลื่อนย้ายไปได้ทั่วจอ สามารถที่จะเปลี่ยนขนาด หรือการแสดงผลไว้ข้างล่างหรือข้างบนของวินโดว์อื่น

ตารางที่ 2.1 แสดงฟังก์ชั่นในเอ็กซ์ไลบรารี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

หลักการทำงานของเอกซ์(X concepts)

เมื่อโปรแกรมเมอร์ ศึกษาโปรแกรมใหม่ ๆ เขามักจะดูโปรแกรมตัวอย่างเพียงเล็กน้อย แล้วก็จะเริ่มเขียนโปรแกรมทันที โดยดูเอกสารอ้างอิง(reference) ประกอบ เมื่อต้องการรู้เรื่องเนื้อหาบางอย่าง หนังสือนี้จัดทำขึ้นในลักษณะที่เป็นทั้งตัวอย่างการใช้งาน(tutorial)และอ้างอิง(reference) ในหนังสือประกอบด้วยตัวอย่างโปรแกรมมากมาย เหมาะมากสำหรับผู้ที่ชอบดูโปรแกรมมากกว่าคำอธิบาย ในตัวอย่างโปรกแกรมนั้น เราจะพบประเด็นสำคัญมากมายที่เรียนรู้ต่อการเรียนรู้โปรแกรมและเขียนพัฒนาการโปรแกรมต่อไป

การดูเฉพาะตัวอย่างโปรแกรมนั้น มันทำให้สามารถเขียนโปรแกรมให้ทำงานได้โดยใช้ระยะเวลาอันสั้น แต่ในที่สุดเราจะพบว่าการที่จะเข้าใจถึงส่วนต่าง ๆ ของระบบ หรือเขียนโปรแกรมได้ทำงานที่ดีกว่าเดิมนั้น จำเป็นต้องมีความรู้ในหัวข้อต่าง ๆ อีกมาก ในระบบเอกซ์วินโดว์มีหลักการพื้นฐานหลายอย่างที่โปรแกรมเมอร์จะต้องรู้ ดังนั้นถ้าหากว่าได้ศึกษาหลักการต่าง ๆ อย่างลึกซึ้งแล้วก็คงจะไม่ต้องเสียเวลาในการแก้ไขข้อผิดพลาดของโปรแกรมที่จะเกิดตามมาภายหลัง

3.1 การทำงานของเอกซ์ไลบรารี่

เริ่มต้นกันด้วยปัญหาของระบบเอ็กซ์วินโดว์ที่เคยมี และวิธีการแก้ไขปัญหาดังต่อไปนี้

1. เอ็กซ์(X)ถูกออกแบบให้ทำงานเกี่ยวกับการสร้างวินโดว์บนบิทแมบเทอร์มินัล(bit maped terminal) ซึ่งจะพบปัญหาว่ามันไม่ง่ายเลยที่จะใช้ได้กับฮาร์ดแวร์(Hardware)ที่ต่าง ๆ กัน ก็ได้มีการแก้ไขให้สามารถทำงานได้แม้กับระบบปฏิบัติการที่ต่างกัน โดยที่ความสามารถต่าง ๆ ยังคงมีอยู่ครบสมบูรณ์

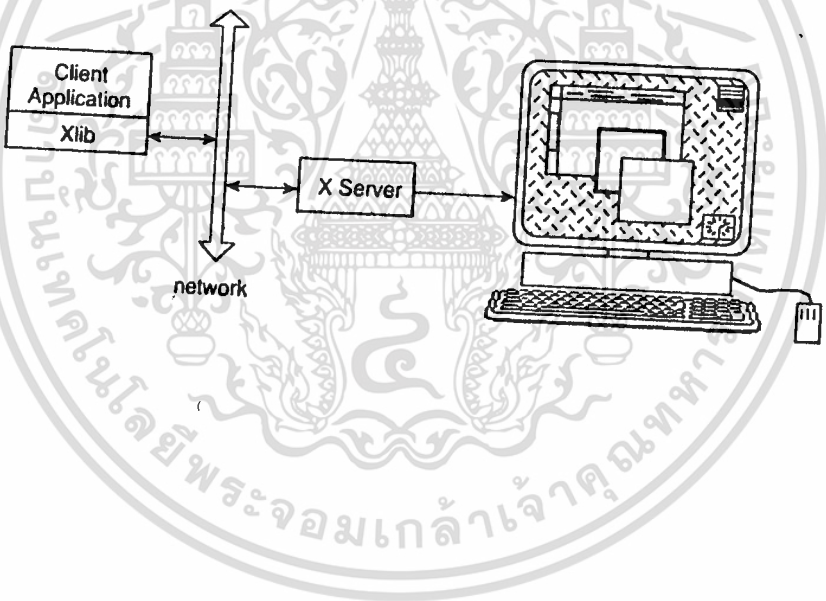
2. เนื่องจากเอ็กซ์ถูกออกแบบให้สามารถใช้ได้กับฮาร์ดแวร์ที่ต่าง ๆ กัน ที่ทำงานร่วมกันอยู่ในระบบเน็ตเวอร์ก ซึ่งก็จะพบปัญหาหลักที่ว่า มาตราฐานของเน็ตเวอร์กโปรโตคอล มีอยู่มากมายแตกต่างกันไป จึงได้มีความพยายามที่จะสร้างโปรโตคอล (protocol) ที่สามารถติดต่อภายในระบบเน็ตเวอร์กได้

3. ส่วนใหญ่บริษัทต่าง ๆ มักจะพัฒนาระบบที่เป็นลักษณะเฉพาะของตนเอง ดังนั้นจึงเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่แน่นอนว่าจะเกิดปัญหาที่จะใช้ร่วมกันไม่ได้ (incompatible)เมื่อนำไปใช้ในเครื่องต่าง ๆ กัน X ได้พยายามที่จะแก้ปัญหานี้ โดยจะทำให้ผู้ใช้ไม่รู้สึกขัดเมื่อมีการเปลี่ยนฮาร์ดแวร์

สำหรับการแก้ปัญหาเรื่องเน็ตเวอร์ก จะต้องมีโปรแกรมควบคุมอยู่ทุกจุดต่อของเน็ตเวอร์ก เพื่อคอยเป็นตัวส่งและรับสัญญาณข้อมูลและทำการแปลความหมายของข้อมูล ตัวที่คอยควบคุมดิสเพลย์และอุปกรณ์อินพุท(input devices)นี้เรียกว่าเซอร์ฟเวอร์ปลายอีกด้านหนึ่งของเน็ตเวอร์ก คือ ไคลแอนท์เป็นโปรแกรมที่เขียนโดยเอ็กซ์ไลบรารี่ใช้ติดต่อกับเอ็กซ์โปรโตคอล (X protocol) ดูรูปที่ 3.1



รูปที่ 3.1 แสดงไคลแอนท์ และเซอร์ฟเวอร์

ที่จริงแล้วภาษาที่สามารถรับคำขอร้อง (request) ของเอ็กซ์โปรโตคอลได้ก็สามารถติดต่อกับเซอร์ฟเวอร์และสามารถใช้กับระบบเอ็กซ์วินโดว์ได้ แต่ในระบบเอ็กซ์วินโดว์ปัจจุบันใช้ภาษาซี(C language) เพราะเป็นภาษาที่ใช้ได้อย่างกว้างขวาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 เอ็กซ์โปรโตคอล(X Protocol)

เอ็กซ์โปรโตคอลเป็นตัวที่บอกว่าอะไรเป็นตัวที่สร้างชุดของข้อมูล ซึ่งมีการส่งผ่านไปกับระหว่างตัว เซอร์ฟเวอร์และเอ็กซ์ไลบรารี่ ในกรณีที่ เซอร์ฟเวอร์และเอ็กซ์ลิปทำงานอยู่ในเครื่องเดียวกัน โปรโตคอลก็ช่วยในการสื่อสารภายใน channel เดียวกันไม่ใช่ลักษณะของระบบโครงข่ายภายนอก(external network) ชุดของข้อมูลที่มีการส่งผ่านโดยทางเอ็กซ์โปรโตคอลมี 4 แบบคือ

3.2.1 คำขอร้อง (Protocol Request)

เป็นการที่เอ็กซ์ไลบรารี่ติดต่อกับเซอร์ฟเวอร์ เพื่อขอร้องให้เซอร์ฟเวอร์กระทำบางสิ่งบางอย่างให้ เช่น การลากเส้นบนจอภาพ เปลี่ยนสีบนจอภาพ หรือต้องการทราบข้อมูลเกี่ยวกับขนาดของหน้าต่าง(ซึ่งเปลี่ยนแปลงได้) อันที่จริงแล้วแทบทุกคำสั่งในเอ็กซ์ไลบรารี่นั้นเป็นคำขอร้องทั้งสิ้น มีข้อยกเว้นอยู่เพียงไม่กี่คำสั่ง ทั้งนี้เพราะผู้บัญชาการที่แท้จริงคือ เซอร์ฟเวอร์ จะทำอะไรก็ต้องขึ้นตรงต่อเซอร์ฟเวอร์

3.2.2 คำตอบรับ(Protocol Replies)

คือการที่เซอร์ฟเวอร์ติดต่อกับมายังโปรแกรมของเราผ่านทาง เอ็กซ์ไลบรารี่เนื่องจากที่เราร้องขอบางอย่างไป การขอร้องนั้นไม่จำเป็นต้องมีการรับเสมอไป ปกติการตอบรับจะเกิดขึ้นเมื่อคำร้องขอนั้นเกี่ยวกับความต้องการทราบข้อมูล ตามตัวอย่างที่ผ่านมา การขอให้ลากเส้นบนจอภาพก็ดี การเปลี่ยนสีของจำภาพก็ดี ไม่ต้องการคำตอบกลับมาจากเซอร์ฟเวอร์แต่อย่างใด ในทางตรงกันข้ามกับการขอทราบขนาดหน้าต่างนั้น ต้องการคำตอบจากเซอร์ฟเวอร์ ดังนั้นจะเห็นได้ว่า การขอร้องที่รอการตอบรับจะใช้เวลา และต้องติดต่อผ่านทางเครือข่ายเพิ่มขึ้นเป็นสองเท่า ซึ่งอาจทำให้โปรแกรมของเราทำงานช้าพอสมควร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 การแจ้งเหตุ(Event)

การแจ้งเหตุ เป็นการติดต่อสื่อสารจากเซอร์ฟเวอร์มายังโปรแกรมของเรา(ผ่านเอ็กซ์ไลบรารี่) เกี่ยวกับการเปลี่ยนแปลงของทรัพยากรต่าง ๆ ที่เซอร์ฟเวอร์ดูแลอยู่ทางตรง(ผู้ใช้เลื่อนเมาส์ ฯลฯ) และทางอ้อม(โปรแกรมอื่นเปิดหน้าต่างมาบังหน้าต่างของเรา)ซึ่งเป็นข้อมูลที่เป็นประโยชน์ต่อโปรแกรมของเรา ทางเซอร์ฟเวอร์จะแจ้งเหตุเข้ามาเองโดยไม่ต้องมีการร้องขอ และข้อมูลของเหตุการณ์เหล่านี้จะถูกเก็บไว้ในคิว (ไม่ได้มีการทำงานทันทีเหมือนคำตอบรับ) ทำให้ต้องใช้แนวคิดในการจัดการกับเหตุการณ์ที่รับมา ต่างจากการเขียนโปรแกรมธรรมดาอยู่บ้าง

3.2.4 การแจ้งเหตุข้อผิดพลาด (Error)

เป็นการแจ้งเหตุประการหนึ่ง แต่เป็นการแจ้งเหตุข้อผิดพลาดที่เกิดขึ้น ดังนั้นจึงต้องมีการจัดการเพิ่มเติมกว่าปกติ โดยที่ในเอ็กซ์ไลบรารี่ จะมีส่วนโปรแกรมอยู่สองส่วน สำหรับการแจ้งเหตุข้อผิดพลาดนี้ ส่วนหนึ่งใช้สำหรับการกู้ หรือ แก้ไขผิดพลาดชนิดที่แก้ไขได้ อีกส่วนหนึ่งใช้ในกรณีที่ข้อผิดพลาดนั้นไม่สามารถแก้ได้แล้ว

3.3 ระบบคำร้องขอในเอ็กซ์ไลบรารี่(Buffering)

ปัญหาความไม่เคยชินอันหนึ่งที่จะเกิดขึ้นกับการเขียนโปรแกรมโดยใช้เอ็กซ์ไลบรารี่ คือ การที่เราร้องขอ (request) บางสิ่งบางอย่างแล้ว ไม่ได้ตามต้องการ เหตุเกิดขึ้นเพราะเอ็กซ์ไลบรารี่ นั้นจะเก็บคำร้องขอเอาไว้ในคิว แทนที่จะส่งคำร้องไปยังเซอร์ฟเวอร์ทันที ที่ทำเช่นนั้นก็ด้วยความหวังดีว่าโปรแกรมของเราจะได้ทำงานต่อไปโดยไม่ต้องรอติดต่อกับเซอร์ฟเวอร์ เพื่อยืนคำร้องขอทุก ๆ ครั้ง เพราะคำร้องขอทุกอย่างไม่ต้องการการตอบสนองอย่างเร่งด่วนจากเซอร์ฟเวอร์ ยิ่งถ้าเป็นการติดต่อผ่านระบบเครือข่ายยิ่งเป็นการดีที่จะรวบรวมคำร้องทั้งหลาย ๆ อันเข้าด้วยกันแล้วส่งไปทีเดียว เพื่อป้องกันความล่าช้า แต่บางครั้งความหวังดีอันนี้ ก็ส่งผลกระทบให้ผู้เขียนโปรแกรมต้องประสบปัญหาหนักเช่นกัน ตัวอย่างเช่น หากต้องการให้โปรแกรมส่งข้อความขึ้นจอภาพเพื่อถามผู้ใช้ให้กรอกข้อความบางอย่างกลับมา ปกติก็ทำโดยคำร้องให้เซอร์ฟเวอร์นำข้อความคำถามไปแสดงที่จอภาพ จากนั้นก็จะรอการแจ้งเหตุที่ให้ผู้ใช้ตอบกับมา มองดูก็ตรงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรงมาดี ไม่น่ามีอะไรผิดพลาด แต่ปรากฏว่าพอใช้งานโปรแกรมเข้าจริง ๆ มักจะมาหยุดอยู่ตรงนี้ เพราะคำร้องขอนั้นถูกนำไปเก็บไว้ในคิว

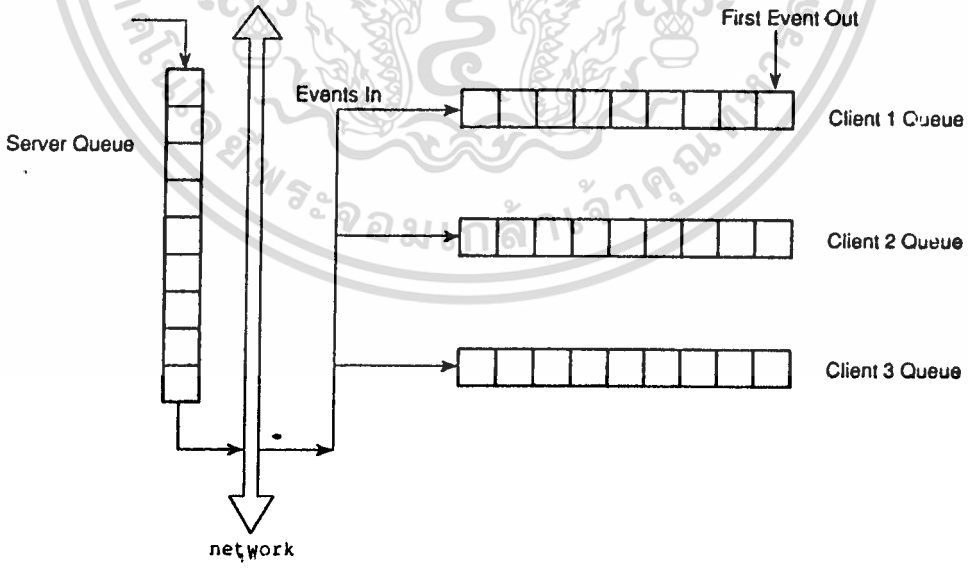
การที่คำขอร้องทั้งหลายในเอ็กซ์ไลบรารี่จะถูกส่งต่อไปให้แก่เซอร์ฟเวอร์นั้นมี 3 กรณีคือ

1. เมื่อโปรแกรมของเราต้องการรอการแจ้งเหตุเฉพาะอันใดอันหนึ่ง ในกรณีนี้โปรแกรมของเราจะว่างงาน และจะต้องรอเหตุการณ์ให้เกิดขึ้นอยู่แล้ว เอ็กซ์ไลบรารี่ จะจัดการส่งคำร้องต่าง ๆ ของเราไป เพื่อไม่ให้เราต้องรอ

2. จะเกิดขึ้นเมื่อเราทำด้วยคำร้องพิเศษ (มักเป็นการร้องขอข้อมูลบางอย่าง) ซึ่งจำเป็นจะต้องได้รับการตอบสนองจากเซอร์ฟเวอร์ทันที แทนที่จะส่งคำร้องเฉพาะคำร้องพิเศษ ก็จะใช้โอกาสนี้ส่งข้อมูลในคิวไปพร้อมกันทั้งหมดในครั้งเดียว จะเป็นการติดต่อกันครั้งเดียวแต่ได้งานหลายหลายอย่าง

3. การใช้คำสั่งเฉพาะในการบังคับให้ เอ็กซ์ไลบรารี่ จัดการยื่นคำร้องไปให้เซอร์ฟเวอร์ทันที โดยไม่ต้องรอให้เกิดกรณีแรก ๆ เสียก่อน ซึ่งคำสั่งบังคับอันนั้นทำให้ผู้เขียนโปรแกรมสามารถบังคับ เอ็กซ์ไลบรารี่ จัดการยื่นคำร้องขอตามจังหวะเวลาที่เราต้องการได้

นอกจากเอ็กซ์ไลบรารี่จะเก็บคำร้องขอของโปรแกรมไว้ในคิวแล้ว ยังมีการจัดคิวคำตอบรับหรือการแจ้งเหตุจากเซอร์ฟเวอร์มายังโปรแกรมด้วย



รูปที่ 3.2 แสดงการจัดคิวคำร้อง และคำตอบรับในระบบเอ็กซ์วินโดว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ทรัพยากร (resource)

เป็นเทคนิคในการลดความยุ่งยากของระบบ โดยการตั้งชื่อ (ID) ให้กับวินโดว์ หรือฟอนท์ต่าง ๆ คล้ายกับเป็นชื่อที่ใช้เรียกแทนสิ่ง ๆ นั้น และชื่อนี้จะถูกเรียกใช้โดยไคลแอนท์ แต่ละส่วนที่ถูกแทนด้วย ID เรียกว่าทรัพยากร(resource)ทรัพยากรสามารถเป็นได้ทั้งวินโดว์,พิกแมบ (pixmap),ตารางสี,เคอร์เซอร์,ฟอนท์ หรือ กราฟฟิคคอนเท็กซ์และ ID จะเป็นเลขประจำของทรัพยากรแต่ละตัว

เมื่อใดที่ต้องการใช้วินโดว์ หรือทรัพยากรอื่น ๆ เลข ID นี้จะถูกใช้เป็นอากิวเม้นต์ (argument) สำหรับรูทีนนั้น นั่นคือแทนที่เราจะต้องมาใส่ค่าโครงสร้าง ข้อมูลทรัพยากรนั้น ๆ เพื่อที่จะส่งข้ามเน็ตเวอร์กไป เราก็เพียงแต่ส่งค่าของ ID ก็สามารถใช้แทนวินโดว์นั้นได้แล้ว เพราะงานบางอย่างไม่ได้ต้องการใช้ข้อมูล แต่ถ้าเมื่อใดที่ต้องการข้อมูลของทรัพยากร ไคลแอนท์ก็จะส่งคำร้องไปให้เซอร์ฟเวอร์ เพื่อขอข้อมูลที่เกี่ยวข้องกับทรัพยากรนั้น ๆ

3.5 พรอบเพอตี้และอะตอม(property และ atoms)

พรอบเพอตี้ เป็น โปรแกรมสำเร็จรูป(package) ของข้อมูลที่เกี่ยวข้องกับวินโดว์เพื่อให้ไคลแอนท์สามารถทำงานได้ภายใต้เซอร์ฟเวอร์ ไคลแอนท์จะเป็นผู้ใช้ พรอบเพอตี้ เพื่อเก็บค่าข้อมูลที่ไคลแอนท์ต้องการทราบ และจะมีการอ่านข้อมูลนี้ไปเมื่อไคลแอนท์ตัวอื่นต้องการทราบข้อมูลนั้นเช่นกัน

อะตอม เป็น ID ที่ใช้แทน พรอบเพอตี้ อะตอมจะถูกใช้เพื่อใช้แทนค่า พรอบเพอตี้ ในรูทีนคอลล์(call routine)เพราะว่าไม่ต้องการที่จะส่งเป็นข้อมูลยาว ๆ ข้ามเน็ตเวอร์ก อะตอมของ พรอบเพอตี้ เทียบได้กับ ID ของทรัพยากร ต่างกันที่อะตอมจะแทน propertyได้ในวินโดว์เดียวเท่านั้น ถ้าอะตอมนี้อยู่ในวินโดว์อื่นก็จะแตกต่างกันออกไป

ประโยชน์ของ พรอบเพอตี้ คือใช้ในการติดต่อสื่อสารกันระหว่างแอพพลิเคชัน และวินโดแมนนาเจอร์ แอพพิเคชันจะเป็นผู้กำหนดมาตราฐานของพรอบเพอตี้ (standard property) ที่เรียกว่ามาตรฐานเพราะเป็นการกำหนดค่าน้อยที่สุดที่ยอมรับได้ของสิ่งที่ควรกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 วินโดว์แมนนาเจอร์ (window manager)

วินโดว์แมนนาเจอร์ เป็นไคลแอนท์ตัวหนึ่งที่เขียนด้วย เอ็กซ์ไลบรารี่ เพื่อความสะดวกดังนั้นจึงต้องมีความไวในการตอบสนอง เป็นส่วนที่คอยจัดการเกี่ยวกับการเปลี่ยนขนาดและย้ายตำแหน่งของวินโดว์ ไปยังตำแหน่งต่าง ๆ บนจอภาพ นอกจากนี้ยังจัดการเกี่ยวกับการสร้างวินโดว์ใหม่ด้วย ส่วนใหญ่การติดต่อระหว่างไคลแอนท์และวินโดว์แมนนเจอร์จะเกิดขึ้นโดยผ่านพรอบเพอตี้

ลักษณะเด่นของวินโดว์แมนนาเจอร์คือซับสตรัคเจอร์รีไดเรกชั่น(substructure redirection) ซึ่งเป็นการยอมให้แอพพิเคชันสามารถเปลี่ยนแปลงขนาด,ตำแหน่งของวินโดว์,ความกว้างของขอบ,สี,และลำดับของวินโดว์ที่จะแสดงออกทางจอภาพได้ นั่นคือแอพพลิเคชันจะต้องร้องขอไปยังเซอร์ฟเวอร์เพื่อบอกว่าต้องการที่จะเปลี่ยนค่ารูปแบบของระบบคอมพิวเตอร์(configuration) ก็จะมีการส่งเหตุการณ์ไปยังวินโดว์แมนนาเจอร์ เพื่อให้มาจัดการตามที่ได้ร้องขอไป

ข้อจำกัดของ ซับสตรัคเจอร์รีไดเรกชั่น คือ

1. แอพพลิเคชันไม่สามารถที่จะกำหนดค่าโครงสร้าง ให้เป็นแบบที่ยังไม่ได้กำหนดไว้ในวินโดว์
2. แอพพลิเคชันไม่สามารถที่จะเรียกสิ่งที่วาดลงไปมาดูได้ทันทีทันใด เพราะจะต้องรอให้วินโดว์ map เรียบร้อยเสียก่อน

3.7 วินโดว์คืออะไร

วินโดว์เป็นพื้นที่สี่เหลี่ยม ซึ่งสามารถทำงานได้หลายอย่างคล้ายกับจอภาพเล็ก ๆ วินโดว์บนจอสามารถที่จะวางซ้อนทับกันได้ และมีได้หลายขนาด แต่ละวินโดว์จะทำงานที่แตกต่างกันได้ และวินโดว์ที่กำลังใช้งานอยู่ก็จะปรากฏอยู่หน้าสุดของจอภาพ นอกจากนี้วินโดว์แมนนาเจอร์ก็จะยอมให้ยูสเซอร์ดึงวินโดว์ที่อยู่ข้างล่างขึ้นมาใช้งานได้ด้วย

3.8 ลักษณะของวินโดว์ (window characteristic)

ลักษณะของวินโดว์พอจะสรุปได้ดังต่อไปนี้

1. วินโดว์จะมีวินโดว์หลัก เรียกว่า พ่อ (parent) ซึ่งใช้ในการสร้างวินโดว์ต่อไป วินโดว์ที่สร้างขึ้นมาใหม่นี้จะเป็นลูก (child) ของวินโดว์ราก วินโดว์จะไม่สามารถแสดงผลในพื้นที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่อยู่นอกขอบเขตตัวมันเองได้ และก็ไม่สามารถรับค่าอินพุทหรือ pointer ได้ถ้า pointer นั้นอยู่นอกวินโดว์ วินโดว์ที่สร้างขึ้นมาตัวแรกเลย จะมีชื่อเรียกโดยเฉพาะว่าวินโดว์ราก (root window) ซึ่งจะเป็นวินโดว์เดียวที่ไม่มีพ่อ และลูกสร้างโดยเอ็กซ์เซอร์ฟเวอร์ (X server)

2. จุดเริ่มต้น(origin)ของวินโดว์จะอยู่ที่มุมบนซ้ายของวินโดว์ ซึ่งจะเป็นตำแหน่งที่อธิบายความสัมพันธ์ของตัวมันเองกับพ่อของมัน นอกจากนี้วินโดว์จะต้องมีความกว้าง,ความยาว,และขนาดของของวินโดว์ด้วย ดูรูปที่ 3.3

มีวินโดว์จำนวนมากที่มีพ่อเป็นวินโดว์อันเดียวกัน ดังนั้นจึงต้องมีลำดับสแตกเพื่อใช้ในการหาว่าวินโดว์อยู่ก่อนหรืออยู่หลัง ลักษณะ 4 อย่างของวินโดว์ข้างบนนี้เรียกว่าโครงสร้างของวินโดว์ (window configuration)

โครงสร้างของวินโดว์ จะเก็บค่าต่อไปนี้

-ความกว้าง,ความยาวของวินโดว์เป็นพิกเซล (Pixel) ไม่ใช่ค่าขอบของวินโดว์

-ความกว้างของขอบวินโดว์ซึ่งสามารถเปลี่ยนแปลงค่าได้ ถ้าค่านี้เป็นศูนย์ แสดงว่าวินโดว์นี้จะไม่มีขอบ

-ตำแหน่งต่าง ๆ ของวินโดว์ กำหนดเป็นค่า x,y ที่เก็บเป็นพิกเซล ค่านี้จะมีความสัมพันธ์กับจุดเริ่มต้นของวินโดว์ย่อย

-ลำดับสแตก(stack order) จะเก็บค่าเฉพาะ วินโดว์ที่อยู่ในระดับเดียวกันในโปรแกรมแอพพลิเคชัน ในการสร้างวินโดว์จะต้องมีการกำหนด ความกว้าง,ความยาว,ตำแหน่ง และความกว้างของขอบ

3. วินโดว์จะต้องมีเดบ(depth) และ วิชวลไทป์(visual type) ซึ่งเป็นการบอกลักษณะของสี

-เดบ คือ จำนวนบิทของแต่ละพิกเซลที่ต้องการแสดงสี

-วิชวลไทป์ คือวิธีการแปลงค่าของพิกเซล เพื่อหาว่าเป็นสีอะไร แบบไหน

4. วินโดว์มี คลาส(class) 2 แบบคือ

-แบบเอาท์พุทอินพุท(output input) คือจะรับค่า อินพุทและแสดงค่าออกทางเอาท์พุท

-แบบอินพุททางเดียว(inputonly) คือจะรับค่า อินพุทอย่างเดียวไม่มีเอาท์พุท ออกมา

5. วินโดว์จะต้องมีลักษณะเฉพาะ (attribute) ซึ่งวินโดว์ลักษณะเฉพาะ (Window

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

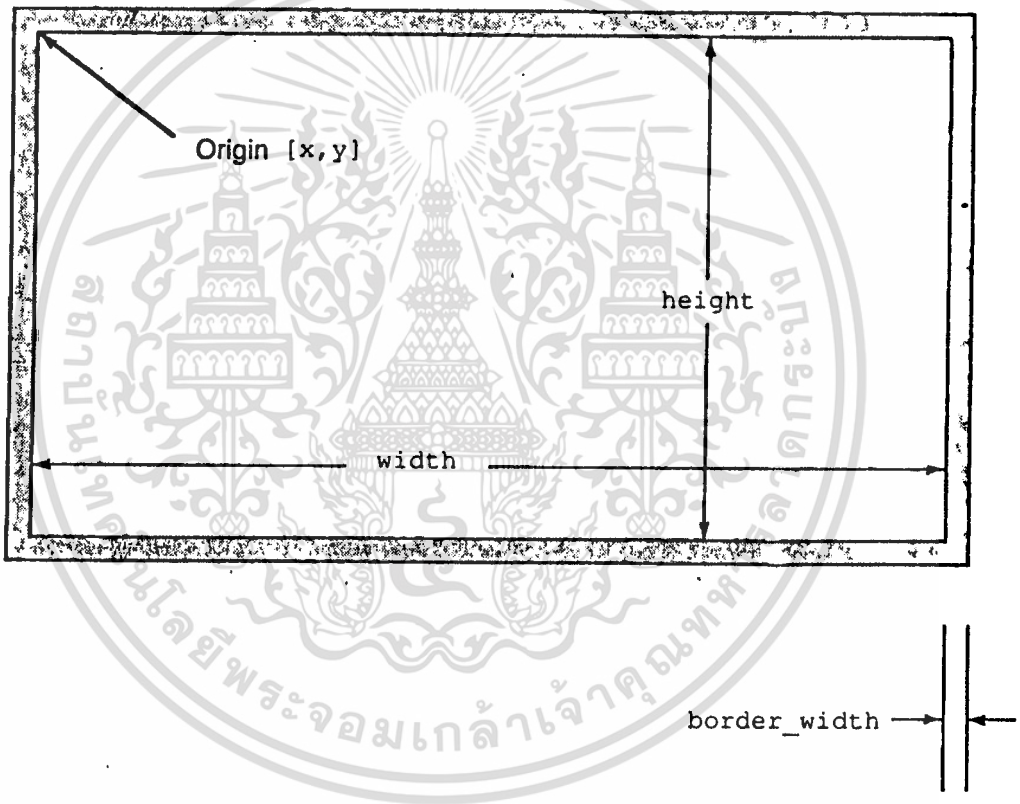
attribute) จะเป็นตัวควบคุมการแสดงผล และการตอบสนองของวินโดว์เช่น

-สีของขอบและสีพื้น (backgruond)

-เคอร์เซอร์ควรจะอยู่ที่ไหน

-เมื่อมีการสร้างหรือย้ายหรือเปลี่ยนแปลงขนาดของวินโดว์ จะมีการเก็บข้อมูลอย่างไร

-ตารางสีอันใดที่ใช้



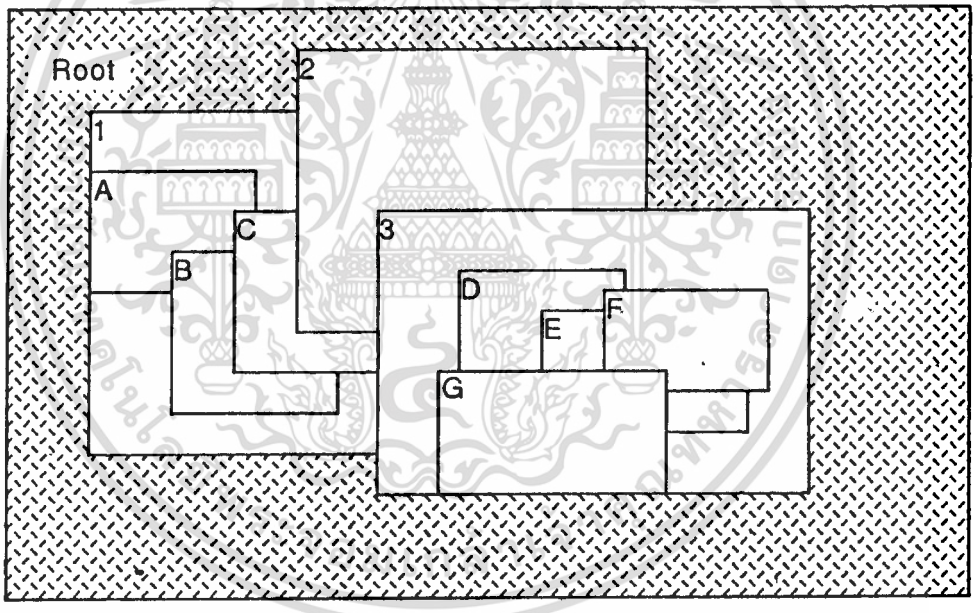
รูปที่ 3.3 แสดงโครงสร้างทั่วไปของวินโดว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.0 แผนภูมิต้นไม้แบบลำดับชั้น(Tree Hierachy)

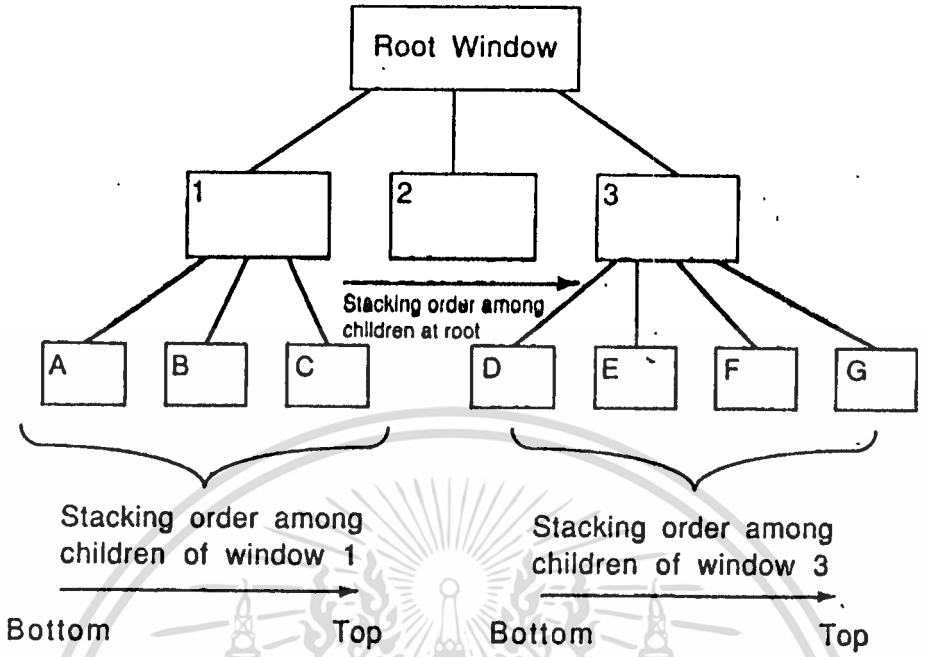
ลำดับชั้น(Hierachy) ของวินโดว์มีลักษณะเป็นแบบต้นไม้ ส่วนบนสุดจะเป็นวินโดว์ราก ซึ่งจะมีพื้นที่เต็มหน้าจอ วินโดว์แรกจะถูกสร้างด้วยไคลแอนท์ แต่ละอันจะเป็นลูกของวินโดว์ราก ซึ่งสร้างโดยใช้รูทีน XCreateWindow หรือ XCreateSimpleWindow นอกจากนี้ วินโดว์ที่สร้างขึ้นมาแต่ละตัวก็จะมีลูกเป็นตัวของมันอีก ถ้ามีการสร้างวินโดว์ขึ้นภายใต้ตัวของมันเอง

ตำแหน่งของวินโดว์ลูกสามารถที่จะอยู่นอกขอบเขตของวินโดว์พ่อได้ แต่เอาท์พุทที่ต้องการจะแสดงผล และการรับอินพุทของลูก จะทำได้เฉพาะพื้นที่บริเวณที่ลูกและพ่อซ้อนทับกันเท่านั้น จากรูปที่ 3.4 แสดงว่าลูกของวินโดว์ 1,2 และ 3 ไม่ได้อยู่นอกขอบเขตของพ่อเลย



รูปที่ 3.4 แสดง Window heirachy ที่ปรากฏบนจอภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดงบล็อกไดอะแกรมของ Window heirachy

ศัพท์แต่ละคำมีความหมายดังต่อไปนี้คือ

-พ่อ(parent) วินโดว์ที่ใช้สร้างลูก

-ลูก(child) วินโดว์ที่สร้างมาจากวินโดว์อื่น

-วินโดว์ย่อย(subwindow) คล้ายกับ ลูก

-ซิบลิง (sibling) วินโดว์ที่สร้างมาจากพ่อเดียวกัน (พี่และน้อง)

-เดสเซนแดนซ์ (descendants) มองจากวินโดว์ที่เราอยู่ วินโดว์ที่อยู่ล่างจะถือว่าเป็นเดสเซนแดนซ์ทั้งหมด

-แอนเซสเตอร์ (ancestors) มองจากวินโดว์ที่เราอยู่ วินโดว์ที่อยู่ข้างบนของเราจะถือว่าเป็นแอนเซสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10 ระบบพิกัด (coordinate system)

ในระบบ X Window

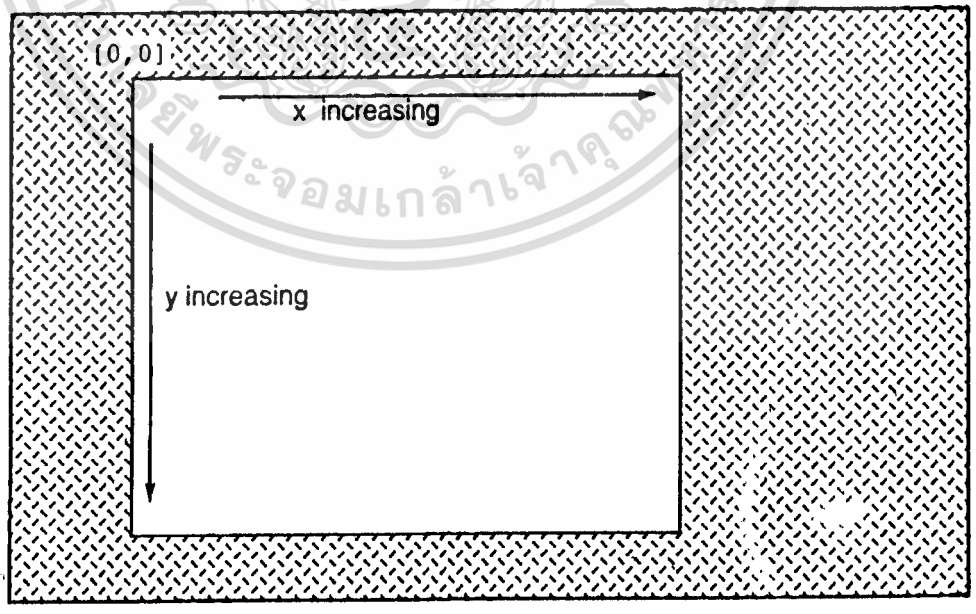
-แกนนอนคือแกน x และแกนตั้งคือแกน y

-ถ้าค่าแกน x และy เป็น 0 คือจุดที่อยู่มุมบนซ้าย (ขอบใน) ของวินโดว์ที่ใช้งานอยู่ จุดนี้ก็คือจุดเริ่มต้นนั่นเอง เมื่อเราสร้างวินโดว์จะต้องมีการกำหนดขนาดของวินโดว์ เป็นโคออร์ดิเนต x,y โดยแต่ละวินโดว์จะมี ID เป็นหมายเลขประจำ ทุกรูทีนมีความเกี่ยวข้องกับวินโดว์จะใช้หมายเลข ID เป็นอาร์กิวเม้นต์ในการทำงานกับวินโดว์

3.11 ลำดับการเก็บวินโดว์ในสแตก (Window Stacking order)

เมื่อวินโดว์มีการซ้อนทับกันลำดับสแตก(stack order)จะเป็นส่วนที่จะหาว่าวินโดว์ไหนที่จะต้องปรากฏอยู่บนหน้าสุดของจะภาพ ลำดับของวินโดว์นั้นสามารถที่จะเปลี่ยนได้โดยใช้รูทีนเรียนขึ้นมา รูทีนนี้จะใช้ได้เฉพาะกับชิบลิง และเดสเซนเดนซ์เท่านั้น สำหรับแอนเซสเตอร์ใช้ไม่ได้

ตามปรกติลูกจะปรากฏอยู่หน้าพ่อเสมอ ดังนั้นเมื่อพ่อมีการเคลื่อนย้ายตำแหน่ง ลูกก็จะเคลื่อนย้ายตามไปด้วย



รูปที่ 3.6 แสดงระบบพิกัดของเอ็กซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.12 แมบบิ้งและการเห็น(Mapping และ Visibility)

วินโดว์ที่สร้างขึ้นมาใหม่ ๆ จะไม่สามารถปรากฏบนจอภาพได้ทันทีทันใด เพราะจะมีการแมบก่อน การแมบบิ้งวินโดว์(mapping window) คือการเพิ่มวินโดว์เข้าไปในสแตกออร์เดอร์ที่เป็นชิบลิงเดียวกัน และแสดงผลออกทางจอภาพ

XMapWindow เป็นฟังก์ชันที่ใช้สำหรับแมบวินโดว์จากตำแหน่งปัจจุบัน เข้าไปในลำดับสแตก และ XMapRaised ใช้สำหรับการวางวินโดว์ที่บนสุดของลำดับสแตกของชิลลิงของตัวมันก่อนที่จะแมบมัน

องค์ประกอบที่มีผลต่อวินโดว์คือ

1. วินโดว์จะต้องแมบด้วย XMapWindow หรือ รูทีนที่เกี่ยวข้อง
2. แอนเซสเตอร์ของทุกวินโดว์จะต้องถูก map ก่อน
3. วินโดว์จะต้องไม่ถูกบังด้วยชิบลิง หรือแอนเซสเตอร์ ถ้าถูกบังคังก็สามารถแก้ไขได้โดยไปแก้ที่ลำดับสแตก โดยใช้รูทีน XConfigureWindow หรือ XRestackWindows
4. เอาท์พุทบัฟเฟอร์(Output buffer)จะต้องถูกกระตุ้นด้วย ฟังก์ชัน XFlush หรือฟังก์ชันที่ร้องขอข้อมูลจากเซอร์เวอร์
5. เมื่อสร้างวินโดว์ขึ้นมาใหม่ วินโดว์จะยังไม่ปรากฏจนกว่าไคลแอนท์จะได้รับเหตุการณ์ที่บอกว่าวินโดว์ได้แมบแล้ว หลังจากนั้นวินโดว์จะปรากฏบนจอภาพ

เมื่อต้องการที่จะลบวินโดว์ และเคสเซนเดนซ์ จะใช้รูทีน XUnmapWindow ซึ่งจะทำให้วินโดว์ไมแมบแต่ค่าโครงสร้างและลักษณะเฉพาะจะยังเก็บเอาไว้ เพียงแต่เอ็กซ์เซอร์ฟเวอร์ไมได้นำข้อมูลนั้นออกมาที่จอภาพเท่านั้น

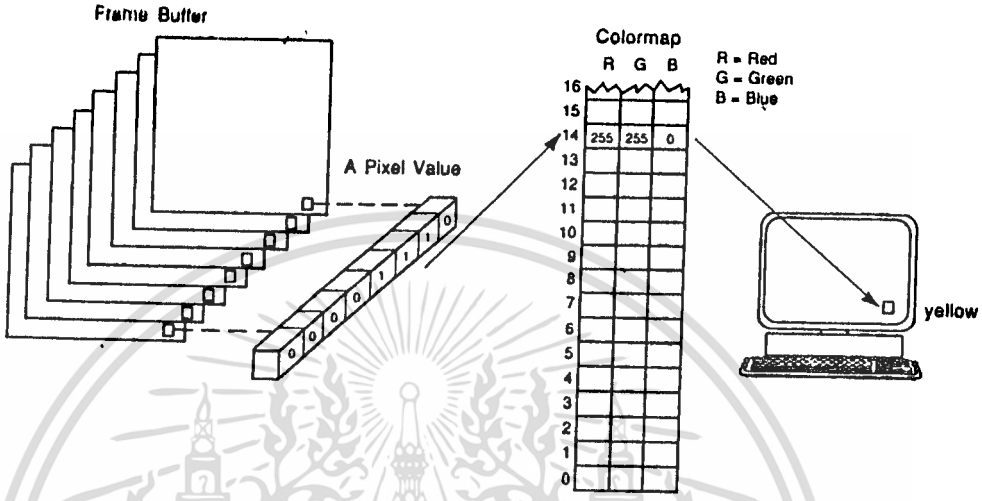
3.13 กราฟฟิกของเอกซ์ (X Graphics)

3.13.1 พิกเซล (Pixel)และสี (color)

ในระบบเอ็กซ์วินโดว์นั้น ได้ถูกออกแบบมาให้กับบิทแมบกราฟฟิคดีสเพลย์(bitmapped graphics display)โดยรูปแบบที่พบบ่อยที่สุดคือ ดิสเพลย์ที่แสดงผลเป็นจุดซึ่งมีความสว่าง 2 ระดับคือ ขาว และ ดำ ซึ่งก็คือสว่างและมืด ดิสเพลย์แบบนี้ใช้ 1 บิทต่อ 1 พิกเซล สถานะของแต่ละบิทจะเป็นตัวบอกว่าพิกเซลนั้นเป็นสีขาวหรือสีดำ ในระบบที่เป็นสี หรือระบบการแสดงผลแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมโนโครมที่แสดงผลเป็นระดับเกรย์สเกล (gray scale) ได้ จะมีการใช้ข้อมูลหลาย ๆ บิทต่อ 1 พิกเซล



รูปที่ 3.7 แสดงการแมบค่าพิกเซลไปเป็นตารางสี

สถานะของบิทต่าง ๆ ที่ประกอบกันนั้น ไม่ได้เป็นตัวควบคุมให้เกิดสีในพิกเซลต่อพิกเซลโดยตรง แต่จะเป็นตัวชี้ตารางสี (colormap)ดังรูปที่ 10 ในระบบจอสีนั้น 1 พิกเซลจะประกอบไปด้วยแม่สีของแสงทั้งสามคือ แดง เขียว และน้ำเงิน ซึ่งจะมาจากลำอิเลคตรอนทั้งสามลำ ความเข้มจของแต่ละแม่สีประกอบกันทำให้ตาเรามองเห็นสีเป็น 1 สี ในตารางประกอบไปด้วยอาเรย์ของแม่สีทั้งสาม (RGB) ตัวอย่างการอ้างสีในตารางสี เช่น ถ้าบิทต่าง ๆ ประกอบกันเป็น 00001110(14)ค่าอาร์จีบี(RGB value) อันดับที่ 14 ของตารางสีจะถูกนำไปแสดงที่พิกเซลตำแหน่งซึ่ง data bit เหล่านั้นกำหนด

สมาชิกแต่ละตัวของตารางสี เรียกว่าคัลเลอร์เซล (colorcell) ซึ่งจะผ่านค่าจากพิกเซล (pixel value) ไปเป็นค่าของสีแต่ละสี bitmapped ไดลแอนท์ต่าง ๆ จะสามารถใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางสีเดียวกันได้ โดยการอ้างชื่อของคัลเลอร์เซลที่ต้องการ ถ้าไคลแอนท์ต้องการที่จะใช้สีพิเศษเกินกว่าที่มีอยู่ในตารางสี ก็สามารถที่จะสร้างตารางสีจำลองได้ (writeable colormap) แต่ละวินโดว์ยังสามารถใช้ตารางสีต่าง ๆ กันได้ เป็นการชี้ให้เห็นว่าตารางสีนั้น ก็เป็นอีกองค์ประหนึ่งลักษณะเฉพาะของวินโดว์

3.13.2 พิกเซล (pixel)และ แพลน (plane)

จำนวนบิทต่อ 1 พิกเซลเป็นตัวบอกว่ากราฟฟิคดีสเพลย์(graphic display) นั้นมีจำนวนเพลนเท่าไรด้วยจอภาพขาวดำ จะมีเพียง 1 แพลน จอภาพสีที่มี 4-28 แพลน จอเกรย์สเกลมีได้ 2 ถึง 4 แพลน เอ็กซ์เวอร์ชั่นที่สิบเอ็ด(X11) สามารถ สนับสนุน ได้ถึง 32 แพลน

จากที่ได้กล่าวมาแล้วว่าบิทต่าง ๆ ของแต่ละพิกเซลจะเป็นตัวชี้ไปยังตารางสี พบว่าจำนวนสีทั้งหมดที่จะมีได้ 2^n โดยที่ n เป็นจำนวน แพลน ใน ดีสเพลย์ จะเห็นว่าในดิสเพลย์ที่มีจำนวนแพลนน้อยจะมีสีที่แสดงได้น้อย ซึ่งอาจจะต้องมีการสร้างค่าของ RGB เองแล้วเขียนเพิ่มลงในตารางสีที่มีอยู่

การคำนวณเกี่ยวกับกราฟฟิคจะทำบนค่าของพิกเซลก่อนที่จะถูกส่งผ่านไปยังค่าอาร์จีบีสอร์สพิกเซล (RGB value source pixel)ซึ่งกำหนดโดยการร้องขอ และค่าของพิกเซลเดิมจะรวมกัน โดยวิธีการรวมจะสอดคล้องกับแพลนมาสก์(plane mask),คลิบมาสก์(clip mask), ฟังก์ชั่นทางตรรก(logical function) เพื่อให้ได้ค่าพิกเซลปลายทางสุดท้าย(final destination pixel value) ค่า แพลนมาสก์,คลิบมาสก์,ฟังก์ชั่นทางตรรกเป็นส่วนหนึ่งของโครงสร้างข้อมูลที่เรียกว่ากราฟฟิคคอนเทกซ์(graphic context หรือ GC) ซึ่งจะได้อธิบายในตอนต่อไป

มาโคร (macro) ที่ชื่อแบลคพิกเซล(BlackPixel) และไวท์พิกเซล(WhitePixel) จะส่งค่าของพิกเซลซึ่งแมปไปยังจุดขาวและจุดดำ โดยใช้ตารางสีของจอภาพนั้น ๆ มาโครส่วนใหญ่จะใช้ในจอโมโนโครม แต่ก็อาจจะใช้ในการทำสีกรอบหรือพื้นของจอสีได้ ในระบบสีนั้น สีขาวและสีดำอาจจะกำหนดได้โดยผู้ใช้ ซึ่งก็ขึ้นอยู่กับตารางสีด้วยว่ายอมให้เขียนทับหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.13.2 พิกแมบและดรอเอเบิล(Pixmaps และ Drawables)

การวาด draw นั้น ไม่จำเป็นต้องทำบนวินโดว์เสมอไป เราสามารถวาด ได้ด้วยวินโดว์และพิกเซลรวมเรียกว่าดรอเอเบิล

พิดแมบ (Pixamp) คือ อวเรย์ของค่าพิกเซลซึ่งมีเดบเหมือนวินโดว์ แต่ไม่มีตำแหน่งซึ่งสัมพันธ์กับวินโดว์ หรือพิกแมบและไม่มีลักษณะเฉพาะสิ่งเหล่านี้จะมีผลต่อพิกแมบเมื่อพิกแมบ ถูกวางบนวินโดว์แล้ว

รูทีนบางอย่างทำงานบนเฉพาะวินโดว์หรือเฉพาะบนพิกแมบโดยใช้วินโดว์หรือพิกแมบเป็นอาร์กิวเม้นต์ ในลักษณะนี้อาร์กิวเม้นต์ของรูทีนเอ็กซ์ไลบรารี่จะถูกำหนดเป็นดรอเอเบิล

ทุก ๆดรอเอเบิลรูทีน(drawing routine) จะต้องมีการกำหนดเป็นดรอเอเบิล

พิดแมบจะไม่ได้รับผลกระทบใด ๆ แม้ว่าจะมีวินโดว์มาทับ ส่วนวินโดว์นั้นจะวาดได้ขณะที่วินโดว์ประกฏอยู่เท่านั้น เพราะเมื่อวินโดว์ถูกบัง หรือยังไม่ได้ถูกแมบไป สิ่งที่วาดจะเสียไป (กรณีที่ไม่ได้มีการจัดเก็บในเนื้อที่สำหรับเก็บ(backing store)โดยเฉพาะ)

พิกแมบจะเก็บในหน่วยความจำ แต่ก็ยังปรากฏจนกว่าจะถูกคัดลอก(copy) ลงบนวินโดว์ที่ปรากฏอยู่ การคัดลอกพิกแมบ(copy pixmap) ใด ๆ ไปยังวินโดว์ พิดแมบนั้นจะต้องมีเดบเท่ากับเดบของวินโดว์นั้น เมื่อ คัดลอกไปแล้วตารางสีของวินโดว์นั้น จะทำการแปลงค่าของพิกเซลของพิกแมบให้เป็นสีที่เรามองเห็นได้ หลังจากคัดลอกแล้วการกระทำใด ๆ บน pixmap จะไม่มีผลกับวินโดว์ที่พิกแมบว่างอยู่

วินโดว์เองก็มีข้อบกพร่องคือ เราจะวาดอะไรบนวินโดว์ไม่ได้ถ้าวินโดว์นั้นไม่ปรากฏอยู่ (visible) แต่พิกแมบซึ่งเป็นตัวแทนของพื้นที่บนจอภาพขึ้นอยู่กับหน่วยความจำ ดังนั้นจึงสามารถวาดได้ตลอดเวลา แต่อย่างไรก็ตามข้อเสียของพิกแมบก็คือ เราจะมองไม่เห็นพิกแมบถ้าไม่ถูกคัดลอกไปบนวินโดว์เสียก่อน และการคัดลอกก็จะมีผลเรื่องประสิทธิภาพการทำงาน(performance) เนื่องจากต้องใช้หน่วยความจำซึ่งมีอยู่อย่างจำกัด

พิกแมบที่มีเดบเท่ากับ 1 เราเรียกว่าบิทแมบ(bitmap) เป็นอาเรย์ 2 มิติของบิทซึ่งนำไปใช้ได้หลายกรณี เช่น ใช่ร่วมกับการกำหนดเคอร์เซอร์,ฟอนท์ เป็นต้น แต่ละบิทแทนแต่ละค่าของพิกเซลซึ่งเป็นไปได้ 2 ค่าคือ 1 และ 0 หรือขาวและดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.13.4 การวาดและองค์ประกอบกราฟฟิค(Drawing และ Graphics context)

ในการเขียนโปรแกรมกราฟฟิกนั้นเอ็กซ์ได้จัดเตรียมรูทีนสำหรับวาด จุด, เส้นตรง, สี่เหลี่ยม, รูปหลายเหลี่ยม, เส้นโค้ง, ตัวหนังสือและอื่น ๆ รูทีนเหล่านี้โดยทั่วไปเรียกว่ากราฟฟิคพื้นฐาน(graphics primitive) แต่ในเอ็กซ์เรียกว่าองค์ประกอบกราฟฟิค(Graphics context หรือ GC) ซึ่งเป็นตัวแปรและข้อมูลอื่น ๆ ที่จำเป็นเช่น ขนาดของสี, เส้น และลักษณะการระบาย (fill) GC จะเป็นอาร์กิวเม้นต์ของรูทีนวาด และเปลี่ยนแปลงลักษณะที่เราเห็นบนดรอเอเบิล ไคลแอนท์ทำการสร้าง GC ก่อนที่จะมีการวาดเสมอ GC ที่สร้างขึ้นมาจะถูกนำไปเก็บไว้ในเซอร์ฟเวอร์ ดังนั้นเราจึงไม่ต้องส่งข้อมูลไปให้ทุกครั้งที่จะวาด เป็นการลดความคับคั่งระหว่างเอ็กซ์ไลบรารี่กับเซอร์ฟเวอร์ ทุกอย่างที่กำหนดไว้ใน GC จะถูกนำไปใช้กับการวาดทุกอย่างที่มีการเรียก GC นั้น

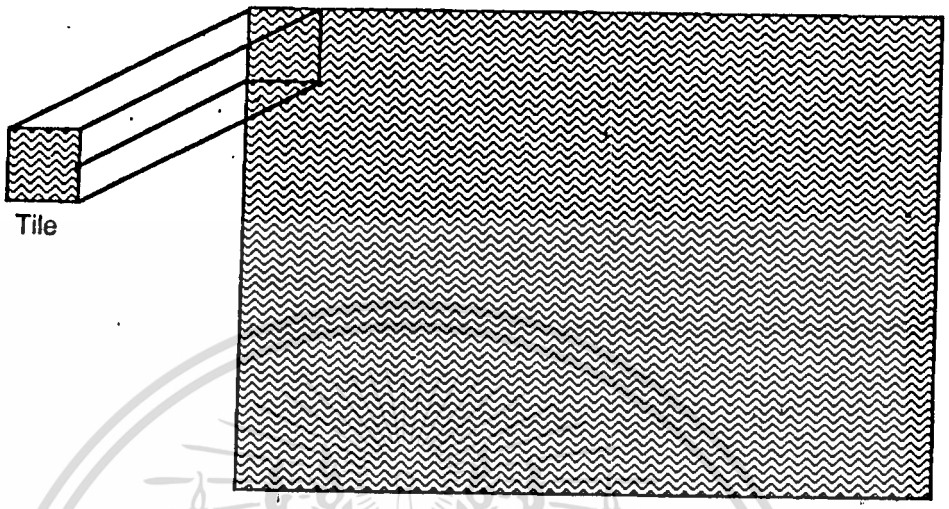
3.13.5 ไทล์และสติปเปิ้ล(Tile และ Stipple)

พิกแมบ ที่ใช้ในการระบายบนพื้นที่หนึ่ง ๆ เช่นเป็นสีพื้น(background) ของวินโดว์หรือในคำขอร้องต่อการวาด(drawing request) เราเรียกไทล์ (Tiles)หรือสติปเปิล(stripples)

ไทล์เป็น pixmap ซึ่งมีเดบเท่ากับเดบของดรอเอเบิลที่มันระบาย โดยทั่วไปไทล์มีขนาด 16x16 พิกเซล แต่อาจมีขนาดอื่นได้ซึ่งขึ้นอยู่กับฮาร์ดแวร์ ไทล์นี้สร้างได้ไม่ยาก เนื่องจากมีค่าของพิกเซลได้ 2 ค่า คือ ขาวกับดำ แต่ก็มีค่าของพิกเซลหลายค่าได้ พื้นที่ซึ่งถูกวาดไปแล้วด้วยรูทีนอื่นสามารถใช้ไทล์ระบายอีกได้ สีพื้นและขอบ(border) ของวินโดว์ก็สามารถใช้ไทล์ระบายได้เช่นกัน โดยกำหนดไว้ใน องค์ประกอบกราฟฟิคของลักษณะวินโดว์(window attribute GC)ไม่มีผลต่อไทล์พวกนี้

สติปเปิลเป็นพิกแมบที่มีเดบเท่ากับ 1 สติปเปิลบางครั้งใช้ร่วมกับฟอร์กราวด์(foreground) ค่าพิกเซล(pixel value) หรือค่าพิกเซลสีพื้น(background pixel value) ในการระบายพื้นที่ต่าง ๆ คล้ายกับไทล์ สติปเปิลที่ใช้ใน GC มี 2 แบบ คือ แบบแรก บิทที่ถูกกำหนดในสติปเปิลจะวาดด้วยสีฟอร์กราวด์ ส่วนบิทที่ไม่ถูกกำหนดจะวาดด้วยสีพื้นอีกแบบหนึ่ง บิทที่ถูกกำหนดในสติปเปิลจะวาดด้วยสีฟอร์กราวด์ นอกนั้นไม่เปลี่ยนแปลง สติปเปิล
ใช้ระบายเฉพาะพิกเซลที่กำหนดใน GCเท่านั้น และไม่สามารถใช้เป็นสีพื้นของวินโดว์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงให้เห็นการใช้ไทล์ระบายลงเป็นสีพื้นของวินโดว์

3.13.6 ลักษณะอื่น ๆ ของวินโดว์

ในส่วนนี้จะขยายความเกี่ยวกับเรื่องลักษณะ(characteristics) ของวินโดว์และอธิบายรายละเอียดเกี่ยวกับลักษณะเฉพาะของวินโดว์ (window attribute),โครงสร้าง(configuration),คลาส(class),เดล(depth) และ วิชวล(visual)

3.14 ลักษณะเฉพาะของวินโดว์(Window Attribute)

วินโดว์ลักษณะเฉพาะประกอบด้วยข้อมูลเกี่ยวกับวินโดว์ว่าจะให้เห็นในรูปแบบใด และทำอะไรบ้างแต่ละวินโดว์จะมีลักษณะเฉพาะของตัวมันเอง ซึ่งสามารถกำหนดได้ด้วยรูทีน XSetWindowAttributes หรืออาจใช้รูทีนซึ่งเปลี่ยนแปลงเฉพาะลักษณะเฉพาะบางส่วน ลักษณะเฉพาะจะควบคุมลักษณะต่าง ๆ ดังต่อไปนี้

สีพื้นอาจเป็นลักษณะของสี ๆ เดียวเป็นรูปแบบโปร่งใส(pattern transparent) ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอบ (Border) อาจเป็นลักษณะของสีหรือรูปแบบ(pattern)

บิทกาวิตี้ (Bit Gravity) เกี่ยวกับเรื่องความพอใจ(content) ต่าง ๆ ในวินโดว์เมื่อมีการเปลี่ยนแปลงขนาดของวินโดว์

วินโดว์กาวิตี้(Window gravity) เกี่ยวกับเรื่องลูกของวินโดว์ว่าจะมีการเปลี่ยนแปลงตำแหน่งอย่างไร ถ้าวินโดว์หลักมีการเปลี่ยนแปลงขนาด

แบคกิ้ง (Backing)เกี่ยวกับการเก็บข้อมูลและรายละเอียดต่าง ๆ ของวินโดว์ เช่นดีสเพลย์แพลน(display plane)ตัวใดจะเก็บค่าของพิกเซลไว้ ค่าใดจะถูกเก็บมาใช้เมื่อมีการสำรองแพลน(restore plane) ที่ไม่ได้มีการเก็บค่าไว้ เซอร์ฟเวอร์บางตัวอาจจะไม่ทำการแบคกิ้งได้ เราอาจดูได้ว่า เซอร์ฟเวอร์ของเราทำได้หรือไม่ โดยดูค่าที่รีเทอร์น(return) มาจาก Does-BackingStore

เซฟวิ้งอานเดอร์(Saving Under) บอกให้เรารู้ว่า เราสามารถเก็บค่าวินโดว์ที่ถูกบังได้หรือไม่ โดยที่เมื่อวินโดว์ที่บังอยู่นั้นได้ถูกเคลื่อนย้ายออกไปแล้ว จะต้องสามารถวาดวินโดว์ที่ถูกบังอยู่นั้นให้ขึ้นมาได้ดังเดิมการที่จะดูว่าเซอร์ฟเวอร์ตัวใดสามารถจัดการเก็บภายใต้วินโดว์(save under window)ได้ เราดูจากค่าที่ส่งมา เมื่อเรียกมาโคร ที่ชื่อ DoesSaveUnders

เหตุการณ์(Events) บอกให้เรารู้ว่าขณะนั้นสามารถรับเหตุการณ์ใดได้บ้าง และรับเหตุการณ์ใดไม่ได้บ้าง

Substructure Redirect Override บอกว่าวินโดว์นี้สามารถวางบนจอภาพ โดยไม่ถูกควบคุมจากวินโดว์แมนนาเจอร์ มักพบในพวกเมนู(menu) หรือวินโดว์ที่ใช้งานอยู่บ่อย ๆ และปิดไปโดยเร็ว

ตารางสี(colormap) บอกว่าตารางสีใดถูกนำมาใช้ในวินโดว์นั้น ๆ

เคอร์เซอร์ (Cursor) บอกว่าเคอร์เซอร์แบบใดจะถูกแสดงเมื่อ pointer อยู่ในวินโดว์นั้น

ความจริงลักษณะเฉพาะของวินโดว์อาจจะไม่มีผลบางอย่างจากระบบก็ได้ เช่น ไม่ได้กับขนาดหรือตำแหน่งของวินโดว์,พ่อ,ขอบ,ความกว้าง,เคบ,คลาส,วิชวล หรือค่าของร้องกราฟฟิคซึ่งเป็นงานของ GC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.15 โครงสร้างของวินโดว์(Window configuration)

โครงสร้างของวินโดว์จะประกอบไปด้วยตำแหน่ง , ความกว้าง ,ความสูง,ความหนาของเส้นขอบ,ตำแหน่งของสแตก การจัดการองค์ประกอบเหล่านี้จัดแยกออกจากลักษณะเฉพาะของวินโดว์ แม้ว่าจะเก็บใน โครงสร้าง (structure) ของ XWindowAttributes เนื่องจากการเปลี่ยนโครงสร้างของวินโดว์สามารถเปลี่ยนโครงสร้างของวินโดว์ ดังนั้นจึงต้องเปลี่ยนโดยความช่วยเหลือของวินโดว์แมนนาเจอร์

3.16 คลาสของวินโดว์แบบอินพุทเอาท์พุทและอินพุททางเดียว

ระบบเอ็กซ์วินโดว์เตรียมคลาสสำหรับวินโดว์ 2 คลาสคืออินพุทเอาท์พุทและอินทพุททางเดียว ข้อแตกต่างกันที่เห็นก็คือ อินพุททางเดียวนั้น วินโดว์ไม่สามารถใช้เป็นดรอเอเบิลได้(คือ ตัวรับ ค่าขอร้องกราฟฟิคต่างๆ) ดังนั้นวินโดว์แบบอินพุททางเดียว จึงมีข้อจำกัดเกี่ยวกับลักษณะเฉพาะ มากกว่าเช่น ไม่มีขอบวินโดว์และสีพื้นที่โปร่งใส (transparent background) และไม่สามารถมีลูก ซึ่งเป็นวินโดว์แบบอินพุททางเดียวได้

วินโดว์แบบอินพุททางเดียวทำให้เกิดพื้นที่ที่มองไม่เห็นบนจอ ซึ่งอินพุทก็จะมีได้ต่าง ๆ กันไป แต่ว่าดิสเพลย์จะไม่มีการเปลี่ยนแปลงให้เราเห็น วินโดว์ส่วนมากจะต้องมีการทำให้เคอร์เซอร์แตกต่างวินโดว์ปกติ เพื่อให้แยกความแตกต่างได้

คลาสของวินโดว์เมื่อได้กำหนดครั้งที่เริ่มสร้างวินโดว์แล้ว จะเปลี่ยนแปลงไม่ได้

3.17 เดบและวิชวล

เดบ(dept) และวิชวล(visual)ของวินโดว์เมื่อได้ค่าไว้เมื่อตอนสร้างวินโดว์แล้วเปลี่ยนแปลงไม่ได้ เดบ คือจำนวน แพลนบนวินโดว์ ซึ่งใช้เป็นตัวบอกเกร์สเกล(gray scale)หรือสีในวินโดว์นั้น เดบยังเป็นจำนวนบิทต่อพิกเซลด้วย สำหรับวินโดว์แบบ อินพุททางเดียวนั้นมีเดบสูงสุดเท่ากับจำนวนแพลนซึ่งจอภาพนั้นจะยอมรับได้ เช่น ถ้าจอภาพมี 12 แพลนวินโดว์นั้นจะมีได้ถึง 12 บิท ต่อพิกเซล และแน่นอนว่าสามารถมีระดับสี(shade) ต่างๆ กันได้ถึง 2^{12} ระดับสี หรือมีสีได้ 2^{12} สีในระบบที่เป็นจอสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วินโดว์แบบดินพุททางเดียวมีเดบเป็น 0 ส่วนวินโดว์แบบอินพุทเอาท์พุทถ้าเราใช้ CopyFrom-Parent ก็จะได้เดบของวินโดว์นั้นเท่ากับ depth ของ วินโดว์พ่อ

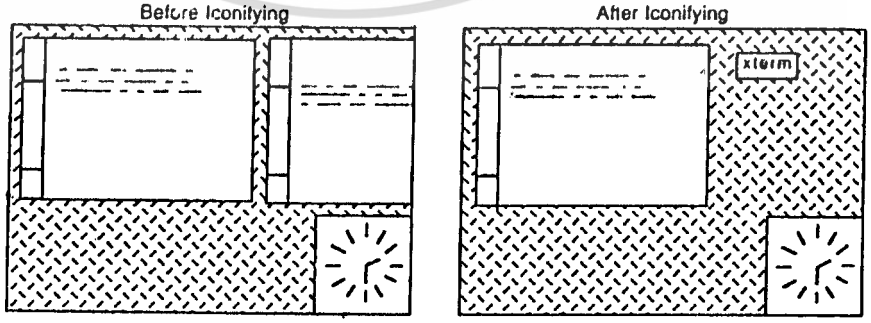
วิชวลจะบันทึกความแตกต่างระหว่างดิสเพลย์ต่าง ๆ เพื่อกำหนดวิธีการแปลงค่าของพิกเซลไปเป็นสีที่มองเห็นได้บนวินโดว์หนึ่ง ๆ จอภาพอาจมีเพียง 1 visual หรือมากกว่า 1 วิชวลก็ได้ XVisualInfo เป็นโครงสร้างข้อมูลที่เก็บค่าของ visual หนึ่งเป็นสมาชิกของ XVisualInfo อันหนึ่งก็คือคลาสของวิชวลซึ่งจะมีค่าอย่างใดอย่างหนึ่งจากค่าต่อไปนี้คือPseudocode,StaticColor, DirectColor,TrueColor,GrayScale หรือ StaticGray ค่าเหล่านี้เป็นตัวบอกลักษณะของตารางสี ที่ใช้กับวินโดว์ว่าตารางสีเป็นแบบ อ่าน/เขียน เป็นสีหรือเป็นโนโนโครม และสามารถแยกเป็นแม่สี 3 สี หรือสี องค์ประกอบ(composite) ได้หรือไม่สมาชิกตัวอื่นของ XVinsualInfo กำหนดของเขตของค่าของพิกเซล กำหนดจำนวนบิทของแต่ละพิกเซลซึ่งจะใช้ในการกำหนดสีต่อไป และก็ยังมีตัวแปรอื่น ๆ อีก ซึ่งจะได้กล่าวในที่นี้

ทั้งเดบและวิชวล จะถูกส่งมาจากพ่อถ้าเราสร้างวินโดว์โดยคำสั่ง XCreateSimpleWindow

3.18 ไอคอน(Icons)

ไอคอน(icon) ก็คือวินโดว์ขนาดเล็ก ซึ่งใช้เป็นตัวแทนของวินโดว์นั้นขณะมีขนาดใหญ่ ไอคอนนั้นยังทำงานอยู่ตลอดเวลา เพียงแต่ไม่ได้ถูกแมบลงบนจอในขณะนั้นเท่านั้นเอง

วินโดว์แมนนาเจอร์ส่วนใหญ่จะยอมให้ผู้สามารถกำหนดไอคอนด้วยแอพพลิเคชัน เพื่อไม่ให้สิ้นเปลืองเนื้อที่ แต่ก็ยังไม่ลบมันออกไป การ ดีไอคอนนีไฟ(deiconify) คือการขยายขนาดมันขึ้นมาเหมือน



รูปที่ 3.9 แสดงวินโดว์ชื่อ Xterm ก่อนและหลังการ iconify

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปกติจะทำให้เร็วกว่าการที่จะยกเลิกแอพพลิเคชันนั้นไปเลย แล้วเรียกขึ้นมาใหม่ แอพพลิเคชันที่ถูกไอคอนนิไฟ(iconify) ยังสามารทำงานได้อย่างต่อเนื่องตั้งแต่เริ่มต้นไอคอนนีไฟจนดีไอคอนนีไฟแล้ว (ถ้ามันไม่ได้ ฮอลท์(halt) ในขณะที่ไปทำ ไอคอนนีไฟ)เมื่อต้องการใส่ข้อมูล เราอาจจะรอจนกว่าจะมีการ ดีไอคอนนีไฟวินโดว์(deiconify window) นั้นหรือรับอินพุทในไอคอนนั้นเลย

วินโดว์แมนนาเจอร์เป็นตัวจัดการไอคอนวินโดว์(icon window) โดยแอพพลิเคชันจะผ่านค่าชื่อไอคอน(icon name) และพิกแมบ เพื่อแสดงผลในไอคอนวินโดว์ ถ้าแอพพลิเคชันต้องการจัดการกับไอคอนวินโดว์ ของตัวมันเอง อาจจะเพื่อเปลี่ยนนสีพื้น ทำได้โดยการผ่านค่าวินโดว์ไอดี(window ID)
ไปยังวินโดว์แมนนาเจอร์ มิฉะนั้น วินโดว์แมนนาเจอร์ จะสร้างไอคอนวินโดว์ ขึ้นมาใหม่

วินโดว์แมนนาเจอร์อาจจะกำหนดขนาดของไอคอนพิกแมบที่ต้องการในพรอบเพอตี้ของวินโดว์ราก เมื่อ ทำการเซ็ตพรอบเพอตี้แล้ว แอพพลิเคชันก็สามารถเตรียมไอคอนพิกแมบตามขนาดที่กำหนดได้ วินโดว์แมนนาเจอร์บางรุ่นของเอ็กซ์ สามารถกำหนดตำแหน่งของไอคอนพิกแมบได้

3.19 ลักษณะพิเศษของวินโดว์ราก

เมื่อโปรแกรมเอ็กซ์เซอร์เวอร์เริ่มทำงาน วินโดว์รากจะถูกสร้างขึ้น ลักษณะของวินโดว์รากต่างกับวินโดว์อื่น ๆ วินโดว์เป็นวินโดว์แบบอินพุททางเดียว ต้องถูกแมบเสมอ ขนาดของวินโดว์รากเปลี่ยนแปลงไม่ได้ มุมซ้ายของวินโดว์รากก็คือ มุมซ้ายของจอซึ่งโดยทั่วไปก็คือพิกัด(0,0) วินโดว์ราก ไม่มีกรอบ และโดยทั่วไปจะไม่ถูกไอคอนนิไฟโดยวินโดว์แมนนาเจอร์

3.20 Event คืออะไร

การย้ายตัวชี้(pointer) หรือการกด คีย์บอร์ด (keyboard) จะเป็นการทำให้เหตุการณ์เกิดขึ้น ซึ่งเป็นกรณีง่ายและเห็นได้ชัด ซึ่งอันที่จริงแล้วยังมีอีกหลายวิธีที่จะสร้างเหตุการณ์ขึ้นได้ เหตุการณ์อันที่จริงก็หมายถึงชุดของข้อมูล ซึ่งเซอร์ฟเวอร์สร้างขึ้นเมื่อมีการกระทำอย่างใดอย่างหนึ่ง เหตุการณ์จะถูกเก็บไว้ในคิว เพื่อให้ไคลแอนท์เรียกใช้ต่อไป ไคลแอนท์สามารถที่จะอ่านค่าเหตุการณ์จากคิวได้ตลอดเวลาภายใต้เซอร์ฟเวอร์ตัวเดียวกัน

ตัวอย่างชนิดของเหตุการณ์ เช่นการกดปุ่มเมาส์หรือปล่อยออก การแมบ หรือไม่มีการแมบวินโดว์ การเคลื่อนที่ของเมาส์เข้าออกจากวินโดว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดของเหตุการณ์มีกเป็นอินพุท ในการควบคุมการติดต่อระหว่างยูสเซอร์กับเครื่อง อีกอย่างหนึ่งก็คือ เหตุการณ์จะยอมให้ไคลแอนท์หลายตัวติดต่อกัน หรือติดต่อกับวินโดว์แมนนาเจอร์ เหตุการณ์ประเภทหลังอาจยกตัวอย่างได้ดังต่อไปนี้

1. ไคลแอนท์ต้องการให้มีการรับคีย์บอร์ด ที่วินโดว์ที่ต้องการโดยไม่สนใจว่าอยู่ที่ตำแหน่งไหนเรียกว่า คีย์บอร์ดโฟกัส(keyboard focus) การเปลี่ยนทิศทางของคีย์บอร์ดจากวินโดว์หนึ่งไปยังอีกวินโดว์หนึ่ง จะเกิดเหตุการณ์ชื่อ FocusIn FocusOut ชี้ให้เห็นว่าไคลแอนท์นั้นสามารถรับคีย์บอร์ดได้หรือไม่

2. การเปลี่ยนรูปแบบคีย์บอร์ดและรหัสที่สร้างขึ้นจะสร้างเหตุการณ์ที่ชื่อ KeyMapNotify ส่งไปยังไคลแอนท์ทั้งหมด

3. การเพิ่มกรอบของวินโดว์บนจอทำให้เกิดเหตุการณ์ชื่อ ReParentNotify

4. เมื่อมีการเปลี่ยนขนาดของวินโดว์ ซึ่งมีวัตถุอยู่ภายในจะเกิดเหตุการณ์ชื่อ GravityNotify

5. เมื่อไคลแอนท์ทำการเปลี่ยนแปลงพรอบเพอตี้ของวินโดว์จะเกิด PropertyNotify

6. การติดต่อระหว่างไคลแอนท์ ที่ยอมให้ยูสเซอร์เลือกข้อมูลต่าง ๆ และไคลแอนท์ที่ยอมให้ยูสเซอร์จัดว่างข้อมูลในวินโดว์ จะเกิดเหตุการณ์คือ SelectionNotify,SelectionRequest,การส่งเหตุการณ์เหล่านี้โดยใช้คำสั่ง XSendEvent

3.21 การเลือกเหตุการณ์มาเก็บไว้ในคิว

ไคลแอนท์ต้องทำการเลือกชนิดของเหตุการณ์ที่ต้องการสำหรับแต่วินโดว์ การเลือกนี้ทำได้โดยคำสั่ง XSelectInput ซึ่งจะทำการเปลี่ยนค่า event_mask ซึ่งเป็นลักษณะเฉพาะของวินโดว์ หรือโดยการใช้คำสั่งที่เปลี่ยนลักษณะเฉพาะ โดยตรงคือXChangeWindowAttribute

ยกตัวอย่างเช่น สกอลบาร์ (scroll bar) อาจต้องการเหตุการณ์ของการกดปุ่มเมาส์ แต่วินโดว์หลักอาจต้องการการรับคีย์บอร์ดแต่ไม่รับการกดเมาส์ วินโดว์แต่ละวินโดว์อาจมีเหตุการณ์ต่าง ๆ กันไปก็ได้

เหตุการณ์ของคีย์บอร์ดและตัวชี้จะถูกสร้างขึ้นโดยวินโดว์ที่เล็กที่สุดที่ล้อมรอบตัวชี้ นั้นอยู่ ถ้าพบเหตุการณ์ใน event_mask เหตุการณ์นั้นอาจถูกจัดเข้าในคิว แต่ถ้าไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พบเหตุการณ์นั้นจะถูกทิ้งไป ถ้าเหตุการณ์อยู่ในคิว IDของวินโดว์ ซึ่งได้รับเหตุการณ์นั้นจะถูกเก็บเป็นสมาชิกหนึ่งของโครงสร้างของเหตุการณ์

do not propagate mask สามารถตั้งค่าได้โดย XChangeWindowAttribute เหตุการณ์ซึ่งไม่ใช่คีย์บอร์ด หรือ ตัวชี้ ไม่สามารถขยายออกไป(propagate)ได้ มักปรากฏบนวินโดว์ใดจะรบให้ ซีพียู(CPU) เลือกไปใช้เมื่อมีเหตุการณ์เกิดขึ้น

เหตุการณ์หลายชนิดสามารถส่งไปยังไคลแอนท์มากกว่า 1 ตัวได้ ถ้าไคลแอนท์นั้นได้เลือกชนิดของเหตุการณ์นั้นไว้ ไคลแอนท์แต่ละตัวจะมี event_mask เป็นของตนเอง แม้ว่าจะมีวินโดว์หรือลักษณะเฉพาะของวินโดว์ชุดเดียวก็ตาม ไคลแอนท์ที่เป็นตัวสร้างวินโดว์ไม่ต้องการกระทำการใด ๆอีกแต่ไคลแอนท์ที่ต้องการรับเหตุการณ์จากวินโดว์ซึ่งไคลแอนท์นั้นไม่ได้สร้างขึ้น จะต้องเลือกเหตุการณ์ที่ต้องการจากวินโดว์ นั้น ๆ ด้วยรูทีน XSelectInput เหตุการณ์ที่ซ้ำซ้อนกันนั้นจะถูกส่งไปยังแต่ละวินโดว์ ซึ่งก็มีการจัดการต่อไปเป็นอิสระต่อกัน

3.22 คิวของเหตุการณ์(Event queue)

ไคลแอนท์แต่ละตัวจะมีคิวของเหตุการณ์ของตัวเอง ซึ่งจะรับเหตุการณ์ที่เลือกไว้แล้ว ซึ่งส่งไปยังเซอร์ฟเวอร์ ดูรูปที่ 5

ไคลแอนท์สามารถลบเหตุการณ์เมื่อใดก็ได้แล้วจักเหตุการณ์นั้น ๆ ตามชนิดของข้อมูลในโครงสร้างของเหตุการณ์ มีฟังก์ชันหลายอย่างที่รับอินพุทเข้ามา ซึ่งมีความแตกต่างกันในเรื่องของจำนวนวินโดว์และชนิดของเหตุการณ์ ไคลแอนท์ยังสามารถอ่านค่าเหตุการณ์จากคิวโดยไม่ต้องลบทิ้งหรือลบเหตุการบางอย่าง และเอากลับคืนมาเมื่อทำสำเสร็จแล้ว หรือลบเหตุการณ์ทิ้งไปเลยทั้งคิว เหตุการณ์จะถูกสร้างจากโปรแกรม เพื่อส่งข้อความไปยังวินโดว์แมนนาเจอร์หรือโปรแกรมอื่น ๆ

3.23 โครงสร้างของเหตุการณ์(Event structure)

มีเหตุการณ์ที่สำคัญมากที่สุดอันหนึ่งคือ expose eventซึ่งเขียนได้เป็น structure XExposeEvent ดังตัวอย่างในรูป 2-1เหตุการณ์นี้จะถูกสร้างขึ้นเมื่อวินโดว์ถูกทำให้ปรากฏบนจอภาพ และเป็นตัวบอกว่า ไคลแอนท์ต้องมีการวาดพื้นที่ใหม่ สิ่งเหล่านี้ถูกทำให้เมื่อวินโดว์ถูกเคลื่อนย้าย,เปลี่ยนแปลงขนาด,iconify,เปลี่ยนแปลงขนาดของกรอบ หรือการปิดวินโดว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งถูกบังอยู่ exposesure eventเป็นเหตุการณ์ที่เกิดขึ้นมาเสมอ เนื่องจากสามารถเกิดขึ้นเพราะการกระทำใด ๆ ของไคลแอนท์ตัวอื่น ๆ ด้วย

3.24 <u>ลูปของเหตุการณ์(event loop)</u>

เนื่องจากลำดับการเข้ามาของเหตุการณ์ต่าง ๆ ไม่แน่นอน ดังนั้นจึงต้องเตรียมโปรแกรมที่จะจัดการกับเหตุการณ์ให้ดี ทุก ๆ โปรแกรมจะประกอบด้วยลูปของเหตุการณ์ ซึ่งจะมีเหตุการณ์เข้ามาและถูกประมวลผล ปกติเราจะใช้การวนลูปแบบไม่สิ้นสุด(infinite while loop) เริ่มต้นด้วยรูทีนที่จะรอรับเหตุการณ์ ตามด้วยคำสั่งภาษาซี คือสวิช(switch) ซึ่งจะมีตัวเลือกไปยังโปรแกรมต่าง ๆ ภายในแต่ละตัวเลือกของชนิดของเหตุการณ์อาจมีตัวเลือกย่อยลงไปอีกซึ่งสอดคล้องกับวินโดว์ที่เกิดเหตุการณ์นี้ขึ้น

ในลูปส่วนใหญ่จะประกอบไปด้วยเอ็กซ์โพสอีเวนต์ (expose event)เสมอ เนื่องจากปกติแล้วระบบเอ็กซ์วินโดว์ไม่ได้มีการจัดการเก็บส่วนประกอบต่าง ๆ ของวินโดว์ซึ่งถูกบังอยู่ ซึ่งเป็นหน้าที่ของโปรแกรมซึ่งต้องมั่นใจได้ว่าสามารถวาดวินโดว์ใหม่ได้เหมือนเดิม เมื่อวินโดว์นั้นปรากฏขึ้นมาอีก โปรแกรมต้องมีการจัดการให้มีการรับและทำงานบนเอ็กซ์โพสอีเวนต์ ตลอดมา

เมื่อวินโดว์แมบครั้งแรก ฟังก์ชั่นแรกของโปรแกรมต้องเป็นการอ่าน ซึ่งถูกสร้างขึ้นเมื่อมีการแมบ วินโดว์ใด ๆ และมีการตอบสนองการเปลี่ยนแปลง ตำแหน่ง และความหนาของขอบ ซึ่งวินโดว์แมนนาเจอร์ยอมให้ทำได้ ต่อมาโปรแกรมก็จะสามารถ draw องค์ประกอบภายในวินโดว์ได้ นอกจากนั้นเมื่อออกจากส่วนของการจัดการเอ็กซ์โพสอีเวนต์ ไปแล้วต้องมีการเตรียมการสำหรับเอ็กซ์โพสอีเวนต์ ครั้งต่อไป ซึ่งอาจจะมีขึ้นได้อีก การวาดครั้งแรกและการวาดใหม่ครั้งต่อ ๆ มาจะต้องทำในลักษณะเดียวกันและใช้ส่วนของโปรแกรมเดียวกันด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

----------***----------

บทที่ 4

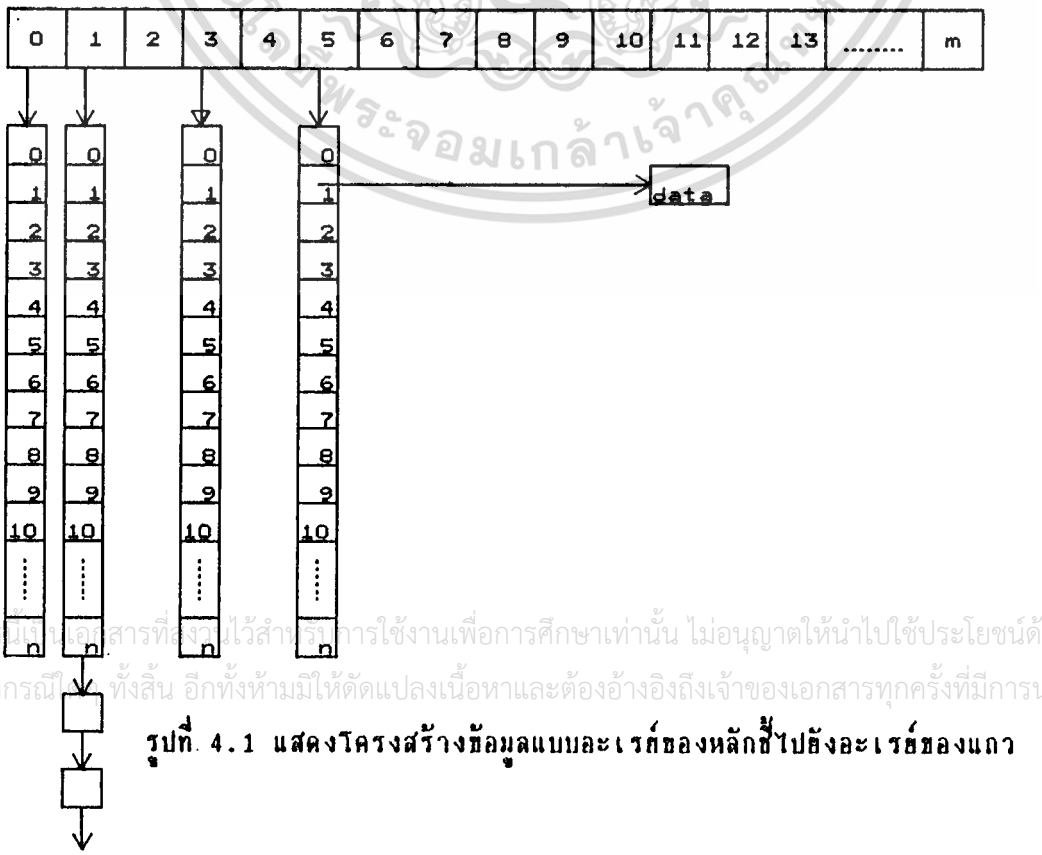
โครงสร้างข้อมูล

(Data structure)

ในการทำสเปรดชีต สิ่งสำคัญอย่างหนึ่งก็คือ โครงสร้างข้อมูลควรจะมีลักษณะเป็นอย่างไร เนื่องที่เวลาทำงานแล้วจะสามารถจัดการข้อมูลที่มีลักษณะเป็นเซลล์ ๆของสเปรดชีตที่มีเป็นนับล้านเซลล์ได้ ดังนั้นการสร้างสเปรดชีตครั้งนี้ ลักษณะของโครงสร้างข้อมูล จึงมีความสำคัญเป็นอย่างมาก และในการทำครั้งนี้ ได้มีการออกแบบโครงสร้างข้อมูลอยู่ 3 ลักษณะ แต่ละลักษณะมีข้อดีข้อเสียแตกต่างกันคือ

4.1 แบบแรก

จะมีอะเรย์ของตัวชี้ (Pointer) ขนาด 1 มิติ เก็บตำแหน่งเริ่มต้นของแต่ละหลัก (Column) โดยที่แต่ละหลักก็จะเป็นอะเรย์ของตัวชี้ ขนาด 1 มิติ ที่มีตัวชี้ ชี้ไปยังตำแหน่งที่เก็บข้อมูล และถ้าข้อมูลในแต่ละหลักนั้นมีจำนวนมากกว่าที่กำหนดไว้ ในแถว (Row) ต่อจากนั้น ก็จะเก็บเป็นลักษณะของลิงค์ลิส(Link List) ดังรูป



รูปที่ 4.1 แสดงโครงสร้างข้อมูลแบบอะเรย์ของหลักชี้ไปยังอะเรย์ของแถว

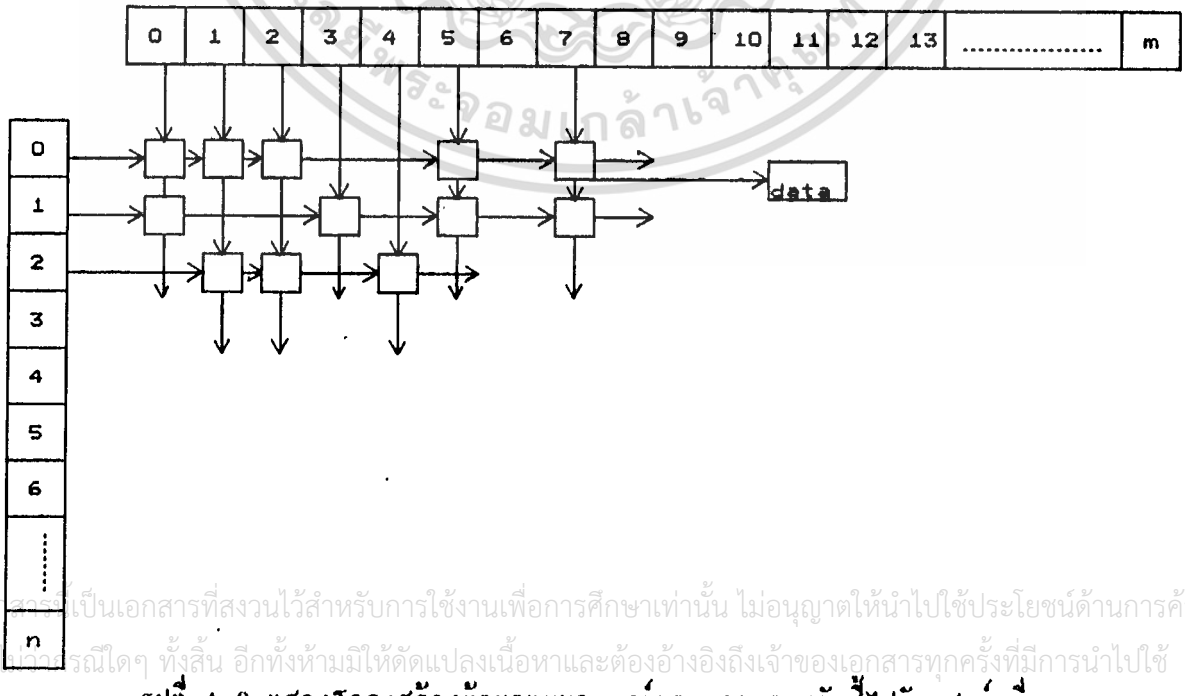
เอกสารนี้จัดทำขึ้นสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างข้อมูลแบบนี้ มีข้อเสียอยู่ตรงที่ ถ้ามีการอ้างข้อมูลไปทีละแถว ก็จะไม่มีตัวชี้ไปยังแถวเดียวกันในหลักถัดไป ต้องอ้างไปที่แถวแรกของหลักถัดไปก่อน แล้วจึงอ้างไปยังแถวที่ต้องการได้ ทำให้เกิดการล่าช้า แต่ถ้ามีการอ้างข้อมูลไปทีละหลัก โครงสร้างข้อมูลแบบนี้ก็จะทำงานได้เร็ว เพราะโครงสร้างแบบอะเรย์ในแต่ละหลักนั้นจะช่วยให้ทำงานได้เร็วขึ้น

ข้อเสียอีกอย่างหนึ่งของโครงสร้างข้อมูลแบบนี้คือ ถ้าหากมีการใช้งานเฉพาะแถวแรกๆ หรือแถวใดแถวหนึ่งของทุกๆ หลัก ก็จะเป็นการเปลืองเนื้อที่หน่วยความจำในการจองอะเรย์ของแต่ละหลักมากจนเหมือนกับการจองอะเรย์ขนาด 2x2 ใหญ่ ๆ ที่ใช้งานจริงเพียงเล็กน้อย ทั้งนี้ขึ้นกับขนาดของอะเรย์ในแต่ละหลักว่าจะมีขนาดเท่าไร ถ้ามีขนาดใหญ่ก็จะเปลืองมาก ถ้ามีขนาดเล็กก็จะเปลืองน้อย แต่ก็จะสูญเสียจุดประสงค์ในการมีโครงสร้างแบบนี้ ถ้าหากว่าจะต้องต่อลิงค์ลิสต์(link list) ในแต่ละหลักออกไปอาวๆ

4.2 <u>แบบที่สอง</u>

เป็นการแก้ปัญหาในแบบแรก คือ จะมีอะเรย์ของตัวชี้ เก็บตำแหน่งเริ่มต้นของแต่ละแถว และแต่ละตัว ก็จะมีตัวชี้ไปยังหลักถัดไปที่อยู่ในแถวเดียวกัน โดยที่จะชี้ไปทีละเซลล์ๆ (เพื่อประหยัดเนื้อที่หน่วยความจำ)



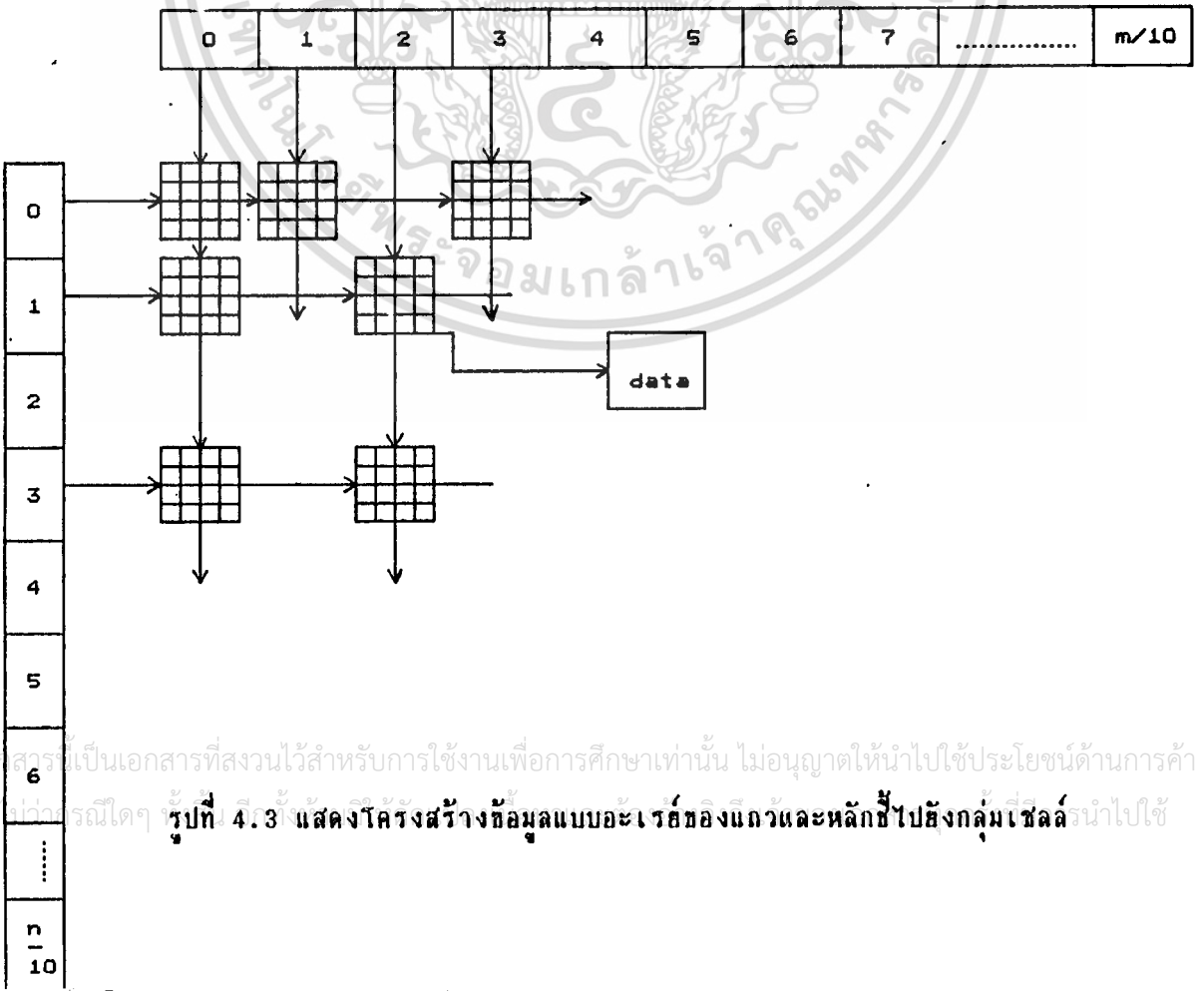
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 แสดงโครงสร้างข้อมูลแบบอะเรย์ของแถวและหลักชี้ไปยังเซลล์เดี่ยว ๆ

โครงสร้างแบบนี้ช่วยแก้ปัญหาในแบบแรกได้ แต่ก็ยังมีปัญหาอีกคือ ถ้าหากมีข้อมูลอยู่มากๆ การอ้างถึงเซลล์ที่อยู่ทางท้ายๆ จะต้องอ้างหลายครั้ง เช่นในสูตรคำนวณ จะต้องเอาข้อมูลจากเซลล์ที่อยู่แถวที่ 1,000 หลักที่ 2,000 มาใช้ ถ้าอ้างแถวไปเรื่อยๆ ก็อาจจะต้องอ้างถึง 2,000 ครั้ง ถ้าในแถวนั้นมีข้อมูลอยู่เกือบจะทุกหลัก ถ้าอ้างตามหลักก็อาจจะต้องอ้างถึง 1,000 ครั้ง ถ้าหลักนั้นมีข้อมูลอยู่เกือบทุกแถว ทำให้ทำงานได้ช้า

4.3 <u>แบบที่สาม</u>

เป็นการแก้ปัญหาของโครงสร้างข้อมูลแบบที่สอง จากแนวความคิดที่ว่าลักษณะของข้อมูลในสเปรดชีต จะอยู่กันเป็นกลุ่มๆ ดังนั้นแทนที่จะแยกไปเป็นเซลล์ ๆ เหมือนในแบบที่สอง ก็ควรจะสร้างโครงสร้างที่เป็นกลุ่มขึ้นมา ขนาดของกลุ่มเซลล์แต่ละกลุ่มนั้น ไม่ควรจะเล็กเกินไป เพราะถ้าเล็กเกินไปก็ยังไม่ถือว่าแก้ปัญหาของแบบที่สองได้ แต่ถ้ากลุ่มใหญ่เกินไป ก็อาจจะเกิดปัญหาการปลืองเนื้อที่โดยเปล่าประโยชน์ขึ้นมา ถ้าหากไม่มีการใช้เซลล์ในกลุ่มนั้นเป็นส่วนมาก ดังนั้นในที่นี้จึงใช้กลุ่มขนาด 10x10 โดยเป็นอะเรย์ของตัวชี้ 2 มิติ ขนาด 10x10 โดยที่แต่ละเซลล์ในกลุ่มนั้น ก็จะมีตัวชี้ไปยังข้อมูลในตำแหน่งเซลล์นั้นๆ ดังรูป



รูปที่ 4.3 แสดงโครงสร้างข้อมูลแบบอะเรย์ของแถวและหลักชี้ไปยังกลุ่มเซลล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

โครงสร้างข้อมูลแบบนี้ช่วยแก้ปัญหาของโครงสร้างข้อมูบแบบที่ 2 ได้ แต่ถ้าหากว่ามีการทำงานเกี่ยวกับการแทรกหลัก (Insert Column) แทรกแถว (Insert Row) ลบหลัก (Delete Row) โครงสร้างข้อมูลแบบนี้จะทำได้ช้ากว่าแบบที่สอง แต่การทำงานลักษณะ ลบ, แทรก, แถวหรือหลัก มีขึ้นไม่บ่อยนักถ้าเทียบกับการอ้างถึงเซลล์เป็นเซลล์ๆ ดังนั้นการสร้างสเปรดชีตในครั้งนี้ จึงใช้โครงสร้างข้อมูลแบบนี้

4.4 <u>การแทรกแถว</u>

ก่อนอื่นจะตรวจสอบดูก่อนว่า จะมีข้อมูลถึงหลักสุดท้ายหลักที่เท่าไร จากนั้นก็จะเริ่มสร้างกลุ่มข้อมูลเพิ่มขึ้นมาในแต่ละหลัก แล้วก็ทำการย้ายตัวชี้ที่เก็บไว้ในแต่ละเซลล์ลงไปในเซลล์แถวถัดไปเรื่อย ๆ จนถึงแถวที่ต้องการแทรกจึงหยุดย้าย แล้วก็ทำเช่นเดียวกันกับกลุ่มข้อมูลในหลักถัดไป จนกระทั่งถึงหลักสุดท้ายที่มีข้อมูลอยู่

4.5 <u>การแทรกหลัก</u>

ก่อนอื่นจะตรวจสอบดูก่อนว่า จะมีข้อมูลถึงแถวสุดท้ายแถวที่เท่าไร จากนั้นก็จะเริ่มสร้างกลุ่มข้อมูลเพิ่มขึ้นมาในแต่ละแถว แล้วก็ทำการย้ายตัวชี้ที่เก็บไว้ในแต่ละเซลล์ลงไปในเซลล์หลักถัดไปเรื่อย ๆจนถึงหลักที่ต้องการแทรกจึงหยุดย้าย แล้วก็ทำเช่นเดียวกันกับกลุ่มข้อมูลในแถวถัดไป จนกระทั่งถึงแถวสุดท้ายที่มีข้อมูลอยู่

4.6 <u>การลบแถว</u>

ก่อนอื่นจะตรวจสอบดูก่อนว่า จะมีข้อมูลถึงหลักสุดท้ายหลักที่เท่าไร แล้วก็ทำการลบข้อมูลในเซลล์ที่อยู่ในแถวที่ต้องการลบออก จากนั้นก็เริ่มทำการย้ายข้อมูลจากเซลล์แถวล่างขึ้นมาเซลล์แถวบน จะหมดในกลุ่มข้อมูลกลุ่มแรก แล้วทำเช่นเดียวกันในกลุ่มข้อมูลในหลักถัดไป จนถึงหลักสุดท้ายที่มีข้อมูลอยู่

4.7 <u>การลบหลัก</u>

ก่อนอื่นจะตรวจสอบดูก่อนว่า จะมีข้อมูลถึงแถวสุดท้ายแถวที่เท่าไร แล้วก็ทำการลบข้อมูลในเซลล์ที่อยู่ในหลักที่ต้องการลบออก จากนั้นก็เริ่มทำการย้ายข้อมูลจากเซลล์หลักล่างขึ้นมาเซลล์หลักบน จะหมดในกลุ่มข้อมูลกลุ่มแรก แล้วทำเช่นเดียวกันในกลุ่มข้อมูลในหลักถัดไป จนถึงแถวสุดท้ายที่มีข้อมูลอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 การเติมกลุ่มข้อมูลที่เป็นตัวเลข (Fill Data)

เมื่อรับค่าแถวและหลักเริ่มต้น และแถวและหลักสุดท้ายที่ต้องการเติมข้อมูล, ค่าเริ่มต้น, ค่าสุดท้าย, และสเต็ปของการเพิ่มหรือลดข้อมูล แล้วก็จะทำการสร้างกลุ่มข้อมูลในช่วงที่ต้องการเติมนั้น แล้วเริ่มทำการเติมข้อมูลจากเซลล์แรก ในแถวแรก ไปเรื่อยๆ จนถึงแถวสุดท้าย แล้วเริ่มในแถวแรกในหลักถัดไป จนกระทั่งถึงค่าสุดท้ายที่ต้องการ หรือถึง แถว, หลักสุดท้ายที่ต้องการเติม

4.9 การคัดลอกกลุ่มข้อมูล

เมื่อรับค่า แถว, หลักเริ่มต้น และแถว, หลักสุดท้ายที่จะทำการคัดลอก และแถว, หลักแรกที่จะคัดลอกข้อมูลไปแล้ว ถ้าแถว, หลักที่จะคัดลอกไปไม่อยู่ในช่วงของแหล่งที่จะคัดลอก ก็จะเริ่มทำการคัดลอกข้อมูลเซลล์ต่อเซลล์ไปเรื่อยๆ จนหมด โดยไม่ลบข้อมูลที่แหล่งเดิม

4.10 การย้ายข้อมูล

เมื่อรับค่า แถว, หลักเริ่มต้น และแถว, หลักสุดท้ายที่จะทำการย้าย และแถว, หลักแรกที่จะย้ายข้อมูลไปแล้ว ถ้าแถว, หลักที่จะย้ายไปไม่อยู่ในช่วงของแหล่งที่จะย้าย ก็จะเริ่มทำการย้ายข้อมูลเซลล์ต่อเซลล์ไปเรื่อยๆ จนหมด โดยลบข้อมูลที่แหล่งเดิม

4.11 การเก็บข้อมูลของสเปรดชีต (Save Data)

เมื่อทำการรับชื่อไฟล์ที่จะเก็บแล้ว ก็จะทำการตรวจดูว่ามีไฟล์นั้นอยู่หรือไม่ ถ้ามีแล้วต้องการจะเขียนทับหรือไม่ ถ้าหากยังเป็นไฟล์ใหม่หรือไฟล์เก่าที่ต้องการเขียนทับ ก็จะทำการเขียนคำว่า "Spread sheet" ลงไป เพื่อที่จะได้รู้ว่าเป็นไฟล์ของสเปรดชีต จากนั้นก็ทำการเก็บค่ากลุ่มแถวและหลักสุดท้ายที่มีข้อมูลอยู่ จากนั้นก็เริ่มเก็บทีละเซลล์ ๆ โดยที่จะเก็บค่า หลัก,แถว,แอตทริบิวที่บอกว่าเซลล์นั้นๆ เป็นตัวอักษร,ตัวเลข,หรือสูตรคำนวณลงไป และเก็บค่าของเซลล์วNนั้นๆ ลงไปจนกระทั่งหมดทุกเซลล์ จึงทำการปิดไฟล์

4.12 การโหลดข้อมูล

ถ้ามีไฟล์ที่ต้องการโหลดเก็บอยู่จริงๆ ก็จะทำการอ่านมา ถ้ามีคำว่า "Spreadsheet" เขียนอยู่ตรงหัว แสดงว่าเป็นไฟล์ของสเปรดชีตจริง ก็จะทำการอ่านมา โดยจะทำการอ่านค่ากลุ่ม แถว และหลักสุดท้าย ค่าหลักของเซลล์ ค่าแถวของเซลล์ และข้อมูลในเซลล์ แล้วทำการตรวจดูว่าเป็นข้อมูลแบบไหน ก็ทำการเก็บข้อมูลนั้นๆ ไปตามแบบของข้อมูล อ่านมาเรื่อยๆ จนหมด จึงทำการปิดไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
อินเตอร์เฟส

5.1 เอ็กซ์วิว(X View)

การศึกษาเอ็กซ์วิวนับว่ามีประโยชน์เป็นอย่างมาก โดยเฉพาะในส่วนอินเตอร์เฟสแล้วนั้น โครงสร้างที่ดีของเอ็กซ์วิว สามารถช่วยผู้เขียนโปรแกรมได้มากทีเดียวในการพัฒนาโปรแกรมบนระบบวินโดว์ให้ใช้งานได้อย่างมีประสิทธิภาพและง่ายต่อการใช้งาน เอ็กซ์วิวสามารถเชื่อมต่อกับระบบเอ็กซ์วินโดว์ได้โดยตรง ด้วยรูปแบบและฟังก์ชันที่กำหนดให้ การโปรแกรมกับเอ็กซ์วิวเป็นการโปรแกรมเชิงวัตถุ (Object Oriented Programming) ช่วยให้การโปรแกรมทำได้ง่ายขึ้น

5.2 ระบบเอ็กซ์วิวและเอ็กซ์วินโดว์

Xview Toolkit ยอมให้ผู้เขียนโปรแกรมสร้างส่วนอินเตอร์เฟสบนโปรแกรมประยุกต์ได้โดยไม่จำเป็นต้องเรียนรู้รายละเอียดภายใต้ระบบวินโดว์ อย่างไรก็ตามประโยชน์ต่อการโปรแกรมก็มีเพียงเบื้องต้นเท่านั้น ในการโปรแกรมต้องอาศัยความเข้าใจในระบบวินโดว์อีกมากในกรณีที่ซับซ้อนลงไป ดังนั้นพื้นฐานของระบบวินโดว์ จึงเป็นส่วนสำคัญที่ต้องศึกษาร่วมกันกับเอ็กซ์วิว

5.2.1 พัฒนาการของ Xview Toolkit

หลายๆปีที่ผ่านมาบริษัท Sun microsystem ได้พัฒนาเครื่องมือช่วยทำโปรแกรมขึ้นมากมาย แต่ละอันล้วนแล้วแต่ถูกออกแบบมาอย่างดี และมีฟังก์ชันการทำงานมากขึ้นเรื่อยๆในรุ่นใหม่ ๆ อันแรกที่ได้รับการยอมรับอย่างดีจากผู้ใช้คือ SunView I เพราะมีองค์ประกอบจำเป็นพื้นฐานทั้งหมดต่อการสร้างฟังก์ชันต่างๆของ VSL interface รวมทั้งระบบ API(application programmer'interface) ที่โดดเด่น แบบจำลองของ XView Programmer มีโปรซีเดอร์เรียกใช้งานเพียงเล็กน้อย ตัวอย่างเช่น หากจะสร้างอ็อปเจคใด ๆขึ้นมาใช้ก็เรียกใช้เพียงฟังก์ชันเดียวก็ทำได้ มีข้อเลือกต่างๆให้เลือกใช้งานอีกมาก ในกรณีที่อ็อปเจคถูกสร้างขึ้นด้วยจำนวนบรรทัดเพียงแค่ 2-3 บรรทัด หรือในการเซตค่า ลักษณะเฉพาะเมื่อเกิดเปลี่ยนแปลง ลักษณะเฉพาะเหล่านี้สามารถเซตได้อีกในขณะเวลาที่อ็อปเจคถูกสร้างขึ้น

ปัจจุบันมีโปรแกรมประยุกต์ หลาย ๆอย่างใน Sun View ที่ได้เสนอออกมาและจุดประสงค์ของเอ็กซ์วิวก็คือ เพื่อที่จะเปิดหนทางนำโปรแกรมประยุกต์ต่างๆสู่ระบบเอกซ์วินโดว์ในตลาด มากกว่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sun ได้ทำตัวโปรแกรม(Source code) ให้เอ็กซ์วิวใช้อย่างมากมายและมีมาตราฐานร่วมกันกับ MITX ที่ปล่อยออกมากับระบบยูนิกซ์เวอร์ชั่นห้า(UNIX System V)

5.2.2 เอ็กซ์วิวจัดให้มีชุดของวินโดว์ต่อไปนี้

-Canvases ใช้สำหรับวาดสิ่งต่างๆได้

-Text subwindows จะมีอีดิเตอร์(editor)ซ่อนอยู่ภายใน ทำงานได้อย่างมีประสิทธิภาพ

-Panels บรรจุหัวข้อ(item) ต่างๆเช่น ปุ่ม(buttons),หัวข้อไว้เลือก(choice item) และสไลเดอร์(sliders)

-Tty subwindows ที่จำลองการทำงานของเทอร์มินัลได้

5.3 แบบจำลองของการโปรแกรมกับเอ็กซ์วิว

เอ็กซ์วิวประสงค์ที่จะพัฒนางานประยุกต์ต่าง ๆ บนระบบเอกซ์วินโดว์ โดยได้จัดหาชุดองค์ประกอบของ VSL interface ที่จำเป็นไว้ให้โปรแกรมเมอร์ได้ใช้ องค์ประกอบเหล่านั้นใช้งานแบบ "look and feel" ของ OPEN LOOK GRAPHIC USER INTERFACE ในบทนี้จะได้นำเสนอแบบจำลองของเอ็กซ์วิวสำหรับผู้เขียนโปรแกรม ซึ่งเป็นส่วนสำคัญที่ผู้เขียนโปรแกรมจะต้องเข้าใจแบบจำลองนี้ก่อนเริ่มต้นเขียนงานประยุกต์ อย่างไรก็ตามถ้าหากใครพอมีพื้นฐานแล้วก็สามารถข้ามไปเริ่มเขียนโปรแกรมได้

5.3.1 การโปรแกรมเชิงวัตถุ

(Object-Oriented Programming)

เอ็กซ์วิวเป็นชุดเครื่องมือเชิงวัตถุ(object-oriented toolkit) ดังนั้นเอ็กซ์วิวจึงมีหลักการอยู่บนพื้นฐานหลาย ๆ อย่างของการโปรแกรมเชิงวัตถุ (object-oriented programming)

-อ็อปเจคถูกเสนอด้วยในรูปแบบลำดับชั้น(hierachy)

-อ็อปเจคถูกปิดบังชนิดของข้อมูลเอาไว้

-อ็อปเจคมีลักษณะเฉพาะซึ่งสามารถเซตข้อความได้ตลอดเมื่อส่งผ่านฟังก์ชั่น

-อ็อปเจคมี callback procedure ที่ถูกทริกด้วยเหตุการณ์ซึ่งจะได้กล่าวในราย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ละเอียดการใช้งาน แต่ละอย่างต่อไป

5.3.1.1 <u>คลาสอ็อปเจคแบบลำดับชั้น</u>

(Object class hierachy)

เอ็กซ์วิวกำหนดคลาสของอ็อปเจคในรูปของลำดับทรี(Tree hierachy) ตัวอย่างเช่น เฟรมเป็นคลาสย่อยของวินโดว์ทั่ว ๆ ไป ซึ่งก็เป็นคลาสย่อยของดรอเอเบิลด้วย ส่วนของดรอเอเบิลคือ ส่วนอ็อปเจคเชื่อมต่อผู้ใช้เป็นคลาสย่อยของอ็อปเจ็คทั่ว ๆไป รูปที่ 5.1 แสดงให้เห็นความสัมพันธ์แบบลำดับชั้นของเอ็กซ์วิว แต่ละคลาสได้คุณสมบัติที่ทำให้แต่ละอันเป็นอิสระต่อกันหรือเป็นแพคเกจ (Package) ในเอ็กซ์วิว คลาสต่าง ๆ มักเรียกว่าแพคเกจ หมายถึงเซ็ตของความสัมพันธ์ของสมาชิกฟังก์ชั่น อย่างไรก็ตามก็มีแพคเกจของเอ็กซ์วิวบางอันที่ไม่เป็นสมาชิกของความสัมพันธ์แบบเป็นลำดับชั้น เช่นแพคเกจสำหรับส่งข่าวสาร เป็นต้น

บางชนิดก็เป็นอ็อปเจคเสมือน(Vitual Object) บางอย่างก็ไม่เป็น อ็อปเจคเสมือนก็รวมทั้งวินโดว์,สกอลล์บาร์,เฟรม,พาเนล,หัวข้อของพาเนล,และอื่น ๆ พวกที่ไม่ใช่อ็อปเจคเสมือนจะไม่ปรากฏให้เห็น แต่มีข้อสนเทศที่ช่วยในการแสดงผลแก่อ็อปเจคเสมือน ตัวอย่างอ็อปเจคที่ไม่ใช่อ็อปเจคเสมือนเช่น เซอร์ฟเวอร์,สกรีน,และฟอนต์ตัวอักษรต่าง ๆ สกรีนจะมีข้อสนเทศอาทิเช่นชนิดของสีที่จะไปแสดง คุณสมบัติสามารถถ่ายทอดไปที่อ็อปเจคอื่นในวินโดว์ได้ด้วย ดีสเพลย์ก็จัดให้มีข้อมสนเทศเกี่ยวกับแบบตัวอักษรสำหรับการแสดงเท็กซ์

อ็อปเจคทั้งหมด ทั้งเสมือนและแบบไม่ใช่เสมือนเป็นส่วนหนึ่งของระบบอ็อปเจค ซึ่งระบบนี้เป็นแบบพัฒนาต่อได้ ดังนั้นจึงสามารถสร้างคลาสใหม่ ๆ ขึ้นมาอีกโดยอาจจะไม่อยู่ในประเภทที่มีอยู่แล้วก็ได้

เอ็กซ์วิวใช้คลาสย่อยสแตติกและสืบทอดลักษณะเป็นแบบลูกโซ่ เป็นส่วนหนึ่งของแบบจำลองเชิงวัตถุ อ็อปเจคของแต่ละคลาสจะรับถ่ายทอดคุณสมบัติจากพ่อแม่(ที่เรียกว่าซูเปอร์คลาส) XV_OBJECT ทั่ว ๆ ไปได้บรรจุคุณสมบัติพื้นฐานที่ทุก ๆ อ็อปเจคใช้ร่วมกันได้ เช่น อ็อปเจคชนิดเดียวกันสามารถปรากฏได้ในหลาย ๆ สถานที่บนจอ เพื่อประสิทธิภาพในการใช้หน่วยความจำ เพื่อเก็บทะเบียนของสิ่งเหล่านี้ อ็อปเจคโดยทั่วไปจึงต้องรักษาตัวอ้างอิงของคุณลักษณะร่วม

แต่ละคลาสบรรจุคุณสมบัติที่ใช้ร่วมกันระหว่าง Instancs ทั้งหมดของอ็อปเจค ตัวอย่างเช่น พาเนลเป็นส่วนหนึ่งของพาเนลแพคเกจ ซึ่งมีคุณสมบัติอธิบายได้ดังนี้คือ ระหว่างสิ่งของอันอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงร่างของพาเนล(แนวนอนหรือแนวตั้ง)หรือช่องว่างของปุ่มในพาเนล พาเนลทั้งหมดจะใช้คุณสมบัติร่วมกัน แม้ว่าสถานะของคุณสาบัติจะต่างกันสำหรับแต่ละ Instance ของอ็อปเจค

5.3.1.2 การรับส่งอ็อปเจค

(Object Handles)

เมื่อสร้างอ็อปเจค ฟังก์ชั่นของเอ็กซ์วิวจะส่งค่ากลับมารับค่าสำหรับอ็อปเจคนั้น ต่อมาเมื่อประสงค์จะค่านวณกับอ็อปเจค หรือถามข้อมูลกับสถานะ ทำได้โดยการผ่านค่าในฟังก์ชั่นที่เหมาะสมมา สิ่งต่างๆที่อยู่บนตัวรับอ็อปเจคเป็นลักษณะของการซ่อนข้อสนเทศ(information-hinding) ผู้ใช้จะไม่สามารถเห็นโครงสร้างข้อมูลจริงๆของการแสดงอ็อปเจค

แต่ละชนิดของอ็อปเจคมีชนิดการรับส่งต่างๆกัน เพราะในภาษาซีไม่มีชนิดของการบดบัง แบบของข้อมูลที่ถูกซ่อนทั้งหมดอยู่ภายใต้การกำหนดชนิด(typedef'd)แก่เอ็กซ์วิวใน Xv_opaque และ XV_object

มากกว่านั้นแบบข้อมูลที่ถูกซ่อน ได้นิยามหลายแบบที่ไม่อ้างอิงแบบพอยเตอร์แต่อ้างอิงแบบโครงสร้าง: Event,Rect,และ Rectlist โดยทั่วไปแล้วตัวชี้ข้อมูลโครงสร้างจะถูกผ่านไปให้ฟังก์ชั่นของเอ็กซ์วิว ดังนั้นจึงต้องกำหนดเป็น Event *,Rect * เป็นต้น

Object	Owner	Package	Handle Data Type
canvas	frame	CANVAS	Canvas
canvas view	window or screen	CANVAS_VIEW	Canvas_view
cursor	window or cursor	CURSOR	Xv_cursor
font	root window	FONT	Xv_font
.	.	.	.
.	.	.	.
window	frame	WINDOW	Xv_window

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.1 รายการของเอ็กซ์วิวอ็อปเจค,เจ้าของอ็อปเจค,แพคเกจที่กำหนด,และรูปแบบของข้อมูล

5.4 ฟังก์ชั่นคุณลักษณะเฉพาะพื้นฐาน

แบบจำลองที่ใช้โดยเอ็กซ์วิว ซึ่งถูกวางรากฐานมาอย่างซับซ้อนและอ็อปเจคที่ยืดหยุ่นต่อการทำงาน แสดงถึงปัญหาที่ยุ่งยากในการติดต่อกับไคลแอนท์ แนวความคิดพื้นฐานเบื้องหลังการเชื่อมต่อได้จัดหาไว้ให้เป็นฟังก์ชั่นแต่เพียงเล็กน้อย ซึ่งแบกอาทิวเมันชุดขนาดใหญ่ของลักษณะเฉพาะ

5.4.1 การสร้างและการใช้งานอ็อปเจค

ชุดของฟังก์ชั่นปกติที่อนุญาตุให้ผู้เขียนโปรแกรมทำงานกับอ็อปเจคโดยอ้างอิงการับส่งอ็อปเจค รายละเอียดของฟังก์ชั่งแสดงในตาราง 5.2

ฟังก์ชั่น	กฎการใช้
Xv_init()	สร้างขั้นมาติดต่อกับเซอร์ฟเวอร์,ให้ค่าเริ่มต้นแก่ตัวแจ้งข้อมูลข่าวสารและผู้จัดการทรัพยากรในฐานข้อมูล,ทำการดึงค่าจากฐานข้อมูลและอ่านค่า *Xdefault จากฐานข้อมูล และทำการส่งค่าลักษณะเฉพาะต่าง ๆ
Xv_create()	สร้างอ็อปเจคในวินโดว์
Xv_destroy()	ทำลายอ็อปเจค
Xv_find()	ค้นหาอ็อปเจคที่อยู่ในพื้นที่ หรือทำการสร้างใหม่ถ้าหากอ็อปเจคนั้นไม่มี
Xv_get()	ดึงค่าของลักษณะเฉพาะ
Xv_set	เซ็ตค่าของลักษณะเฉพาะ

ตาราง 5.2 ชุดฟังก์ชั่นหลักของเอ็กซ์วิว

5.4.2 ชนิดของลักษณะเฉพาะ

ลักษณะเฉพาะสามารถแบ่งออกได้เป็น 3 ประเภทคือ แบบทั่วไป(generic attribute),แบบสามัญ(commond attribute),และแบบเฉพาะ(specified attribute) เอ็กซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิวใช้ชื่อดั้งเดิมกำหนดชนิดการทำงานของลักษณะเฉพาะ ที่ประยุกต์ใช้กับแพคเกจเฉพาะอัน ลักษณะเฉพาะมีคำอุปสรรคนำหน้า เช่น CANVAS_*,CURSOR_*,FRAME_*,ICON_*,MENU_*,PANEL_*, SCROLLBAR_*,TEXTSW_*,TTY_* เป็นต้น

ตัวอย่างการแสดงลักษณะเฉพาะ

```
Panel_item panel_item;
panel_item = xv_create(panel,PANEL_CYCLE,
     XV_WIDTH,        50,
     XV_HEIGHT,       25,
     WIN_RIGHT_OF,  other_panel_item,
     PANEL_LABEL_X, 100,
     PANEL_LABEL_Y, 100,
     PANEL_LABEL_STRING,  "Open File",
     PANEL_CHOICE_STRING, "Overwrite contence",
                    NULL,
     NULL);
```

5.4.3 ชนิดของอ็อปเจค

ต่อไปนี้เป็นรายการอ็อปเจคที่เอ็กซ์วิวให้มาและรายละเอียดในการใช้งาน ทั้งการสร้าง การเซ็ตคุณสมบัติ ค่าปกติที่ตั้งให้ใช้(Default) และอื่นๆ รายการอ็อปเจคที่สามารถสร้างได้อยู่ในกรอบดังต่อไปนี้

-อ็อปเจคทั่วไป(Generic Objects)

-วินโดว์(Windows)

-เฟรม(Frame)

-เปิดวินโดว์(Openwins)

-แคนวัส(Canvases)

-เท็กซ์วินโดว์(Text Windows)

-เมนู(Menus)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-สกรอลล์บาร์(Scrollbars)

สำหรับรายละเอียดที่สมบูรณ์ โปรดดูในหนังสืออ้างอิง [4] หน้า 23 ถึง 38

5.5 การสร้างโปรแกรมประยุกต์ในเอ็กซ์วิว

ในส่วนนี้จะได้กล่าวถึงการเชื่อมต่อกับการโปรแกรมบนเอ็กซ์วิว พื้นฐานการกระจายงาน และการคอมไพล์และลิงค์(link)โปรกแกรม โครงสร้างการทำงานพอสรุปได้ดังนี้

-เริ่มสร้างเอ็กซ์วิวโดยใช้ xv_init()

-สร้างวินโดว์ระดับสูงสุด(Frame)เพื่อจัดการกับวินโดว์ย่อย

-เพิ่มวินโดว์ย่อยและลูกของเฟรม

-เพิ่มอ็อปเจคให้แก่วินโดว์ย่อย

-กำหนดการแจ้ง Callbacks และ เลือกเหตุการณ์อินพุท

-เรียก xv_main_loop() เพื่อเริ่มรับส่งเหตุการณ์

5.5.1 การเชื่อมต่อ

การคอมไพล์โปรแกรมทำได้ดังนี้

```
cc -g  -I/usr/home/openwin/include spw.c util.c parinda.c pum.c
   -L/usr/home/openwin/Xlib -lX11 -lxivew -ll -o spw.out
```

ในการคอมไพล์โปรแกรมเอ็กซ์วิว ต้องลิงค์ด้วยไลบรารี่ของเอ็กซ์วิว เพราะมันถูกสร้างไว้เหนือสิ่งอื่น ๆหรือมีความสำคัญมากกว่าอย่างอื่น ตัวโปรแกรมก็ต้องลิงค์เข้าด้วยกันด้วยดังตัวอย่างข้างบน ในการเขียนโปรแกรมต้องตรวจดูว่าจะใช้อ็อปเจคอะไรแล้วให้อินคลูดเฮดเดอร์ไฟล์อันนั้น อันดับแรกต้องอินคลูด xview.h ก่อน ดังตัวอย่าง

```
# include  <xivew/xview.h>
```

ถ้าหากมีการใช้อุปกรณ์ตัวอื่น ๆ ก็ต้องระบุเช่นกัน อย่างเช่น ถ้าในโปรแกรมใช้เฟรม,พาเนล,และฟอนต์ตัวอักษร ต้องอินคลูดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <xview/xivew.h>
#include <xview/frame.h>
#include <xview/panel.h>
#include <xview/font.h>
```

นอกจากนี้ต้องห้ามตั้งชื่อตามคำสงวน หอจะสรุปแบบคำสั่งได้ดังตารางที่ 5.3

attr_	icon_	
canvas_	menu_	server_
cursor_	notify_	server_image_
defaults_	panel_	termsw_
ei_	pixrect_	text_
es_	pr_	textsw_
ev_	pw_	tty_
event_	rect_	ttysw_
font_	rl_	win_
frame_	screen_	window_
fullscreen_	scrpll_	window_
generic_	scrollbar_	xv_

ตาราง 5.3 แสดงคำอุปสรรคที่สงวน

5.5.2 การเริ่มเอ็กซ์วิว

อันดับแรกสุดต้องเริ่มต้นระบบก่อน โดยใช้ฟังก์ชั่น xv_init() ซึ่งฟังก์ชันนี้จะทำงานเหล่านี้ให้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-เปิดเชื่อมต่อกับเซอร์ฟเวอร์

-ให้ค่าเริ่มต้นแก่ตัวแจ้งข่าวสาร

-ให้ค่าเริ่มต้นฐานข้อมูลของผู้จัดการทรัพยากร

รูปแบบของฟังก์ชั่น xv_init() เป็นดังนี้

```
void
xv_init(attrs)
    <attribute-value list> attrs;
```

5.5.3 การสร้างและการแก้ไขอ็อปเจค

สามารถใช้ฟังก์ชั่น xv_create();,xv_get();,xv_set();และ,xv_find(); โดยแต่ละฟังก์ชั่นมีการใช้ดังนี้

1. รูปแบบการใช้ xv_create()

```
Xv_object
Xv_create(owner,package,attrs)
    Xv_object   owner;
    Xv_pkg      package;
    <attribute-value list> attrs
```

ตัวอย่างการใช้ :-

```
#include <xview/xview.h>
main(argc,argv)
char *argv[];
{
    Frame  frame;
    xv_init(XV_INIT_ARGC_PTR_ARGV,&argc,argv,NULL)
    frame = (Frame)xv_create(NULL,FRAME,NULL);
    xv_main_loop(frame);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. รูปแบบของ xv_find()

```
Xv_opaque
xv_find (own,package,attrs)
    Xv_object   owner;
    Xv_pkg       package;
    <attribute-value list> attrs;
```

ตัวอย่างการใช้:-

```
Xv_font my_font;
my_font = xv_find(frame,FONT),
    FONT_NAME,  "fixed",
    NULL);
```

3. รูปแบบการใช้ xv_destroy()

```
int
 xv_destroy_safe(object)
    Xv_opaque  object;
```

แล้วต้องเรียนรู้ที่นรูปแบบเหมือนตัวอย่าง

```
void
quit()
{
    if (xv_destroy_safe(frame) == XV_OK)
    exit(0)
}
```

4. รูปแบบการใช้ xv_set();และ xv_get();

```
Xv_opaque
  xv_set (object,attrs)
  Xv_object   owner;
  <attribute-value list> attrs;
Xv_opaque
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าต้องการเลื่อนกลับมายังเซลล์ก่อนหน้านั้นให้เลื่อนเคอร์เซอร์เข้าไปในพื้นที่สีแดงแล้วกดปุ่มขวาสุดของเมาส์ โดยจะเคลื่อนที่ไปได้ที่ละเซลล์เช่นกัน

พื้นที่สีเทา

พื้นที่ตรงนี้กำหนดเพื่อวาดให้เห็นเป็นรูปร่างของสกอลบาร์และใช้ในการเลื่อนดูเซลล์ทีละหนึ่งหน้า โดยการทำงานเป็นดังนี้

ถ้าต้องการเลื่อนดูทางด้านหลักทีละหนึ่งหน้าให้เลื่อนเคอร์เซอร์เข้าไปในพื้นที่สีเทาตำแหน่งใดก็ได้ในบริเวณของสกอลบาร์แนวนอน ผลจะปรากฏดังนี้

เมื่อกดเมาส์ปุ่มซ้ายสุด เป็นการไปดูเซลล์ในกลุ่มต่อไป

เมื่อกดเมาส์ปุ่มขวาสุด เป็นการไปดูเซลล์ในกลุ่มที่ผ่านมา

ถ้าต้องการเลื่อนดูทางด้านแถวทีละหนึ่งหน้าให้เลื่อนเคอร์เซอร์เข้าไปในพื้นที่สีเทาตำแหน่งใดก็ได้ในบริเวณสกอลบาร์แนวดิ่ง ผลจะปรากฏดังนี้

เมื่อกดเมาส์ปุ่มซ้ายสุด เป็นการไปดูเซลล์ในกลุ่มต่อไป

เมื่อกดเมาส์ปุ่มขวาสุด เป็นการไปดูเซลล์ในกลุ่มที่ผ่านมา

5.6.7 <u>การรับข้อมูลคำสั่งจากผู้ใช้</u>

ในส่วนรับข้อมูลจากผู้ใช้จะสร้างเฟรมคำสั่ง(ซึ่งเป็นเฟรมลูก)มารับข้อมูล เมื่อเสร็จจากการรับข้อมูลแล้วเฟรมนี้ก็จะถูกปิดไป ทำให้บริเวณเวอร์คชีตทำงานดูสะอาด ลักษณะของเฟรมเป็นดังรูปที่ 6.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
xv_get (object,attrs)
Xv_object   owner;
<attribute-value list> attrs;
```

5. xv_main_loop()และตัวแจ้งให้ทราบ(Notifier)

เป็นอ็อปเจคตัวหนึ่งที่ต้องสร้างขึ้นมา ก่อนที่จะเรียกใช้ วินโดว์ต้องมีความพร้อมที่จุแสดงและมีเหตุการณ์ที่จะเริ่มแสดงผล งานสำคัญของ xv_main_loop() คือเป็นการเริ่มแก่ตัวแจ้งข่าวสาร พอตัวแจ้งให้ทราบเริ่มทำงาน โปรแกรมก็จะเริ่มรับและประมวลผลเหตุการณ์ต่าง ๆ โดยเหตุการณ์จะถูกส่งมาจากเอ็กซ์เซอร์ฟเวอร์โดยผ่านไคลแอนท์

5.6 งานประยุกต์ใช้เอ็กซ์วิวสร้างสเปรดชีต

เนื่องจากข้อดีของเอ็กซ์วินโดว์และเอ็กซ์วิวที่สามารถขยายความสามารถได้อย่างกว้างขวางโดยใช้ภาษาซี การโปรแกรมกับเอ็กซ์วิวทำได้ง่ายกว่าเพราะเป็นการโปรแกรมเชิงวัตถุที่สร้างมาช่วยในการพัฒนาระบบอินเตอร์เฟสโดยเฉพาะ ในส่วนต่อไปนี้จะได้กล่าวถึงรายละเอียดของการประยุกต์ใช้

5.6.1 ส่วนเชื่อมต่อผู้ใช้(User interface)

ส่วนเชื่อมต่อผู้ใช้ ที่สำคัญมากคือการจัดการหน้าจอ เนื่องจากว่าซอฟแวร์นี้สร้างบนระบบเอ็กซ์วินโดว์ ดังนั้นการจัดการหน้าจอจึงเป็นแบบวินโดว์ทั้งหมด ลักษณะการทำงานแบบวินโดว์ได้กล่าวมาแล้วจึงไม่ขอกล่าวอีก ข้อดีของวินโดว์มีมากมายหลายประการจึงช่วยให้ผู้ใช้สามารถใช้งานได้โดยง่ายและสะดวก ขนาดของวินโดว์สามารถยืดและขยายได้ บนวินโดว์ก็มีสัญลักษณ์ต่าง ๆ ที่สื่อความเข้าใจกับผู้ใช้ได้ง่ายและชัดเจน จึงส่งผลให้การทำงานกับโปรแกรมแบบวินโดว์เป็นที่นิยมใช้ทั่วไปในปัจจุบัน

5.6.2 เมนู(Menu)

เลือกใช้เมนูแบบ Pullright Menu หรือเรียกอีกอย่างหนึ่งว่า Cascading menus ลักษณะคือจะไม่ปรากฏให้ผู้ใช้เห็นขณะที่ไม่เรียกใช้งาน แต่ถ้าหากว่าผู้ใช้ต้องการใช่งานเมนู สามารถทำได้โดยเลื่อนเคอเซอร์เข้าไปในบริเวณพื้นที่การทำงาน ซึ่งเรียกว่า Canvas แล้วกดเมาส์ปุ่มทางด้านขวามือสุด เมนูก็จะปรากฏขึ้นทันที ทั้งนี้เมนูจะมีอยู่หลายระดับ ผู้ใช้ สามารถเข้าไปในแต่ละชั้นได้โดยการกดปุ่มทางด้านขวาสุดของเมาส์ไปเรื่อย ๆ ตรง menu item ที่มีสามเหลี่ยมวางกำกับอยู่ทางด้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขวา เมื่อผู้ใช้ต้องการเลือกทำคำสั่งใดก็สามารถเลือกได้โดยการกดปุ่มทางด้านซ้ายสุดของเมาส์(ปุ่มSelect) คำสั่งนั้นก็จะทำงานทันที หรืออาจกดปุ่มขวาสุดของเมาส์เลื่อนแถบสว่างขึ้นลงได้เพื่อเลือกทำงานตามที่ต้องการ การปรากฏของ Pullright menu ในครั้งแรกจะเหมือน POP UP menu คือเมื่อคลิกส์เมาส์แล้วเมนูจะปรากฏออกมาตรงที่เคอร์เซอร์อยู่ขณะนั้น เมื่อเมนูเปิดขึ้นมาแล้วการทำงานจะเหมือนพูลดาวน์เมนู(Pull-Down menu)

คำสั่งทำงาน (Operation)และรายละเอียดการทำงานของ Operation ต่าง ๆ ที่กำหนดใช้บนเมนูมีดังนี้

SPREADSHEET...

LOAD

ทำการดึงข้อมูลจากดิสก์ที่เก็บโดยคำสั่ง Save มาทำการแก้ไขต่อ การทำงานเป็นดังนี้

1. กดเมาส์ปุ่มขวาสุด
2. เลือกเมนูดังต่อไปนี้

   Spreadsheet/Load
3. เมื่อเลือกตามเมนูแล้วจะปรากฏวินโดว์ลูก(child window) เพื่อให้ใส่ค่าชื่อแฟ้มข้อมูลที่ต้องการโหลด

   3.1 ป้อนชื่อแฟ้มข้อมูล
4. เมื่อใส่ชื่อแฟ้มข้อมูลตามที่ต้องการแล้วให้กดคีย์ Enter หรือ ปุ่ม Load
5. ถ้าไม่มีแฟ้มข้อมูลตามที่ต้องการ โปรแกรมก็จะรายงานให้ทราบ

ถ้ามีไฟล์ที่ต้องการโหลดเก็บอยู่จริงๆ ก็จะทำการอ่านมา ถ้ามีคำว่า "Spreadsheet" เขียนอยู่ตรงหัว แสดงว่าเป็นไฟล์ของสเปรดชีตจริง ก็จะทำการอ่านมา โดยจะทำการอ่านค่าขนาดของสเปรดชีตแล้วทำการวนลูปอ่าน ค่าหลักของเซลล์ ค่าแถวของเซลล์ ขนาดของแต่เซลล์และข้อมูลในเซลล์ แล้วทำการตรวจดูว่าเป็นข้อมูลแบบไหน ก็ทำการเก็บข้อมูลนั้น ๆ ไปตามแบบของข้อมูล ทำการลดค่าของขนาดของสเปรดชีต โดยลบไปเท่ากับขนาดของแต่ละเซลล์อ่านมาเรื่อยๆ จนหมด โดยตรวจดูจากขนาดของสเปรดชีตเป็นศูนย์หรืออ่านค่าหลักได้น้อยกว่าศูนย์ จึงทำการปิดไฟล์

SAVE

ทำการบันทึกข้อมูลลงในดิสก์ หลังจากที่สร้างเวอร์คชีตแล้ว เมื่อต้องการจะสร้างเวอร์ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชีตใหม่ หรือต้องการป้องกันความผิดพลาดที่อาจเกิดจาก ใช้คำสั่งผิด หรือไฟดับ หรือเครื่องเกิดขัดข้องในขณะใช้งาน หรือต้องการเก็บข้อมูลไว้ใช้ครั้งต่อ ๆ ไป ทำได้ดังนี้

1. กดเมาส์ปุ่มขวาสุด
2. เลือกเมนูดังต่อไปนี้

   Spreadsheet/Save
3. เมื่อเลือกตามเมนูแล้วจะปรากฏวินโดว์ลูก(child window) เพื่อให้ใส่ค่าชื่อแฟ้มข้อมูลที่ต้องการเก็บ

   3.1 การป้อนชื่อแฟ้มข้อมูลให้ตั้งตามกฎการตั้งชื่อในระบบยูนิกซ์
4. เมื่อใส่ชื่อแฟ้มข้อมูลตามที่ต้องการแล้วให้กดคีย์ Enter หรือ ปุ่ม Save
5. หากมีแฟ้มข้อมูลอยู่ก่อนแล้วเครื่องจะรายงานว่ามีอยู่แล้วและถามว่าจะเก็บทับของเดิมหรือไม่ แล้วให้เลือกทำงานตามเมนูต่อไป

ถ้าหากยังเป็นไฟล์ใหม่หรือไฟล์เก่าที่ต้องการเขียนทับ ก็จะทำการเขียนคำว่า "Spreadsheet" ลงไปเพื่อที่จะได้รู้ว่าเป็นไฟล์ของสเปรดชีต และเขียนขนาดของสเปรดชีตลงไปโดยที่จะเป็นค่าอะไรก็ได้ค่าหนึ่ง จากนั้นก็เริ่มเก็บทีละเซลล์ ๆ โดยที่จะหาขนาดของแต่ละเซลล์ ถ้าเป็นตัวอักษรจะหาขนาดได้จาก ความยาวของตัวอักษรบวกกับค่าที่จะทำให้ความยาวหารด้วยสี่ลงตัวเพราะว่าเวิร์ดบาวเดอรีของเครื่องที่โปรแกรมนี้พัฒนาอยู่มีเวิร์ดบาวเดอรีเท่าสี่ และบวกเข้าไปอีกสี่ เพื่อที่จะเอาไว้เก็บค่าแอดทริบิวถ้าเป็นตัวเลขจะหาได้จากขนาดของดับเบิล(Double) บวกกับสี่ ถ้าเป็นสูตรคำนวณหาได้จากความยาวของสูตรบวกค่าที่ทำให้ความยาวสูตรหารด้วยสี่ลงตัวบวกกับขนาดของดับเบิลซึ่งใช้เก็บค่าที่ได้จากการคำนวณในสูตรนั้นบวกกับสี่จากนั้นก็เก็บค่า หลัก,แถว,ขนาดและค่าของแต่ละเซลล์ ซึ่งรวมทั้งแอดทริบิวโดยที่ขณะเก็บแต่ละเซลล์ จะทำการนับจำนวนไบท์ทั้งหมดโดยเอาขนาดของเซลล์ทุกเซลล์ที่เก็บมารวมกันจนกระทั่งหมดทุกเซลล์ จึงใช้คำสั่ง lseek หาตำแหน่งที่เขียนขนาดของสเปรดชีต แล้วเขียนจำนวนไบท์ลงไปแล้วจึงทำการปิดไฟล์

PRINT

ใช้พิมพ์ข้อมูลออกทางเครื่องพิมพ์

(ยังไม่มีโมดูลทำงาน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CLEAR

ทำการเคลียพื้นที่เวอร์คชีต โดยจะตรวจดูไปทีละเซลล์ว่าเซลล์ไหนมีข้อมูลอยู่ ถ้ามีก็จะทำการลบและฟรีหน่วยความจำตรงส่วนนั้นไป เมื่อหมดข้อมูลในกลุ่มไหนก็จะทำการฟรีในกลุ่มนั้นไปด้วย จนหมดทั้งเวอร์คชีต

ROW...

INSERT

เมื่อต้องการแทรกแถวในเวอร์คชีต ทำได้ดังนี้

1. กดเมาส์ปุ่มขวาสุด
2. เลือกเมนูดังต่อไปนี้

   Row/Insert
3. เมื่อเลือกตามเมนูแล้วจะปรากฏวินโดว์ลูก(child window) เพื่อให้ใส่ค่าคอลัมน์ที่ต้องการแทรก

   3.1 การป้อนแถวให้ป้อนเป็นตัวเลข 1 ถึง 2,000
4. เมื่อใส่ข้อมูลตามที่ต้องการแล้วให้กดคีย์ Enter หรือ ปุ่ม Insert

เมื่อรับค่าแถวที่ต้องการแทรกแล้ว มีอัลกอริทึมในการแทรกแถวดังนี้

ก่อนอื่นจะตรวจสอบดูก่อนว่า จะมีข้อมูลถึงหลักสุดท้ายหลักที่เท่าไร จากนั้นก็จะเริ่มสร้างกลุ่มข้อมูลเพิ่มขึ้นมาในแต่ละหลัก แล้วก็ทำการย้ายตัวชี้ที่เก็บไว้ในแต่ละเซลล์ลงไปในเซลล์แถวถัดไปเรื่อย ๆ จนถึงแถวที่ต้องการแทรกจึงหยุดย้าย แล้วก็ทำเช่นเดียวกันกับกลุ่มข้อมูลในหลักถัดไปจนกระทั่งถึงหลักสุดท้ายที่มีข้อมูลอยู่

DELETE

เมื่อต้องการแทรกแถวในเวอร์คชีต ทำได้ดังนี้

1. กดเมาส์ปุ่มขวาสุด
2. เลือกเมนูดังต่อไปนี้

   Row/Delete
3. เมื่อเลือกตามเมนูแล้วจะปรากฏวินโดว์ลูก(child window) เพื่อให้ใส่ค่าคอลัมน์ที่ต้องการแทรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 การป้อนแถวให้ป้อนเป็นตัวเลข 1 ถึง 2,000

4. เมื่อใส่ข้อมูลตามที่ต้องการแล้วให้กดคีย์ Enter หรือ ปุ่ม Delete

เมื่อรับค่าแถวที่ต้องการลบแล้ว มีอัลกอริทึมในการลบแถวดังนี้

ก่อนอื่นจะตรวจสอบดูก่อนว่า จะมีข้อมูลถึงหลักสุดท้ายหลักที่เท่าไร แล้วก็ทำการลบข้อมูลในเซลล์ที่อยู่ในแถวที่ต้องการลบออก จากนั้นก็เริ่มทำการย้ายข้อมูลจากเซลล์แถวล่างขึ้นมาเซลล์แถวบน จะหมดในกลุ่มข้อมูลกลุ่มแรก แล้วทำเช่นเดียวกันในกลุ่มข้อมูลในหลักถัดไป จนถึงหลักสุดท้ายที่มีข้อมูลอยู่

COLUMN...

INSERT

เมื่อต้องการแทรกคอลัมน์ในเวอร์คชีต ทำได้ดังนี้

1. กดเมาส์ปุ่มขวาสุด
2. เลือกเมนูดังต่อไปนี้

Column/Insert

3. เมื่อเลือกตามเมนูแล้วจะปรากฏวินโดว์ลูก(child window) เพื่อให้ใส่ค่าคอลัมน์ที่ต้องการแทรก

3.1 การป้อนคอลัมน์ให้ป้อนเป็นตัวอักษร A ถึง SG

4. เมื่อใส่ข้อมูลตามที่ต้องการแล้วให้กดคีย์ Enter หรือ ปุ่ม Insert

เมื่อรับค่าหลักที่ต้องการแทรกแล้ว มีอัลกอริทึมในการแทรกหลักดังนี้

ก่อนอื่นจะตรวจสอบดูก่อนว่า จะมีข้อมูลถึงแถวสุดท้ายแถวที่เท่าไร จากนั้นก็จะเริ่มสร้างกลุ่มข้อมูลเพิ่มขึ้นมาในแต่ละแถว แล้วก็ทำการย้ายตัวชี้ที่เก็บไว้ในแต่ละเซลล์ลงไปในเซลหลักถัดไปเรื่อย ๆ จนถึงหลักที่ต้องการแทรกจึงหยุดย้าย แล้วก็ทำเช่นเดียวกันกับกลุ่มข้อมูลในแถวถัดไปจนกระทั่งถึงแถวสุดท้ายที่มีข้อมูลอยู่

DELETE

เมื่อต้องการลบคอลัมน์ในเวอร์คชีต ทำได้ดังนี้

1. กดเมาส์ปุ่มขวาสุด
2. เลือกเมนูดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Column/Delete

3. เมื่อเลือกตามเมนูแล้วจะปรากฏวินโดว์ลูก(child window) เพื่อให้ใส่ค่าคอลัมน์ที่ต้องการลบ

3.1 การป้อนคอลัมน์ให้ป้อนเป็นตัวอักษร A ถึง SG

4. เมื่อใส่ข้อมูลตามที่ต้องการแล้วให้กดคีย์ Enter หรือ ปุ่ม Delete

เมื่อรับค่าหลักที่ต้องการลบแล้ว มีอัลกอริทึมในการลบหลักดังนี้

ก่อนอื่นจะตรวจสอบดูก่อนว่า จะมีข้อมูลถึงแถวสุดท้ายแถวที่เท่าไร แล้วก็ทำการลบข้อมูลในเซลที่อยู่ในหลักที่ต้องการลบออก จากนั้นก็เริ่มทำการย้ายข้อมูลจากเซลหลักล่างขึ้นมาเซลหลักบน จะหมดในกลุ่มข้อมูลกลุ่มแรก แล้วทำเช่นเดียวกันในกลุ่มข้อมูลในหลักถัดไป จนถึงแถวสุดท้ายที่มีข้อมูลอยู่

SETWIDTH

กรณีที่ความกว้างของคอลัมน์เดิมในเวอร์คชีตมีขนาดที่ไม่เหมาะสมกับการใช้งานเช่นไม่พอกับจำนวนตัวเลขหรือข้อความเราสามารถกำหนดใหม่ได้ดังนี้

1. กดเมาส์ปุ่มขวาสุด
2. เลือกเมนูดังต่อไปนี้

column/setwidth

3. เมื่อเลือกตามเมนูแล้วจะปรากฏวินโดว์ลูก(child window) เพื่อเลือกคอลัมน์และหนดความกว้างตามที่ต้องการ

3.1 การป้อนคอลัมน์ให้ป้อนเป็นตัวอักษร A ถึง SG

3.2 ความกว้างกำหนดได้มากที่สุดเท่ากับ 500x100 พิกเซล

4. เมื่อใส่ข้อมูลตามที่ต้องการแล้วให้กดคีย์ Enter หรือ ปุ่ม Setwidth

UTILITY...

RECALC

รายละเอียดการคำนวณสามารถกำหนดได้หลายกรณี โดยคำสั่งต่อไปนี้

1. กดเมาส์ปุ่มขวาสุด
2. เลือกเมนูดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Utility/Recalc

แล้วเข้าไปทำตามฟังก์ชั่นที่กำหนดให้มาดังนี้

Natural Columnwise Rowwise Automatic

Manual

Natural : กำหนดให้คำนวณตามลำดับความสัมพันธ์ของเครื่องหมายทางคณิตศาสตร์ ปกติจำคำนวณตามแบบนี้เสมอ ลำดับความสัมพันธ์มีดังนี้

1. ทำยกกำลัง (^)
2. ทำคูณหรือหาร (* หรือ /)
3. ทำบวกหรือลบ (+ หรือ -)
4. ถ้าเครื่องหมายอยู่ในลำดับเดียวกันจะทำจากซ้ายไปขวา
5. ถ้าต้องการให้ทำนิพจน์ใดก่อนก็ให้ใส่ไว้ในวงเล็บ อาจจะซ้อนกันหลาย ๆ วงเล็บก็ได้ โดยจะทำวงเล็บในสุดก่อน

Columnwise: กำหนดให้คำนวณตามแนวคอลัมน์ โดยเริ่มจากคอลัมน์แรกถึงคอลัมน์สุดท้าย และทำจากแถวข้างบนลงแถวข้างล่าง

Rowwise : กำหนดให้คำนวณตามแนวแถว โดยเริ่มจากแถวแรกถึงแถวสุดท้าย และทำจากคอลัมน์ข้างบนลงคอลัมน์ข้างล่าง

Automatic: กำหนดการคำนวณเป็นแบบอัตโนมัติ คือจะคำนวณค่าใหม่ทุกครั้งที่มีการเปลี่ยนแปลงค่าในเวอร์คชีต

Manual : กำหนดให้หยุดการคำนวณ เมื่อมีการเปลี่ยนแปลงค่าในเวอร์คชีตจนกว่าจะส่งสัญญาณรับรอง

4. ทำงานตามอัลกอริทึมที่เลือก

***หมายเหตุ ฟังก์ชั่นนี้ได้สร้างตัวเลือกไว้เพื่อรอการพัฒนาต่อไป ยังไม่มีการคำนวณจริง

GRAPH

แสดงความสามารถของวินโดว์ในการวาดกราฟ แต่การประยุกต์ใช้งานยังไม่มีโมดูลสนับสนุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

***หมายเหตุ ฟังก์ชั่นนี้ได้สร้างตัวเลือกไว้เพื่อรอการพัฒนาต่อไป ยังไม่สามารถแสดงกราฟได้จริง แต่ได้เขียนรูทีนในการวาดกราฟไว้เป็นตัวอย่างแล้วทุกรูปแบบ

FILL DATA

ในกรณีที่ต้องการป้อนชุดของตัวเลข ที่เรียงกันในลักษณะแบบอนุกรมจากน้อยไปหามาก หรือมากไปหาน้อย โดยไม่ต้องพิมพ์เองทีละเซลล์ ก็สามารถใช้คำสั่ง Fill dataมาช่วยได้ดังนี้

1. กดเมาส์ปุ่มขวาสุด
2. เลือกเมนูดังต่อไปนี้

   Utility/Fill data
3. เมื่อเลือกตามเมนูแล้วจะปรากฏวินโดว์ลูก(child window) เพื่อเลือกให้ใส่ข้อมูลที่ต้องการป้อนให้แก่คำสั่งฟิลข้อมูล สิ่งที่ต้องป้อนมีดังนี้

   3.1 เซลล์เริ่มต้น ตัวอย่างเช่น A1

   3.2 เซลล์สุดท้ายในการฟิล เช่น B100

   3.3 ค่าเริ่มต้น

   3.4 ค่าสุดท้าย

   3.5 ค่าเพิ่มใหม่ในแต่ละเซลล์(step)
4. เมื่อใส่ข้อมูลตามที่ต้องการแล้วให้กดคีย์ Enter หรือ ปุ่ม Fill data

เมื่อรับค่าแถวและหลักเริ่มต้น และแถวและหลักสุดท้ายที่ต้องการเติมข้อมูล, ค่าเริ่มต้น, ค่าสุดท้าย, และสเต็ปของการเพิ่มหรือลดข้อมูล แล้วเริ่มทำการเติมข้อมูลจากเซลล์แรก ในแถวแรกไปเรื่อยๆ จนถึงแถวสุดท้าย แล้วเริ่มในแถวแรกในหลักถัดไป จนกระทั่งถึงค่าสุดท้ายที่ต้องการ หรือถึง แถว, หลักสุดท้ายที่ต้องการเติม

AUTOCALC...

ปกติไม่ต้องสั่ง การคำนวณจะเป็นแบบนี้เสมอ คือถ้ามีการเปลี่ยนค่าใดค่าหนึ่งในเวอร์คชีต เซลล์ต่าง ๆ ที่มีสูตรเกี่ยวกับเซลล์ค่าที่เปลี่ยนไปนี้ จะถูกคำนวณค่าให้ใหม่ทันที

***หมายเหตุ ฟังก์ชั่นนี้ได้สร้างตัวเลือกไว้เพื่อรอการพัฒนาต่อไป ยังไม่มีการคำนวณจริง

COPY...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นคำสั่งในการเคลื่อนย้ายข้อมูลในเวอร์คชีต โดยข้อมูลในตำแหน่งต้นจะไม่ถูกเปลี่ยนแปลงแต่อย่างใด การทำงานเป็นดังนี้

1. กดเมาส์ปุ่มขวาสุด
2. เลือกเมนูดังต่อไปนี้

   Copy
3. เมื่อเลือกตามเมนูแล้วจะปรากฏวินโดว์ลูก(child window) เพื่อเลือกให้ใส่ข้อมูลที่ต้องการป้อนให้แก่คำสั่งคัดลอกข้อมูล สิ่งที่ต้องป้อนมีดังนี้

   3.1 เซลล์เริ่มต้น ตัวอย่างเช่น A1

   3.2 เซลล์สุดท้าย เช่น B100

   3.3 ตำแหน่งเซลล์ที่ต้องการย้ายข้อมูลไปเก็บ เช่น G20
4. เมื่อใส่ข้อมูลตามที่ต้องการแล้วให้กดคีย์ Enter หรือ ปุ่ม Copy

เมื่อรับค่า แถว, หลักเริ่มต้น และแถว, หลักสุดท้ายที่จะทำการคัดลอก และแถว, หลักแรกที่จะคัดลอกข้อมูลไปแล้ว ถ้าแถว, หลักที่จะคัดลอกไปไม่อยู่ในช่วงของแหล่งที่จะคัดลอก ก็จะเริ่มทำการคัดลอกข้อมูลเซลล์ต่อเซลล์ไปเรื่อยๆ จนหมด โดยไม่ลบข้อมูลที่แหล่งเดิม

MOVE...

เป็นคำสั่งในการเคลื่อนย้ายข้อมูลในเวอร์คชีต โดยข้อมูลในตำแหน่งต้นจะถูกเคลียการทำงานเป็นดังนี้

1. กดเมาส์ปุ่มขวาสุด
2. เลือกเมนูดังต่อไปนี้

   Move
3. เมื่อเลือกตามเมนูแล้วจะปรากฏวินโดว์ลูก(child window) เพื่อเลือกให้ใส่ข้อมูลที่ต้องการป้อนให้แก่คำสั่งคัดลอกข้อมูล สิ่งที่ต้องป้อนมีดังนี้

   3.1 เซลล์เริ่มต้น ตัวอย่างเช่น A1

   3.2 เซลล์สุดท้าย เช่น B100

   3.3 ตำแหน่งเซลล์ที่ต้องการย้ายข้อมูลไปเก็บ เช่น G20
4. เมื่อใส่ข้อมูลตามที่ต้องการแล้วให้กดคีย์ Enter หรือ ปุ่ม Move

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อรับค่า แถว, หลักเริ่มต้น และแถว, หลักสุดท้ายที่จะทำการย้าย และแถว, หลักแรกที่จะย้ายข้อมูลไปแล้ว ถ้าแถว, หลักที่จะย้ายไปไม่อยู่ในช่วงของแหล่งที่จะย้าย ก็จะเริ่มทำการย้ายข้อมูลเซลต่อเซลล์ไปเรื่อยๆ จนหมด โดยลบข้อมูลที่แหล่งเดิม

QUIT...

หากต้องการเลิกใช้งาน ทำได้โดยเลือกคำสั่ง QUIT ในเมนูที่มีไว้ให้แล้วหรือเอาเมาส์ไปคลิ๊กในบริเวณหัวของวินโดว์แล้วกดเมาส์ปุ่มซ้ายสุด เมนูของวินโดว์ก็จะประกฏออกมาให้เห็น ผู้ใช้สามารถเลือกคำสั่ง QUIT ทำงานได้โดยกดปุ่ม Select บนเมนู ท่านก็จะออกจากชีตและวินโดว์ได้ ก่อนที่ท่านจะออกจากชีต ระบบจะถามท่านว่าจะออกจริงหรือไม่ ถ้าหากยังไม่มีการเก็บข้อมูลเข้าแฟ้มข้อมูล ระบบก็จะรายงานว่ายังไม่มีการเก็บข้อมูลให้ผู้ใช้ได้ทราบ ถ้าหากผู้ใช้ไม่ต้องการเก็บข้อมูล ก็ออกได้เลยโดยเลื่อนเมาส์ไปคลิ๊กปุ่ม NO แต่ถ้าต้องการเก็บก็ใช้เมาส์คลิ๊กที่ปุ่ม YES ก็จะเข้าไปที่ชีตทำงานอีกครั้งเพื่อให้ทำการเก็บข้อมูลก่อนออกจากการทำงาน

หรือ

1. กดเมาส์ปุ่มขวาสุด
2. เลือกเมนูดังต่อไปนี้

   Quit
3. เลือกการทำงานตามเฟรมคำสั่ง (Command frame)

5.6.3 <u>Tools ที่นำมาช่วยพัฒนาเมนู</u>

ได้นำเอ็กซ์วิว (XVIEW: X window System based Visual/Integrate Environment for workstation) ซึ่งเป็นเครื่องมือช่วยทำอินเตอร์เฟส(user interface toolkit) สนันสนุนการทำงานแบบโต้ตอบ (Interactive) รวมทั้งโปรแกรมประยุกต์ทางด้านกราฟฟิค (Graphics-based application) ที่รันบนระบบเอ็กซ์วินโดว์ ทำให้ผู้เขียนโปรแกรมสามารถพัฒนาโปรแกรมในส่วนของการอินเตอร์เฟสได้รวดเร็วและมีประสิทธิภาพ แต่ทั้งนี้ทั้งนั้นเอ็กซ์วิวมีเนื้อหาและรายละเอียดที่ต้องศึกษามาก โดยต้องมีความรู้พื้นฐานของวินโดว์และไลบรารี่ (Xlib) มาก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.6.4 ส่วนของชีตและการแสดงเซลล์

ชีตและการแสดงเซลล์ปรากฏอยู่บนพื้นที่ทำงานที่เรียกว่า Canvas ขณะนี้สร้างโปรโตไทป์ขั้นทดสอบการทำงาน ประกอบด้วยจำนวนแถว 2,000 แถว จำนวนคอลัมน์ 500 คอลัมน์ ดังนั้นจึงมีเซลล์ให้ใช้ได้ทั้งหมด 10,000 เซลล์

ในหนึ่งหน้าจอของวินโดว์จะปรากฏจำนวนแถว และคอลัมน์ ไม่แน่นอนขึ้นอยู่กับความกว้างและความยาวของวินโดว์และความกว้างของเซลล์ที่ผู้ใช้กำหนดใช้งาน จะเห็นได้ว่าทุกอย่างเป็นไดนามิกมาก จึงมีความสะดวกในการใช้งาน นอกจากนี้แล้วผู้ใช้ยังสามารถใช้ Scrollbar เลื่อนดูได้ตลอดทั้งชีต

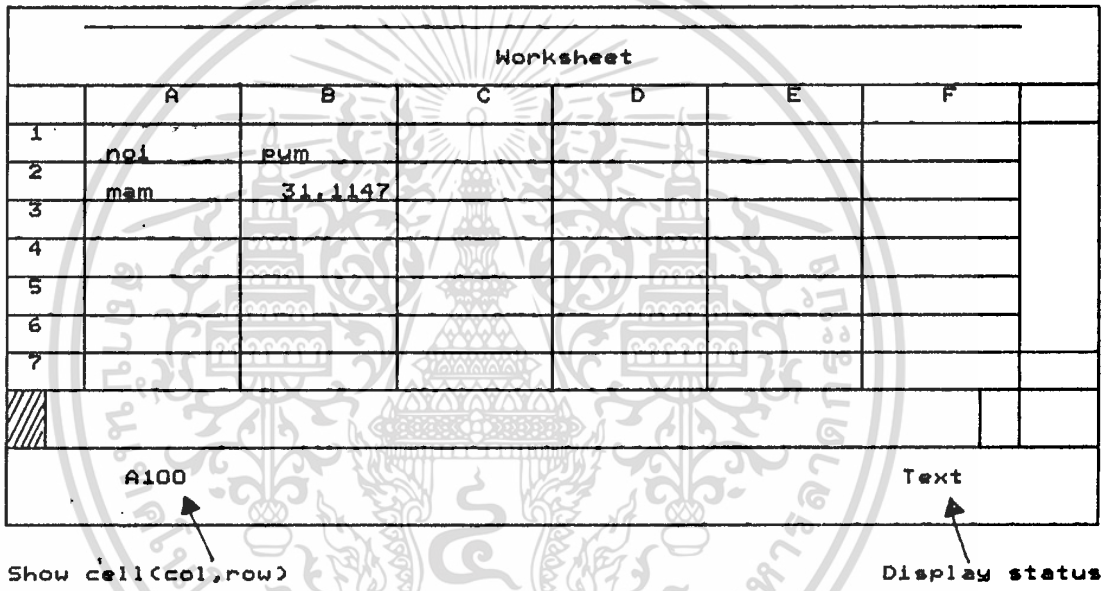
การป้อนข้อมูลลงไปในเซลล์ ทำได้โดยตรงขณะที่เซลล์นั้นทำงานอยู่(Active)ขณะใดขณะหนึ่งจะมีเพียงเซลล์เดียวเท่านั้นที่ทำงาน ผู้ใช้สามารถเลือกเซลล์มาทำงานได้โดยเลื่อนเมาส์ไปคลิกส์ที่เซลล์นั้นแล้วป้อนข้อมูลเข้าไปเก็บในเซลล์ได้ทันที การป้อนเข้าไปเก็บผู้ใช้ต้องกดคีย์ Enter เพื่อรับรองการเก็บทุกครั้ง ถ้าไม่กดคีย์ Enter รับรองก็จะไม่มีการเก็บข้อมูลใด ๆ หากต้องการลบข้อมูลในเซลล์ก็สามารถทำได้ในสองกรณีด้วยกันคือถ้ายังไม่มีการกดคีย์ Enter ให้ใช้คีย์ Back Space เพื่อลบข้อมูลที่ไม่ต้องการแล้วให้พิมพ์ได้ต่อไป แต่ถ้าได้กดคีย์ Enter รับรองแล้วให้เอาเมาส์ไปคลิกที่เซลล์นั้นให้ทำงานแล้วสามารถพิมพ์ข้อมูลใหม่ทับข้อมูลเก่าได้ทันที

ในเวอร์คชีตอ้างอิงเซลล์ ด้วยค่าของแถว(Row)และหลัก (Column) จุดตัดของ แถวและหลักจะแสดงตำแหน่งของเซลล์ โดยเวอร์คชีตนี้อ้างอิงเซลล์ได้ทั้งหมดเท่ากับ 1,000,000 เซลล์ในขณะใดขณะหนึ่งจะมีเซลล์ที่ทำงาน(active) อยู่เพียงเซลล์เดียว ผู้ใช้จะเห็นได้จากมี 4 เหลี่ยมสีแดงตีกรอบเอาไว้รอบเซลล์ ขณะที่เซลล์ทำงาน สามารถป้อนข้อมูลเข้าเซลล์ ได้ทันที ภายใน 1 เซลล์ผู้ใช้สามารถป้อนข้อมูลได้ไม่เกิน80 ตัวอักษร ทั้งนี้ผู้ใช้สามารถกำหนดความกว้างของเซลล์ได้ตามต้องการ ตำแหน่งของเซลล์ ปัจจุบันจะถูกรายงานอยู่ที่ตำแหน่ง FOOTNOTE ด้านล่างซ้ายของวินโดว์ ส่วน FOOTNOTE ทางด้านขวาจะรายงานสถานะการทำงานของเวอร์คชีต เช่นสถานะของเซลล์ (Empty,Text,Value,Formula) และความผิดพลาดต่าง ๆ ที่เกิดขึ้นขณะใช้งาน

การป้อนข้อมูลเข้าเซลล์ จะป้อนข้อมูลได้ก็ต่อเมื่อเซลล์นั้น ๆ ทำงานอยู่เท่านั้น ผู้ใช้สามารถป้อนข้อมูลเข้าได้ตามต้องการ เมื่อป้อนข้อมูลเสร็จแล้ว ให้กดคีย์ Enter เพื่อเป็นการรับรองว่าได้ส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้าไปเก็บไว้ในเซลล์แล้ว หากไม่กด Enter ข้อมูลจะไม่ถูกนำไปเก็บในเซลล์ และจะหายไปในที่สุด เมื่อรับค่ามาแล้วก็จะนำมาแยกว่าข้อมูลนั้นเป็นตัวอักษร ตัวเลข หรือสูตรคำนวณ เพื่อที่จะได้นำไปเก็บตามรูปแบบนั้นหากต้องการกลับไปแก้ไขข้อมูลในเซลล์ที่ผ่านมา ทำได้โดยเลื่อนเคอร์เซอร์ไปในบริเวณเซลล์นั้น ๆ แล้วคลิ๊กให้เซลล์นั้น ๆ ทำงาน แล้วป้อนข้อมูลเข้าตามปกติ แต่อย่าลืมว่าต้องกดคีย์ Enter เพื่อรับรองอีกครั้ง



รูปที่ 5.1 ลักษณะของเซลล์ที่ปรากฏบนวินโดว์

5.6.5 <u>การแสดงสี</u>

บนระบบวินโดว์สามารถ Allocate สีได้มากมายและมีความละเอียดสูงแต่ในซอฟต์แวร์ชุดนี้กำหนดให้มีเพียง 15 สีซึ่งน่าพอสำหรับการใช้งาน สีต่าง ๆ ที่ปรากฏให้เห็นเป็นการเซ็ตมาจากข้างในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามปกติแล้วผู้ใช้ควรจะเซ็ตสีได้แบบทันทีทันใด แต่ที่ไม่สนับสนุนอย่างนั้นเพราะต้องการให้การใช้เป็นไปโดยง่ายและผลตอบสนองการทำงานที่รวดเร็ว

5.6.6 การใช้สกอลบาร์

โดยปกติ บนวินโดว์จะมีสกอลบาร์ให้ใช้ได้ แต่ลักษณะการทำงานค่อนข้างจำกัดนั่นคือสกอลบาร์เลื่อนดูได้เฉพาะบริเวณแคนวัส(Canvas) ที่ใหญ่กว่าวินโดว์เท่านั้น และถ้าติดตั้งใช้กับ Text Subwindow ก็จะทำงานได้ตามข้อกำหนดคือปุ่มลิฟต์(บนสกอลบาร์)จะเลื่อนทีละหนึ่งบรรทัดหากแต่มีเฉพาะสกอลบาร์แนวดิ่งเท่านั้น นอกจากนี้แล้วสกอลบาร์จะไม่ทำงานแม้ว่าจะได้ติดตั้งเอาไว้ก็ตาม เมื่อดูลักษณะดังกล่าวแล้วจะเห็นได้ว่าการทำงานไม่เหมาะกับการประยุกต์ใช้งานของสเปรดชีต ดังนั้นจึงไม่เอามาติดตั้งใช้งาน สกอลบาร์ที่ใช้ในงานประยุกต์นี้จึงต้องกำหนดเอง การทำงานต่าง ๆ สามารถกำหนดใช้เองโดยไม่ขึ้นอยู่กับการทำงานของระบบสกอลบาร์ของวินโดว์แต่อย่างใด

ผู้ใช้สามารถใช้สกอลบาร์เพื่อเลื่อนดูข้อมูลในเซลล์ได้ตลอดทั้งชีตทำงาน โดยประกอบไปด้วยสกอลบาร์ในแนวดิ่ง(vertical scrollbar) และสกอลบาร์ในแนวนอน (horizontal scrollbar) การทำงานมีรายละเอียดดังต่อไปนี้คือ

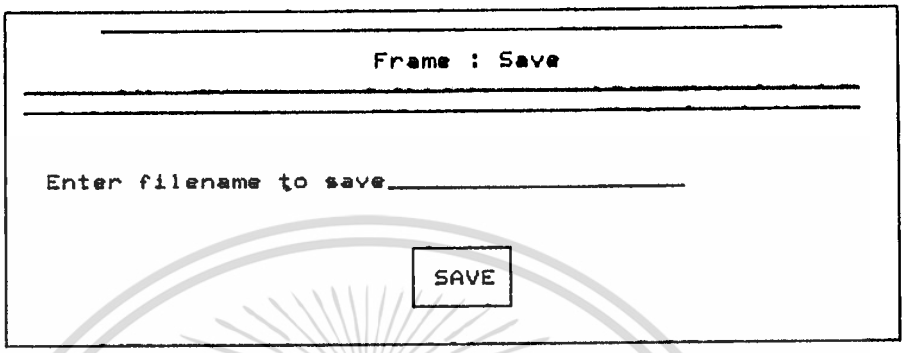
ปุ่มสีน้ำเงิน

ปุ่มสีน้ำเงินด้านบนของ vertical scrollbar ใช้เมื่อต้องการไปแถวแรก
ปุ่มสีน้ำเงินด้านล่างของ vertical scrollbar ใช้เมื่อต้องการไปแถวสุดท้าย
ปุ่มสีน้ำเงินด้านบนของ horizontal scrollbar ใช้เมื่อต้องการไปหลักแรก
ปุ่มสีน้ำเงินด้านล่างของ horizontal scrollbar ใช้เมื่อต้องการไปหลักสุดท้าย

ปุ่มสีแดง

ถ้าต้องการเลื่อนไปข้างหน้าให้เลื่อนเคอร์เซอร์เข้าไปในพื้นที่สีแดงแล้วกดปุ่มซ้ายสุดของเมาส์ โดยจะเคลื่อนที่ไปได้ทีละเซลล์เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 แสดงเฟรมคำสั่ง

5.6.8 ออกจากการทำงาน

หากต้องการเลิกใช้งาน ทำได้โดยเลือกคำสั่ง Quit ในเมนูที่มีไว้ให้แล้วหรือเอาเมาส์ไปคลิ๊กในบริเวณหัวของวินโดว์แล้วกดเมาส์ปุ่มซ้ายสุด เมนูของวินโดว์ก็จะประกฏออกมาให้เห็น ผู้ใช้สามารถเลือกคำสั่ง Quit ทำงานได้โดยกดปุ่ม Select บนเมนู ท่านก็จะออกจากชีดและวินโดว์ได้ ก่อนที่ท่านจะออกจากชีด ระบบจะถามท่านว่าจะออกจริงหรือไม่ ถ้าหากยังไม่มีการเก็บข้อมูลเข้าแฟ้มข้อมูล ระบบก็จะรายงานว่ายังไม่มีการเก็บข้อมูลให้ผู้ใช้ได้ทราบ ถ้าหากผู้ใช้ไม่ต้องการเก็บข้อมูล ก็ออกได้เลยโดยเลื่อนเมาส์ไปคลิ๊กปุ่ม NO แต่ถ้าต้องการเก็บก็ใช้เมาส์คลิ๊กที่ปุ่ม YES ก็จะเข้าไปที่ชีดทำงานอีกครั้งเพื่อให้ทำการเก็บข้อมูลก่อนออกจากการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

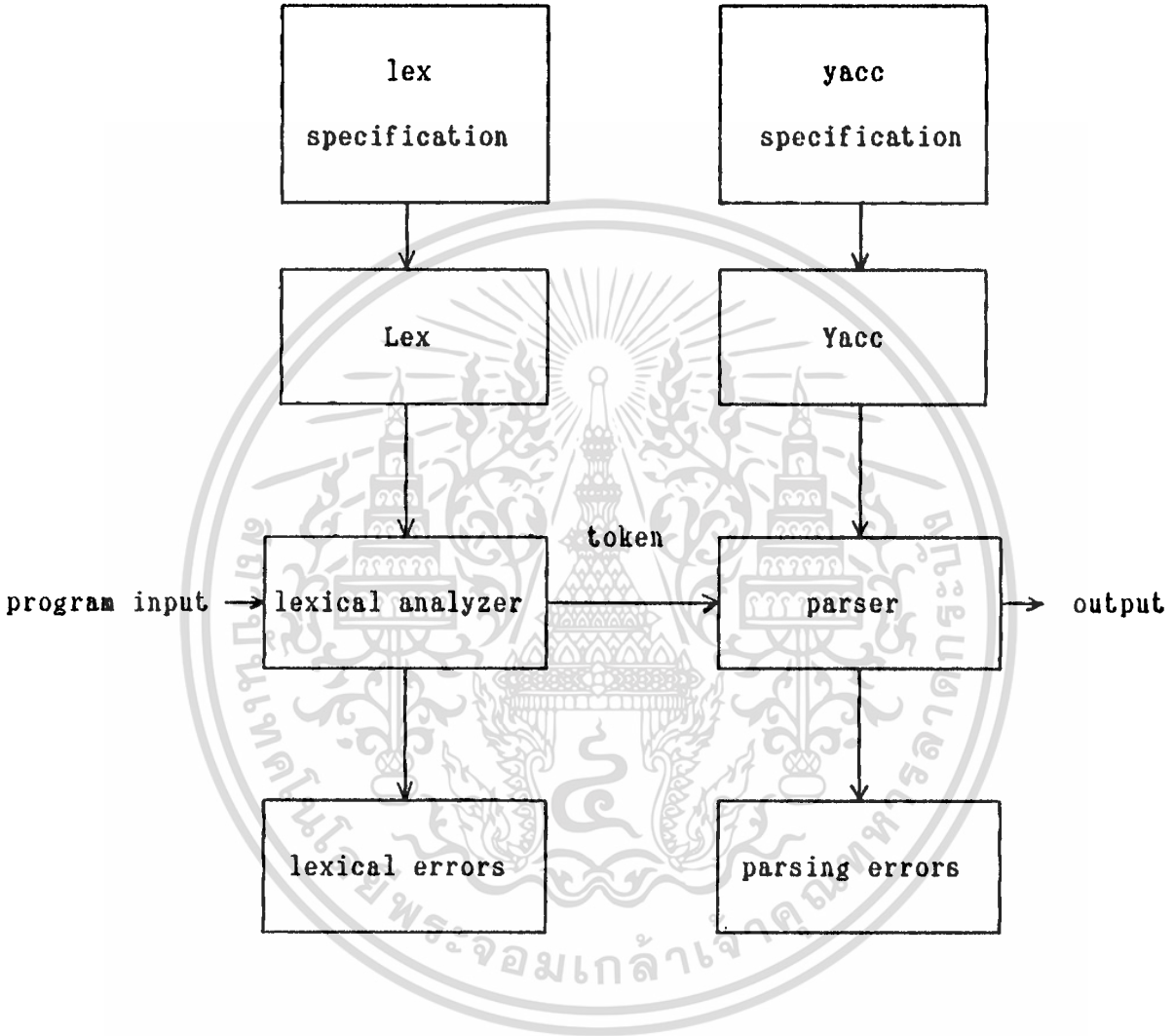
บทที่ 6
ส่วนรับข้อมูล

การเขียนโปรแกรมเพื่อสร้างส่วนรับอินพุทจากผู้ใช้ ใช้หลักการของคอมไพล์เลอร์ โดยใช้ Lex และ Yacc ซึ่งเป็นซอฟต์แวร์ช่วยการตรวจเช็คไวยากรณ์ของภาษาในระบบยูนิกซ์ ช่วยในการวิเคราะห์คำและแยกคำออกมาตีความหมาย ซึ่งปกติเราจะพบการประยุกต์ใช้งานของทั้งสองโปรแกรมนี้ในลักษณะงานคอมไพล์เลอร์เป็นส่วนมาก

ส่วนแรกของคอมไพล์เลอร์ที่ทำหน้าที่ในการตรวจสอบภาษา คือ ฟรอนท์เอนด์คอมไพล์เลอร์ (front end compiler) ซึ่งประกอบไปด้วยส่วนสำคัญสองส่วน ส่วนการวิเคราะห์คำ (lexical analysis) และการวิเคราะห์ไวยากรณ์ (syntax analysis) ส่วนการวิเคราะห์คำทำหน้าที่ในการวิเคราะห์อักขระประเภทต่าง ๆ เพื่อแยกออกเป็นกลุ่มคำ เช่น สูตร ตัวแปร เครื่องหมายทางคณิตศาสตร์ เป็นต้น ส่วนการวิเคราะห์ไวยากรณ์ ทำหน้าที่ในการวิเคราะห์ความสัมพันธ์ระหว่างคำว่าถูกต้องตามกฎไวยากรณ์ที่กำหนดไว้หรือไม่ สามารถบอกข้อผิดพลาดได้หากไวยากรณ์ของภาษาไม่เป็นไปตามที่กำหนด โปรแกรมส่วนที่เป็นตัววิเคราะห์คำ มักจะเรียกอีกอย่างหนึ่งว่า สแกนเนอร์ (scanner) และโปรแกรมส่วนที่เป็นตัววิเคราะห์ไวยากรณ์ มักจะเรียกอีกอย่างหนึ่งว่า ตัวแยกคำ (parser) การทำงานของทั้งสองส่วนนี้ มีความสัมพันธ์ดังรูป

ในรูปแบบการใช้ซอฟต์แวร์ Lex และ Yacc นั้น Lex จะรับเอานิพจน์ปกติ (regular expression) และการโปรแกรมแบบภาษาซี เพื่อทำการตรวจสอบดูว่า ข้อมูลที่ส่งเข้ามานั้น มีไวยากรณ์ตรงกับส่วนที่ระบุไว้ในนิพจน์ปกติหรือไม่ และปฏิบัติการทำงานตามที่ระบุไว้ในโปรแกรมภาษาซี โดยปกติแล้ว Lex กับ Yacc จะทำงานร่วมกัน โดย Yacc จะรับกฎไวยากรณ์พร้อมทั้งโปรแกรมภาษาซีที่กำหนดรูปแบบการตอบสนองต่อการตรงกับกฎไวยากรณ์นั้น ๆ หรือกล่าวอีกนัยหนึ่งว่า Lex จะทำงานในส่วนการวิเคราะห์คำ และ Yacc จะทำงานในส่วนวิเคราะห์ไวยากรณ์หรือการแยกคำ ต่อไปนี้จะกล่าวถึงความรู้ทั่ว ๆ ไปเกี่ยวกับขั้นตอนการวิเคราะห์คำ และขั้นตอนการวิเคราะห์กฎไวยากรณ์ รวมไปถึงการนำเอาความรู้ส่วนการวิเคราะห์คำ และการวิเคราะห์ไวยากรณ์ ไปประยุกต์ใช้ในการสร้างส่วนวิเคราะห์คำสิ่งของสเปรดชีต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.1 แสดงการกระจายค่าโดยใช้ Lex และ Yacc

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1 ตัววิเคราะห์คำ (Lexical Analyzer)

ตัววิเคราะห์คำเป็นขั้นตอนแรกของการประมวลผลเพื่อวิเคราะห์คำ หน้าที่หลักคืออ่านอักขระทีละตัวจากโปรแกรมต้นแบบ หรือจากอินพุทที่รับเข้ามา แล้วสร้างโทเคน (token) เพื่อส่งต่อให้กับขั้นตอนการวิเคราะห์ไวยากรณ์ และจะเก็บค่าที่จำเป็นต้องใช้ในขั้นตอนอื่น ๆ ไว้ในตารางสัญลักษณ์

ในส่วนของการสร้างตัววิเคราะห์คำ มักจะกล่าวถึงลักษณะของสิ่งเหล่านี้ คือ

โทเคน (Token) เป็นสิ่งที่ตัววิเคราะห์คำสร้างขึ้นมาจากอินพุทหรือโปรแกรมต้นแบบ เพื่อแยกคำแต่ละคำ ส่งให้ตัววิเคราะห์ไวยากรณ์ตีความหมายและหน้าที่ของคำเหล่านั้นต่อไป

รูปแบบ (Pattern) คือข้อกำหนดที่อธิบายความเป็นโทเคน โดยจะบอกถึงลักษณะของอักษรหรือกลุ่มอักขระที่จะจัดเป็นโทเคนแต่ละตัว

เลกซ์ซีม (Lexemes) คือชุดของตัวอักขระที่ติดต่อกันในลำดับอินพุท ซึ่งตรงตามรูปแบบที่กำหนดไว้

6.1.1 การกำหนดโทเคน

การอธิบายรูปแบบของโทเคน ได้ใช้รูปแบบการกำหนดเป็น นิพจน์ปกติโดยแต่ละรูปแบบจะเป็นการอธิบายถึงเซตของกลุ่มอักษรเพียงเซตเดียว ดังนั้น แต่ละนิพจน์ จึงเป็นเหมือนชื่อเฉพาะของแต่ละเซตของอักษร เช่น

```
INT     [0-9]
ALPHA   [A-Za-z]
%%
{INT}       { printf("Integer..."); }
{ALPHA}     { printf("Character..."); }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โทเคนหนึ่ง ๆ จะมีส่วนประกอบ 2 ส่วนคือ

1. ประเภท (Type) เพื่อบอกประเภทของโทเคน เช่น เป็นตัวอักษร เป็นตัวเลข

2. ค่า (Value) เพื่อบอกค่าของโทเคน ในกรณีที่โทเคนนั้นมีค่า เช่น โทเคนที่เป็นตัวเลข จะมีค่าเท่ากับค่าของตัวเลขนั้น

6.1.2 การทำงานในขั้นตอนการวิเคราะห์คำ

1. ตัววิเคราะห์คำจะทำการประมวลผลข้อความจากอินพุททีละตัวอักษร และพิจารณาว่าแต่ละตัวอักษรนั้นเป็นโทเคนหรือไม่ เช่น C12, +, - เป็นต้น ถ้าไม่ จะสามารถรวมกับตัวอักษรข้างเคียงเพื่อให้เป็นโทเคน ได้อย่างไร

2. กำหนดค่าของโทเคน ถ้าสามารถทำได้ และแยกหรือกำจัดคำที่ไม่เกี่ยวข้องหรือไม่ต้องการออกไป เช่น ตัวว่าง คำอธิบาย เป็นต้น

3. แจ้งความผิดพลาดของตัวอักษรหรือกลุ่มตัวอักษรที่ไม่สามารถจัดเป็นโทเคนได้

6.1.3 รูปแบบของโปรแกรม LEX

การเขียนโปรแกรมเพื่อสร้างตัววิเคราะห์คำ โดยการใช้โปรแกรม Lex ซึ่งเป็นเครื่องมือสำหรับการสร้างตัววิเคราะห์คำบนระบบยูนิกซ์ มีรูปแบบดังต่อไปนี้

โปรแกรมของ Lex แบ่งออกเป็น 3 ส่วน โดยมีเครื่องหมาย %% คั่น คือ

```
Declarations
%%
Translation rules
%%
Auxiliary procedures
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนแรกคือส่วนที่อยู่ก่อนเครื่องหมาย %% ตัวแรก จะเป็นการกำหนดอักขระในภาษาต้นแบบ โดยจะกำหนดด้วยคำจำกัดความปกติ (Regular Definition) และสามารถเขียนการทำงานบางอย่างเป็นภาษาซีได้ โดยมีรูปแบบการเขียน คือ

```
%{
    C Programming
%}
regular definitions
```

Lex จะทำการคัดลอกข้อความทั้งหมดที่อยู่ระหว่างเครื่องหมาย %{ และ %} ลงโปรแกรมซึ่งสร้างขึ้นมา สำหรับส่วนที่สองจะเป็นการกำหนดการกระทำของตัววิเคราะห์คำ เมื่อพบสตริงที่มีรูปแบบตามที่กำหนดไว้ โดยมีรูปแบบการกำหนดคือ

```
P1    { action1 }
P2    { action2 }
```

ซึ่งแต่ละ P คือ นิพจน์ปกติ และแต่ละการกระทำคือส่วนของการระบุการทำงานเมื่อพบสตริงตามนิพจน์นั้น ๆ เขียนเป็นลักษณะของโปรแกรมภาษาซี Lex จะทำการคัดลอกลงไว้บนโปรแกรมตัววิเคราะห์คำ ซึ่งสร้างขึ้นโดยเป็นการกระทำในเงื่อนไขเมื่อพบสตริงดังกล่าว (ต่างจากส่วนแรกที่คัดลอกไปโดยตรง)

การคอมไพล์โดย Lex จะได้โปรแกรมภาษาซี ชื่อ lex.yy.c ซึ่งจะถูกนำไปรวมเป็นไฟล์อินคลูด (include file) โดยโปรแกรมหลักอีกทีหนึ่ง รูปแบบของการคอมไพล์โปรแกรมโดย Lex เป็นดังนี้

```
cc -o [outputfile name] [inputfile name] -ll
```

-ll เป็นการระบุว่า การคอมไพล์โปรแกรมจาก Lex คอมไพเลอร์จะลิงค์ฟังก์ชันบางฟังก์ชันให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 ตัววิเคราะห์ไวยากรณ์ (Syntax Analyzer)

ตัววิเคราะห์ไวยากรณ์จะทำการรับโทเคนซึ่งก็คือสัญลักษณ์เทอร์มินัล จากตัววิเคราะห์คำแล้วนำมาตรวจสอบโทเคนเหล่านั้น ว่าเป็นรูปแบบที่ถูกต้องตามกฎไวยากรณ์ที่กำหนดโดยส่วนตัววิเคราะห์ไวยากรณ์นั้นหรือไม่ กรณีที่โทเคนไม่เรียงกันตามรูปแบบที่กำหนด ตัวแยกคำจะต้องสามารถตรวจจับความผิดพลาด และดำเนินการต่อให้ได้ เพื่อจะได้ตรวจจับความผิดพลาดครั้งต่อไปที่อาจจะเกิดขึ้นอีก

6.2.1 รูปแบบโปรแกรม YACC

โครงสร้างโปรแกรม Yacc มีรูปแบบโดยทั่วไป ดังนี้

```
Declarations
%%
Rules
%%
Support Routines
```

6.2.2 ส่วนการประกาศ (Declaration Section)

ในส่วนของการประกาศใช้เป็นที่ระบุโทเคนที่จะใช้ในไวยากรณ์ ซึ่งใน Yacc จะมีโทเคน 2 ชนิด คือ

1. โทเคนแบบอักษรตัวเดียว (single character) โทเคนชนิดนี้ไม่ต้องประกาศไว้ในส่วนของการประกาศ เมื่อจะใช้ให้ใส่อยู่ในเครื่องหมาย '...' เลย เช่น '\n' '\t'

2. โทเคนแบบอักษรหลายตัว (multicharacter token) โทเคนชนิดนี้จะถูกส่งมาจากตัววิเคราะห์คำ แต่ละตัวจะต้องถูกประกาศไว้เป็นชื่อในส่วนประกาศ รูปแบบการประกาศคือขึ้นต้นด้วย %token แล้วตามด้วยชื่อโทเคน ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%{
C declarations and programing
%}
%token  tokenname
```

ในส่วนการประกาศ ข้อความทั้งหมดที่อยู่ระหว่างเครื่องหมาย %{ และ %} จะถูกคัดลอกโดยไม่มีเงื่อนไข ส่วนโทเคนที่จะได้รับเข้ามานั้น จะต้องมีการประกาศก่อน โดยเขียนได้ดังนี้

```
%token  CELL
%token  NUMBER
%token  MINUS
```

หรือสามารถเขียนให้อยู่ในรูปแบบที่รวบรัดได้เป็น

```
%token  CELL  NUMBER  MINUS
```

ในส่วนนี้ยังสามารถรวมเอารูทีนในภาษาซี ที่จะช่วยให้การเขียนโปรแกรมทำได้ง่ายและมีประสิทธิภาพมากขึ้น เอาไว้ด้วย โดยจะต้องอยู่ในเครื่องหมาย %{ และ %} เช่น

```
%{
#include<stdio.h>
int row,col;
%}
```

6.2.3 <u>ส่วนของการสร้างกฎ</u> (Rules Section)

ในส่วนนี้จะเป็นการสร้างกฎไวยากรณ์เพื่อแปลความหมาย หรือการกระทำที่เกิดจากการแปลความหมายนั้น การสร้าง (production) ที่ใช้สร้างกฎไวยากรณ์ใน Yacc มีรูปแบบดังนี้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
non-terminal : right hand side      { actions }
```

การกระทำที่แสดงผลของการตรงกันกับกฎไวยากรณ์ที่ระบุไว้ จะเป็นรูปแบบเหมือนกันกับถ้อยแถลงในภาษาซี และยังสามารถมีรูปแบบของตัวแปร $$ และ $n เพิ่มเข้ามาได้ ซึ่ง $$ จะเป็นลักษณะเฉพาะของนอนเทอร์มินัลทางซ้ายมือ ส่วน $n คือค่าลักษณะเฉพาะของพจน์ที่ n ของการสร้างทางขวามือ ถ้าไม่มีการกระทำใดที่จะแสดงผลของกฎนั้น รูปแบบปกติที่กำหนดไว้ คือ

```
$$ = $1;
```

ถ้าหากมีการระบุนอนเทอร์มินัลหลาย ๆ อันในกฎเดียวกัน สามารถใช้เครื่องหมาย "|" แสดงความสัมพันธ์ได้ เช่น

```
arg  :  arg '+' arg      { $$ = $1 + $3; }
     |  arg '-' arg      { $$ = $1 - $3; }
     |  arg '*' arg      { $$ = $1 * $3; }
     |  arg '/' arg      { $$ = $1 / $3; }
     ;
```

นอนเทอร์มินัลใดที่จะเป็นตัวเริ่มต้นของกฎ จะต้องระบุไว้ในส่วนประกาศ โดยมีรูปแบบการประกาศตามหลัง %start เช่น

```
%start expr
```

6.2.4 ส่วนรูทีนสนับสนุน (Support Routines Section)

ฟังก์ชันของภาษาซี ที่จะช่วยในการระบุการกระทำของแต่ละกฎไวยากรณ์ จะถูกกำหนดขึ้นในส่วนนี้ เช่น การกำหนดฟังก์ชันขึ้นมาเอง เพื่อใช้ในการคำนวณค่าให้ได้ตามรูปแบบที่ต้องการ ซึ่งจะทำให้การเขียนโปรแกรมมีความยืดหยุ่นและมีความสามารถมากขึ้น

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.5 การผ่านค่าลักษณะเฉพาะจากตัววิเคราะห์คำ

เมื่อรูทีน yylex() ถูกเรียกโดยรูทีน yyparse() มันจะทำการส่งค่าโทเคนตัวต่อไปมาให้ตัวแยกคำ เช่น ค่าลักษณะเฉพาะที่สัมพันธ์กับโทเคนนั้น และกลุ่มอักษรที่ yylex() ส่งมาจะถูกเก็บไว้ในอะเรย์ yytext ซึ่งเราสามารถนำเอากลุ่มอักษรที่รับมานี้ ไปใช้ประโยชน์ในการวิเคราะห์หรือแสดงผลบางอย่างได้ตามที่ต้องการ

การคอมไพล์โปรแกรมที่สร้างด้วย Yacc มีรูปแบบเป็น

```
yacc  -d  filename.y
```

ผลลัพธ์ที่ได้จะได้โปรแกรมภาษาซี ที่มีชื่อว่า y.tab.c กับ y.tab.h ซึ่งจะถูกสร้างโดยการกำหนดตัวเลือก -d โดย y.tab.h จะถูกโปรแกรม Lex เรียกใช้เพื่อบอกถึงค่าของโทเคนที่ทั้งสองโปรแกรมใช้ร่วมกัน การคอมไพล์โปรแกรม y.tab.c ทำได้โดยการเรียกคำสั่งดังนี้

```
cc  y.tab.c  -ly
```

-ly จะเป็นตัวเลือกที่ให้สามารถดึงฟังก์ชันบางฟังก์ชันบนระบบยูนิกซ์ เข้ากับโปรแกรมนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 ลำดับการทำงาน

เป็นรายละเอียดที่แสดงถึงขั้นตอนการนำ Lex กับ Yacc มาช่วยในการเขียนโปรแกรมวิเคราะห์คำสั่ง ซึ่งจะแบ่งเป็น 3 ขั้นตอนคือ

1. ขั้นตอนการรับข้อมูลจากโปรแกรมส่วนวินโดว์
2. ขั้นตอนการแปลความหมายของข้อมูลนั้น
3. ขั้นตอนการคำนวณค่าที่ต้องการและส่งให้ส่วนแสดงผล

รายละเอียดในแต่ละขั้นตอนจะได้กล่าวต่อไป

ข้อมูลอินพุทที่จะรับไปแปลความหมาย สามารถเป็นไปได้หลายชนิดดังจะมีรายละเอียดต่าง ๆ ดังต่อไปนี้

6.3.1 ชนิดของข้อมูล

ข้อมูลที่จะบรรจุในเซลล์ของเวอร์คชีต จำแนกได้เป็น 3 ประเภท คือ ตัวเลข ตัวอักษร และสูตร

ข้อมูลชนิดตัวเลข ต้องเป็นไปตามกฎต่อไปนี้คือ

- ต้องเริ่มต้นด้วย 0-9, +, -, $
- ห้ามมีเครื่องหมายจุลภาค หรือช่องว่างระหว่างตัวเลข
- ลงท้ายด้วยเครื่องหมาย % ได้
- สามารถป้อนข้อมูลในรูปแบบสมการกำลังได้ ตัวอย่างเช่น

2.85E+04 = 28500

ข้อมูลชนิดตัวอักษร ต้องเป็นไปตามกฎต่อไปนี้คือ

- เริ่มต้นด้วยเครื่องหมาย ^ ' " หรือตัวอักษรที่ไม่ใช่ตัวเลขในการป้อนตัวเลขหรือสูตร
- สามารถเริ่มต้นด้วยอักษรแล้วตามด้วยตัวเลขได้ เช่น ab123

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่ต้องการให้เวอร์คชีตเก็บข้อมูลที่ประกอบด้วยตัวเลขในลักษณะของตัวอักษร จะต้องป้อนข้อมูลโดยเริ่มต้นด้วยเครื่องหมาย ' " ^ ซึ่งรูปแบบการแสดงผลของแต่ละเครื่องหมายคือ

' แสดงชิดขอบซ้ายของเซลล์

" แสดงชิดขอบขวาของเซลล์

^ แสดงตรงกลางเซลล์

ข้อมูลชนิดสูตร สูตรจะให้ผลลัพธ์การคำนวณที่เป็นตัวเลขหรือตัวอักษร โดยที่สูตรการคำนวณจะไม่ปรากฏในเซลล์ การป้อนข้อมูลที่เป็นสูตร จะต้องเป็นไปตามกฎต่อไปนี้ คือ

- จะต้องเริ่มต้นด้วยตัวเลข 0-9, +, -, @, # หรือ $
- ห้ามมีช่องว่าง

6.3.2 การป้อนข้อมูล

ในการป้อนข้อมูล ผู้ใช้สามารถป้อนข้อมูลเข้าไปในเซลล์ได้โดยตรงขณะที่เซลล์นั้น ๆ ถูกใช้งาน เมื่อป้อนข้อมูลเข้าไปแล้วต้องกดคีย์เอนเทอร์ เพื่อรับรองการนำข้อมูลเข้าไปเก็บจากนั้นข้อมูลจะถูกนำไปตีความเพื่อประมวลผลตามฟังก์ชันการทำงานที่ต้องการต่อไป

6.3.2.1 ขั้นตอนการรับข้อมูล

ส่วนที่ทำการรับข้อมูลนี้ จะทำการรับข้อมูลจากอะเรย์ของตัวอักษรที่ชื่อ buffer ซึ่งถูกส่งมาจากส่วนอินเตอร์เฟสของวินโดว์ จากนั้นเมื่อมีการเรียกใช้รูทีน yyparse() ตัว yyparse จะไปเรียกใช้รูทีน yylex() ซึ่งมีหน้าที่ในการอ่านอักขระทีละตัวจากอินพุทของ Lex

ในการกำหนดอินพุทให้กับ Lex นั้น จะต้องเขียนฟังก์ชันกำหนดอินพุทและเอาท์พุทให้ Lex ใหม่ เนื่องจากตามรูปแบบปกติที่กำหนดไว้ Lex จะรับข้อมูลจากอินพุทมาตรฐาน ซึ่งก็คือรับการคีย์จากจอภาพ ฟังก์ชันที่สร้างขึ้นใหม่เพื่อรับอินพุทจากบัฟเฟอร์ จะเปลี่ยนทิศทางของอินพุทและเอาท์พุทให้เป็นไปตามรูปแบบที่ส่วนอินเตอร์เฟสวินโดว์ส่งมา ซึ่งการแก้ไขนี้ เข้าไปทำในโปรแกรม yy.lex.c

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.2.2 ขั้นตอนการแปลความหมาย

ในขั้นตอนการแปลความหมาย มี 2 ขั้นตอนคือ ขั้นตอนการจัดกลุ่มค่า ซึ่งต้องอาศัยการทำงานของตัววิเคราะห์ค่า โดยมีการกำหนดรูปแบบของตัวอักษรที่จะสร้างเป็นโทเคนเป็นรูปแบบของนิพจน์ปกติ จากนั้น ในส่วนของการวิเคราะห์ค่านี้ ได้กำหนดให้มีการคำนวณค่าต่าง ๆ ที่ต้องการใช้ เช่น การคำนวณหาแถวและหลักของเซลที่เป็นโทเคน เพื่อเก็บไว้ใช้ในการคำนวณในภายหลัง

เมื่อตัววิเคราะห์ค่าสามารถแยกค่าได้เป็นโทเคนได้แล้ว จะทำการส่งโทเคนนั้นไปยังตัวแยกค่าเพื่อเปรียบเทียบว่าตรงกับไวยากรณ์ที่กำหนดไว้หรือไม่ ถ้าไม่ตรงกับไวยากรณ์ที่กำหนด ตัววิเคราะห์ไวยากรณ์นี้จะเรียกให้ตัววิเคราะห์ค่าส่งโทเคนต่อไปมาให้ เพื่อพยายามทำให้โทเคนเหล่านั้นตรงกับไวยากรณ์อันใดอันหนึ่ง ถ้าหากไม่ตรงกับไวยากรณ์ใดเลย ตัววิเคราะห์ไวยากรณ์จะบอกข้อผิดพลาดออกมา

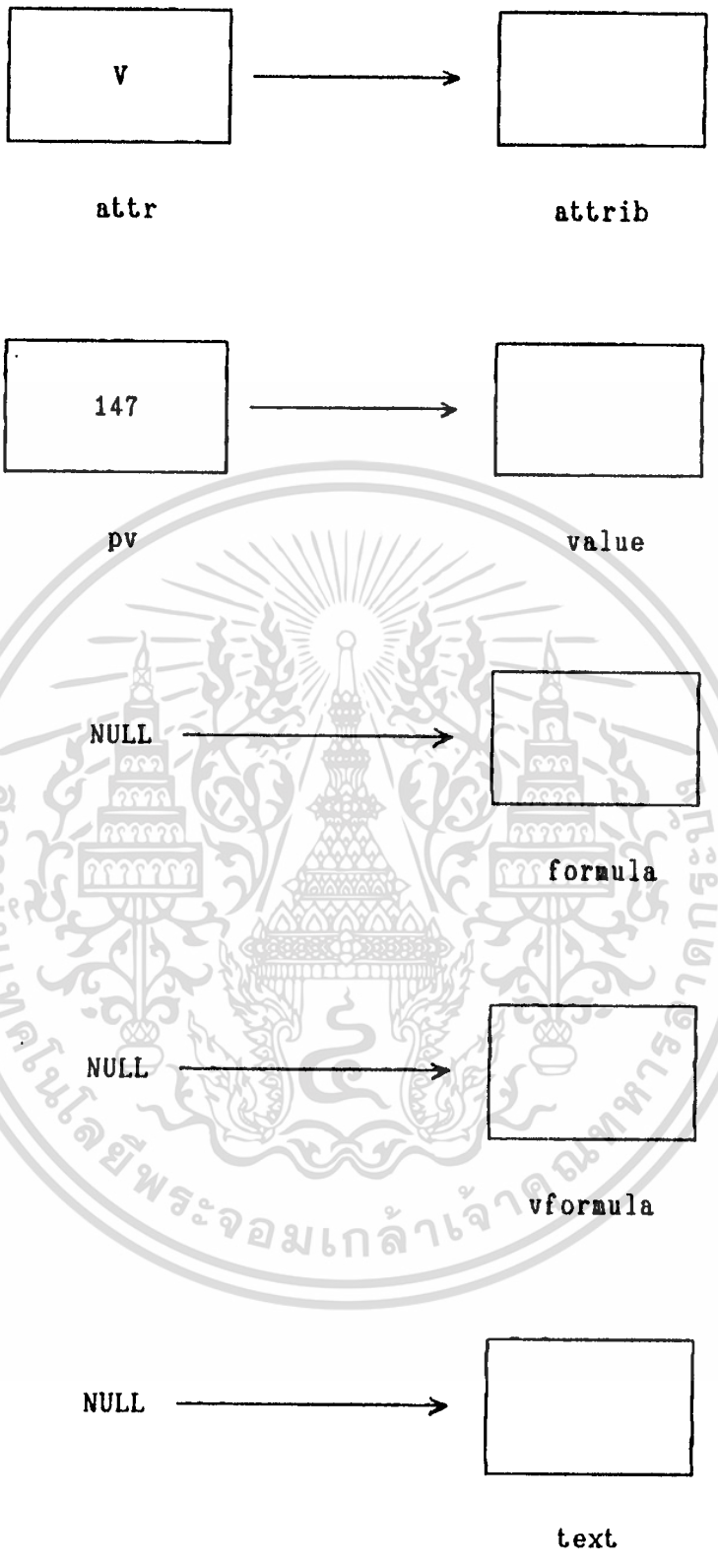
6.3.2.3 ขั้นตอนการคำนวณค่า

ในขั้นตอนการคำนวณค่ามีอยู่ทั้งในขั้นตอนการวิเคราะห์ค่าและในขั้นตอนการวิเคราะห์ไวยากรณ์ ซึ่งสามารถแยกเป็นกรณีต่าง ๆ ได้ดังนี้

3.1 กรณีอินพุทที่รับเข้ามาเป็นตัวเลข หรือการกระทำระหว่างตัวเลข ค่าลักษณะเฉพาะที่ได้จากส่วนวิเคราะห์ไวยากรณ์จะระบุว่าข้อมูลชนิดนั้น เป็นข้อมูลชนิดตัวเลข นั่นคือตัววิเคราะห์ค่าจะส่งค่า 'V' ไปให้ตัวแปร attr เมื่อมีการเรียกใช้รูทีน yyparse ตัวแปรนี้จะถูกตรวจสอบเพื่อระบุชนิดของข้อมูลที่จะนำไปเก็บในโครงสร้างข้อมูล

ผลการคำนวณที่ได้จากส่วนวิเคราะห์ไวยากรณ์จะถูกส่งให้ตัวแปร pv เมื่อมีการเรียกใช้รูทีน yyparse ค่าผลลัพธ์ที่อยู่ในตัวแปร pv นี้ ก็จะถูกส่งไปให้ตัวแปร value ซึ่งเป็นตัวแปรที่ทำหน้าที่เก็บค่าของตัวเลขในเซลล์ ค่านี้สามารถเรียกใช้เพื่อการคำนวณได้เมื่อมีการอ้างถึงค่าของเซลล์นั้นในสูตรของเซลล์อื่น ลักษณะการย้ายข้อมูลแสดงได้ดังรูปที่ 6.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.2 แสดงการย้ายข้อมูลในกรณีที่อินพุทเป็นข้อมูลชนิดตัวเลข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 กรณีอินพุทที่รับเข้ามาเป็นสูตร ค่าลักษณะเฉพาะที่ได้จากตัววิเคราะห์ไวยากรณ์ จะระบุว่าข้อมูลนั้น เป็นข้อมูลชนิดสูตร นั่นคือตัววิเคราะห์ไวยากรณ์จะส่งค่า 'F' ไปให้ตัวแปร attr และจะถูกส่งไปให้กับตัวแปร attrib ของเซลล์นั้นด้วย

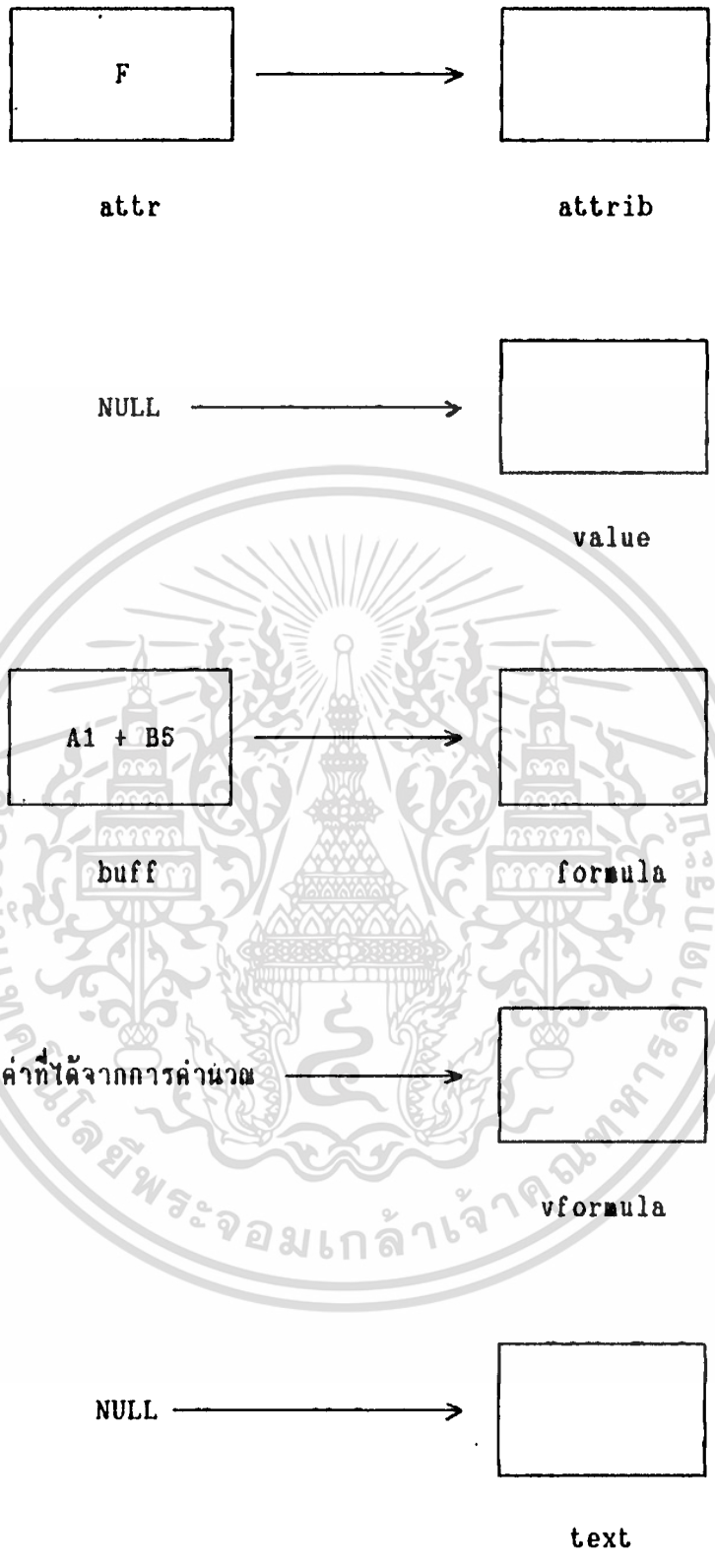
ข้อความที่เป็นตัวอักษรของสูตรเหล่านั้น ซึ่งถูกเก็บไว้ในตัวแปร buff ของตัววิเคราะห์ไวยากรณ์ จะถูกส่งไปให้ตัวแปร formula ซึ่งเป็นบัฟเฟอร์ที่เป็นที่เก็บสูตรของเซลล์นั้นด้วย ข้อความที่เก็บไว้ในบัฟเฟอร์เก็บสูตรนี้ จะถูกนำมาใช้ในการวิเคราะห์คำสั่งอีกในกรณีที่มีการคำนวณค่าของเซลนั้นใหม่ ซึ่งจะเกิดขึ้นเมื่อเซลล์ที่เกี่ยวข้องกับสูตรนั้นมีค่าเปลี่ยนไป

ในกรณีนี้จะต้องมีการส่งค่า NULL ไปให้บัฟเฟอร์ text ของเซลล์นั้นด้วย เพื่อเป็นการระบุถึงการมีเทกซ์เพียงอย่างเดียว ลักษณะการย้ายตัวแปรและข้อมูลต่าง ๆ เป็นดังรูปที่ 6.3

3.3 กรณีอินพุทที่รับเข้ามาเป็นกลุ่มตัวอักษรที่ไม่ใช่สูตร หรืออักขระใด ๆ ที่ตามหลังเครื่องหมาย ' " หรือ ^ ตัววิเคราะห์ไวยากรณ์จะตรวจสอบได้ว่าข้อมูลชนิดนั้นเป็นเทกซ์ นั่นคือค่าลักษณะเฉพาะที่ตัววิเคราะห์ส่งไปให้ตัวแปร attr จะมีค่าเป็น 'T' และจะถูกส่งต่อไปยังตัวแปร attrib ของเซลล์นั้นด้วย ข้อความที่เป็นตัวอักษรเหล่านั้นจะถูกเก็บไว้ในตัวแปร buff และจะถูกส่งไปให้บัฟเฟอร์ text ของเซลล์นั้นด้วย

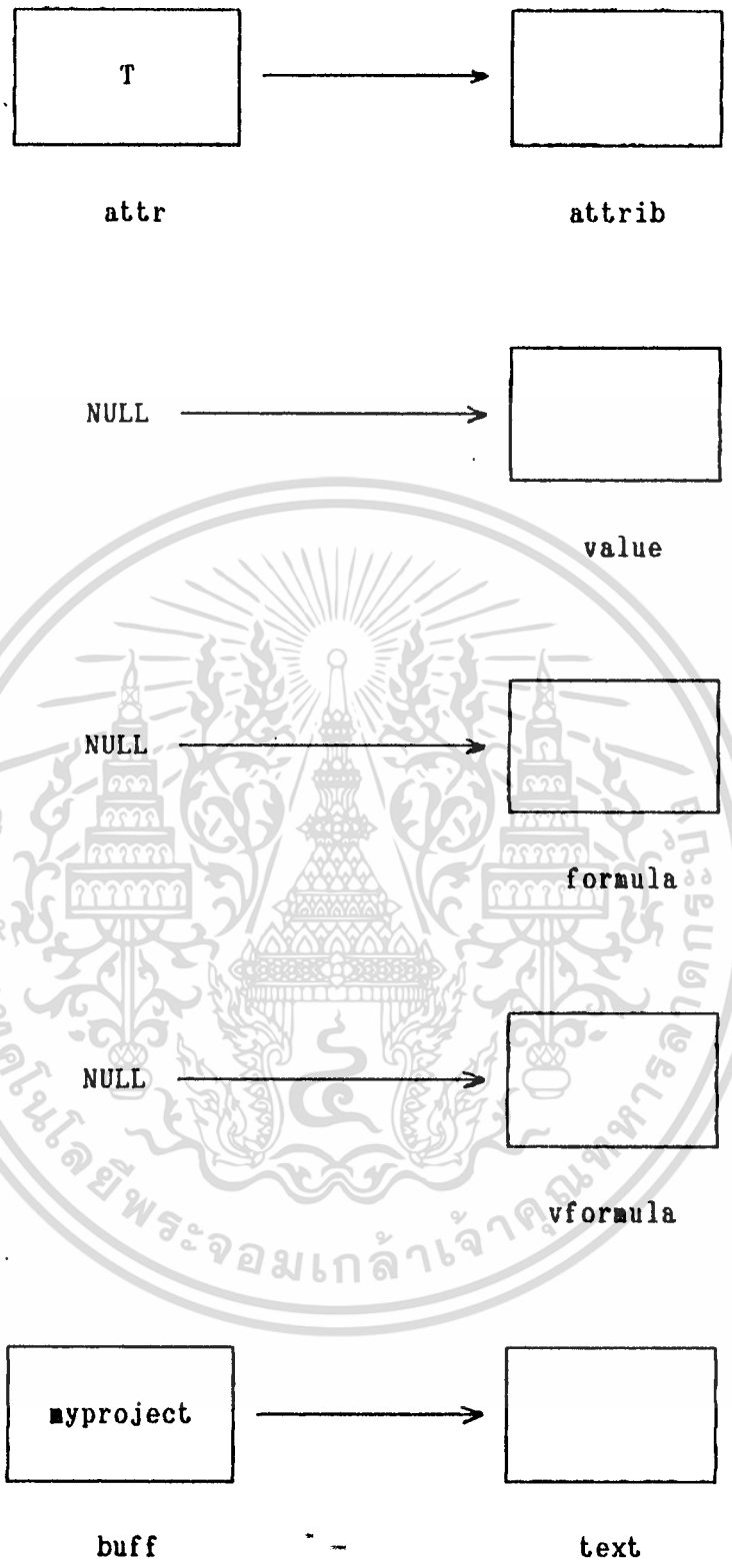
ในกรณีนี้จะต้องมีการส่งค่า NULL ไปให้บัฟเฟอร์ formula ของเซลด้วยเพื่อลบสูตรเดิมของเซลล์นั้น ให้เหลือเพียงข้อมูลชนิดเทกซ์อย่างเดียว ลักษณะการย้ายตัวแปรและข้อมูลต่าง ๆ เป็นดังรูปที่ 6.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3 แสดงการย้ายข้อมูลในกรณีที่อินพุทเป็นข้อมูลชนิดสูตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

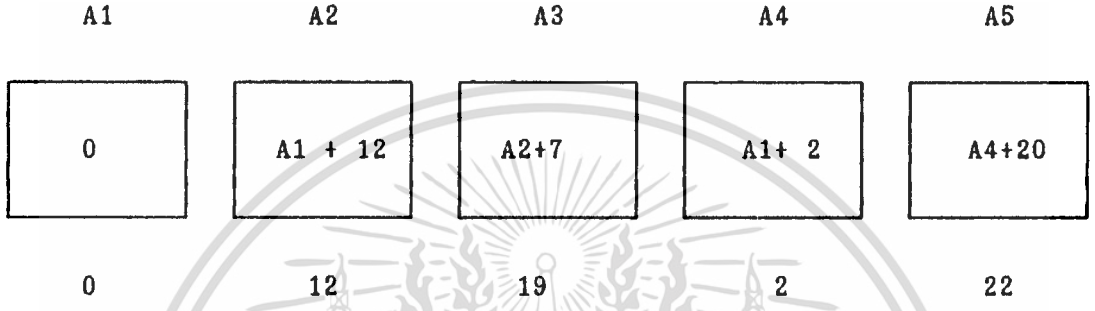


รูปที่ 6.4 แสดงการย้ายข้อมูลในกรณีที่อินพุทเป็นข้อมูลชนิดเทกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

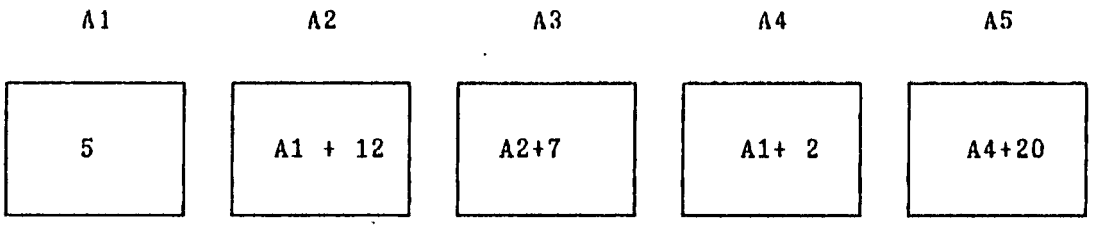
6.4 ขั้นตอนการค้นหาโหนดเพื่อทำการคำนวณใหม่

ในกรณีที่มีการเปลี่ยนค่าของเซลใด ๆ การคำนวณอัตโนมัติทำได้โดยการใช้โครงสร้างของเซลล์ไม่อิสระ ซึ่งจะทำการเก็บค่าของแถวและหลักของเซลล์ที่เกี่ยวข้องกันในสูตรของเซลล์นั้นไว้ การคำนวณจะทำการเข้าถึงเซลที่กำหนดโดยแถวและหลักของเซลล์ทีละเซลล์แล้วนำค่าในเซลล์นั้นมาทำการคำนวณ ดังตัวอย่าง



สมมุติว่ามีการในขั้นแรก A1 มีค่าเป็น 0 ค่าของแต่ละเซลล์ที่อ้างอิง A1 จะมีค่าดังตัวเลขข้างล่างของเซลล์นั้น

ถ้าหากมีการเปลี่ยนค่าของเซลล์ A1 เป็น 5 กระบวนการคำนวณอัตโนมัติจะทำการเข้าถึงโครงสร้างเซลล์ไม่อิสระของเซลล์ A1 ซึ่งจะเก็บค่าแถวและหลักของเซลล์ที่ใช้ค่าของเซลล์ A1 ในสูตรคำนวณ ซึ่งในที่นี้ก็คือ ค่าแถวและหลักของเซลล์ A2 กับ A4 จากนั้นจะทำการเข้าถึงเซลล์โดยใช้ค่าแถวและหลักในโครงสร้างนั้น และจะนำค่าของเซลล์ A1 ที่เปลี่ยนใหม่ไปทำการคำนวณในเซลล์ A2 และ A4 ใหม่ ซึ่งจะได้ค่าใหม่ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่า ในรอบแรกเซลล์ A3 และ A5 ไม่มีการเปลี่ยนแปลง แต่ขณะที่เซลล์ A2 เปลี่ยนไปนั้น จะเก็บสถานะการเปลี่ยนแปลงค่าของตัวเองไว้ เช่นเดียวกันกับ A4 เมื่อ A1 ได้ทำการคำนวณในเซลล์ที่มีสูตรอ้างถึง A1 จนหมดแล้ว เซลล์ A2 ที่มีการเปลี่ยนค่าไปเนื่องจากการคำนวณใหม่ของ A1 ก็จะรับกระบวนการคำนวณใหม่มากระทำ โดยใช้โครงสร้างข้อมูลของเซลล์ไม่อิสระของตัวเองในการอ้างถึงเซลล์ที่ใช้ค่าของเซลล์ A2 ในสูตร แล้วทำการเข้าถึงและคำนวณใหม่ ซึ่งก็คือเข้าไปคำนวณสูตรใน A3 ใหม่ ได้ค่าในขั้นตอนที่ 2 เป็น

A1	A2	A3	A4	A5
5	A1 + 12	A2+7	A1+ 2	A4+20
5	17	24	7	22

ในขั้นตอนสุดท้ายหลังจากที่เซลล์ A2 ได้ทำการคำนวณเสร็จสิ้นแล้ว เซลล์ A4 จะทำการคำนวณค่าในเซลล์ที่อ้างถึงเซลล์ A4 ใหม่ โดยใช้หลักการเดียวกัน ซึ่งจะทำให้ได้ค่าในขั้นตอนสุดท้ายเป็น

A1	A2	A3	A4	A5
5	A1 + 12	A2+7	A1+ 2	A4+20
5	17	24	7	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณใหม่โดยการเปลี่ยนค่าในเซลล์ใด ๆ นี้ จะกระทำเหมือนกับเป็นการใส่ค่าใหม่ให้กับเซลล์ ๆ ซึ่งโปรแกรมวิเคราะห์คำสั่งจะทำการเข้าไปแปลความหมายของสูตรในเซลล์นั้น โดยการอ่านสูตรมาจากบัฟเฟอร์ที่เก็บสูตรของแต่ละเซลล์นั้นไว้ และกระทำเช่นเดียวกันในการเข้าถึงเซลล์ใด ๆ ที่มีการอ้างถึงเพื่อการคำนวณ ลักษณะของการคำนวณในเซลล์อื่นจะอยู่ในรูปแบบเดียวกันนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

สรุปผลโครงงาน

การพัฒนาซอฟต์แวร์สเปรดชีตบนระบบวินโดว์ชุดนี้เป็นเพียงการเริ่มต้นเท่านั้น เวลาส่วนใหญ่จะเสียไปกับการออกแบบส่วนต่าง ๆ โดยเฉพาะในส่วนของโครงสร้างข้อมูลซึ่งเป็นส่วนสำคัญของการทำซอฟต์แวร์ โครงสร้างข้อมูลต้องถูกออกแบบให้การเข้าถึงข้อมูลเป็นไปโดยรวดเร็วและถูกต้อง ได้นำโครงสร้างแบบลิงค์ลิสมาประยุกต์ใช้โดยออกแบบการจัดการใหม่ให้เป็นแบบลิงค์ลิสของอะเรย์ขนาดสิบคูณสิบ จะช่วยลดเวลาในการเข้าถึงข้อมูลให้น้อยลงและประหยัดหน่วยความจำ ขณะนี้ทำงานได้เพียงฟังก์ชั่นหลัก ๆ สำหรับสเปรดชีตเท่านั้นเช่น การจัดเก็บข้อมูล การดึงข้อมูลมาแก้ไข การจัดการทางด้านแถว การจัดการทางด้านหลัก การขยายความกว้างของคอลัมน์ และการคำนวณพื้นฐาน

ลักษณะการทำงานแบบวินโดว์ก็อำนวยความสะดวกในการทำงานได้ดีมาก อย่างน้อยก็ใช้เมาส์ร่วมกับคีย์บอร์ดได้ จึงสนุกและทำงานได้อย่างรวดเร็วกว่าใช้คีย์บอร์ดอย่างเดียว อีกอย่างสีสันก็สวยงามกว่า ช่วยเสริมบรรยากาศการใช้งาน ระบบการอ้างอิงเซลล์ถูกออกแบบเป็นระบบแถวและหลักโดยจุดตัดของแถวและหลักจะเป็นตำแหน่งของเซลล์ ประกอบด้วยห้าร้อยหลักและสองพันแถว ดังนั้นจึงอ้างอิงได้ทั้งหมดหนึ่งล้านเซลล์ ผู้ใช้สามารถขยายความกว้างของคอลัมน์ได้ตามต้องการ โอเปอร์เรชั่นต่าง ๆ มีให้ใช้ได้ในเมนูและในส่วนของผู้พัฒนาสามารถเพิ่มโอเปอร์เรชั่นได้อีกตามต้องการ การใช้เมนูแบบ Pullright menu เป็นไปโดยง่าย ขณะที่ไม่เรียกใช้ เมนูก็จะไม่ปรากฏให้ผู้ใช้เห็นจึงช่วยให้เวอร์คชีตสะอาด ในส่วนของการคำนวณก็ทำได้เพียงพื้นฐาน เช่น บวก ลบ คูณ และหาร

อย่างไรก็ตามในการพัฒนาก็ยังมีปัญหาอยู่อีกหลายประการที่อยากจะแจ้งให้ทราบ พร้อมกับแนวทางในการแก้ปัญหาเพื่อการปรับปรุงให้ใช้งานได้ดีขึ้นในอนาคต พอจะสรุปได้ดังต่อไปนี้

1. ปัญหาการเลื่อนสกอลล์บาร์ยังมีความละเอียดไม่เพียงพอเนื่องมาจากฟังก์ชั่นในการคำนวณ ค่า m และ n ที่เป็นตัวนาบาร์เคลื่อนที่ไปตามที่กำหนด แต่อย่างไรก็ตามผู้ใช้ก็ไม่รู้สึกว่ามันคลาดเคลื่อน แต่ในส่วนของผู้พัฒนาจะทราบดี ถ้าหากต้องการค่าที่ละเอียดมาก ๆ ต้องออกแบบสมการการคำนวณใหม่โดยยึดค่าความกว้างและความสูงของวินโดว์เป็นหลักและต้องหาโครงสร้างข้อมูลมาเก็บตัวแปรชุดที่ควบคุมการทำงานของสกอลล์บาร์

2. ถ้าหากต้องการให้การทำงานเป็นไปด้วยความรวดเร็วควรจะมีการทำงานแบบบล็อกได้ ซึ่งก็ได้เริ่มวางแนวทางในการพัฒนาเอาไว้แล้วในภาคผนวก บังเอิญว่าช่วงหลังมีปัญหาในการใช้เครื่องจึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่มีเวลาพอที่จะทำให้สมบูรณ์ การทำงานแบบบล็อกต้องมีตัวแปรในโครงสร้างข้อมูลมาเก็บต่างหากเพื่อแยกว่าขณะนั้นการทำงานเป็นไปตามข้อกำหนดของเมนูหรือการทำงานของบล็อก รายละเอียดกรุณาอ่านได้ที่ภาคผนวก ก

3. ในส่วนของโครงสร้างข้อมูลได้ออกแบบมาดีแล้ว แต่ก็อยากจะเสนอแนวทางในการพัฒนาต่อนส่วนของเซลล์ไม่อิสระ(Dependence cell)เอาไว้เพื่อช่วยในส่วนการคำนวณ ต้องมีตัวแปรในโครงสร้างข้อมูลมาเก็บค่าตัวแปรเซลล์ไม่อิสระนี้ โดยต้องประกาศเป็นแบบให้ใช้ได้ทั่วทั้งโปรแกรม ตัวแปรนี้จะเก็บค่าแถวและหลักของเซลล์ที่อ้างตัวมันอยู่ ตัวอย่างเช่นในสูตรการคำนวณ C1 เท่ากับ A1+A2 ในตัวแปร C1 จะเก็บค่าของตัวแปรเซลล์ A1 และ A2 พร้อมกับค่าหลักและแถวด้วย เพราะฉะนั้นที่เซลล์ A1 ตรงตัวแปรเซลล์ไม่อิสระต้องเก็บค่าหลักและแถวของเซลล์ C1 เซลล์ A2 ก็เก็บในทำนองเดียวกัน การที่เราเก็บแบบนี้ก็จะทราบได้ทันทีว่าถ้าหากค่าในเซลล์ A1 หรือเซลล์ที่เกี่ยวข้องกันเปลี่ยนไปค่าในเซลล์ไหนจะเปลี่ยนไปบ้าง

4. ปัญหาการคำนวณซึ่งตอนนี้ยังทำงานโดยระบบสูตรคำนวณได้ไม่สมบูรณ์ แต่ได้วางโครงสร้างเอาไว้ให้แล้วเป็นส่วนใหญ่

------***------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

[1] Adrian Nye, The Definition Guides to the X Window System Volumn Two Xlib Reference Manual for Version 11 ,O'Reilly & Associates,Inc,1988.

[2] Adrian Nye, Volumn One Xlib Programming Manual for version 11 of the X Window System ,O'Reilly & Associates,Inc,1988.

[3] Alfraed V. Aho,Ravi Sethi,Jeffrey D. Ullman, Compiles Principles, Techniques,and Tools,Addison-Wesley publishing company.

[4] Dan Heller, Volumn Seven XView Programming Manual for version 11 of the X Window System,O'Reilly & Associates,Inc,1989.

[5] J.P. Bennett,Introduction to compiling Techniques: A First course using ANSI C,LEX and YACC,Mcgraw-hill book company,1990.

[6] Robert W. Scheifler edited and with an introduction by Nye,Volumn Zero X Protocol Reference Manual for version 11 of the X Window System,O'Reilly & Associates,Inc,1988.

------***------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การทำปริญญานิพนธ์ฉบับนี้สำเร็จได้ด้วยดีนั้น ได้รับความช่วยเหลือจากผู้มีอุปการะคุณหลายท่าน ขอกราบขอบพระคุณ ดร. บุญธีร์ เครือตราชู อาจารย์ที่ปรึกษาผู้ซึ่งให้ความอนุเคราะห์ทั้งด้านความรู้ แนวทาง แนวความคิด อุปกรณ์ในการทำงาน เสียสละทั้งแรงกายแรงใจ และเป็นผู้ให้กำลังใจตลอดเวลา ขอกราบขอบพระคุณ รองศาสตราจารย์ ดร. บุญวัฒน์ อัตชู ที่ให้ความอนุเคราะห์เครื่องคอมพิวเตอร์ ห้องทำงาน และอุปกรณ์อื่น ๆ ในการพัฒนาโปรแกรมและทำปริญญานิพนธ์ และขอกราบขอบพระคุณคณาจารย์ควบคุมการสอบปริญญานิพนธ์ทุกท่านที่เสียสละเวลาอันมีค่าของท่านมาควบคุมการสอบ พร้อมกันนี้ต้องขอขอบคุณ พี่ปริญญาโทที่ให้คำแนะนำ เพื่อน ๆโดยเฉพาะ วรชัย ที่ช่วยสอนการใช้พอยเตอร์ และน้อง ๆ ทุกคนที่ให้ความช่วยเหลือมาโดยตลอด

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

-----***-----

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

(ก)

โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* ******spx.c Listing Program of spreadsheet ***** */
#include "heading.h"
Xv_xrectlist  *xrects;
Xv_Font   font;
Canvas   canvas;
char buffer[MAXINPUT+1];
char dex;
int scol,srow;
int curcol,wd;
int x,y;
char c;
/*  ******** function_quit is provide condition to quit ******** */
Notify_value
destroy_func(client,status)
Notify_client  client;
Destroy_status  status;
{
 if (status == DESTROY_CHECKING)
 {
  int answer = notice_prompt(client,NULL,
       NOTICE_MESSAGE_STRINGS, "**Not save sheet**/Really Quit?",NULL,
       NOTICE_BUTTON_YES, "No",
       NOTICE_BUTTON_NO,  "Yes",
       NULL);
       if (answer == NOTICE_YES)
          notify_veto_destroy(client);
 }
 else  if (status == DESTROY_CLEANUP)
   {
      puts ("Cleaning up ");
      /* allow frame to be destroy */
      return notify_next_destroy_func(client,status);
   }
   else if (status == DESTROY_SAVE_YOURSELF)
      puts("Save yourself?");
   else
      puts("Process death");
   return NOTIFY_DONE;
}
/* **************main program **************** */
main(argc,argv)
int  argc;
char *argv[];
{
XGCValues  gc_val;
unsigned char red[MAXCOLORS],green[MAXCOLORS],blue[MAXCOLORS];
static char stipple_bits[] = {0xAA,0xAA,0x55,0x55};
Server_image  icon_image;
void  canvas_repaint();
void  event_proc();
Xv_cmsdata  cms_data;
int   use_dynamic = FALSE;
headptr_width = NULL;

/* **Openwindow and initialize configuration  */
xv_init(XV_INIT_ARGC_PTR_ARGV, &argc,argv,NULL);

/* ************* create base frame ********** */

if ( *++argv && !strcmp(*argv,"-dynamics"))
        use_dynamic = TRUE;
/* ************** create frame ******* */
frame = xv_create(XV_NULL,FRAME,
        XV_WIDTH,    850,
        XV_HEIGHT,   630,
        FRAME_LABEL, "Worksheet",
        FRAME_SHOW_FOOTER, TRUE,
        NULL);
notify_interpose_destroy_func(frame,destroy_func);
/* allocate RGB color to use in program */
 red[WHITE] = 255; green[WHITE] = 255; blue[WHITE] = 255;
 red[RED] = 255;   green[RED] = 0;     blue[RED] = 0;
 red[GREEN] = 0;   green[GREEN] = 255; blue[GREEN] = 0;
 red[BLUE] = 0;    green[BLUE] = 0;    blue[BLUE] = 255;
 red[ORANGE] = 204;green[ORANGE] = 50;blue[ORANGE] =50;
 red[AQUA] = 112; green[AQUA] = 219;  blue[AQUA] = 147;
 red[PINK] = 188;  green[PINK] = 143;   blue[PINK] = 143;
 red[BLACK] = 0;   green[BLACK] = 0;   blue[BLACK] =0;
 red[GRAY] = 192;  green[GRAY] = 192;  blue[GRAY] = 192;
 red[BROWN] = 165; green[BROWN] = 42;   blue[BROWN] = 42;
 red[GOLD] = 204;  green[GOLD] = 127;  blue[GOLD] = 50;
 red[YELLOW] = 255;green[YELLOW] = 255; blue[YELLOW] = 0;
 red[MAROON] = 142;green[MAROON] = 35;blue[MAROON] =107;
 red[VIOLET] = 79;   green[VIOLET] = 47;  blue[VIOLET] = 79;
 red[ORCHID] = 219;  green[ORCHID] = 112;   blue[ORCHID] = 219;
 cms_data.type = use_dynamic? XV_DYNAMIC_CMS : XV_STATIC_CMS;

 cms_data.size = MAXCOLORS;
 cms_data.rgb_count = MAXCOLORS;
 cms_data.index = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

```
    cms_data.red = red;
    cms_data.green = green;
    cms_data.blue = blue;
    initvars();
    initptr();
  /* create a command base frame -- not display until XV_SHOW is set */
    frame_save = (Frame)xv_create(frame,FRAME_CMD,
        FRAME_LABEL,    "SHEET: Save",
        XV_WIDTH,    400,
        XV_HEIGHT,   100,
        NULL);
  /* command base frame have panels already created by default - - get it */
    panel_save = (Panel)xv_get(frame_save,FRAME_CMD_PANEL,
            PANEL_LAYOUT,    PANEL_VERTICAL,
            NULL);
     create_text_save = (Panel_item)xv_create(panel_save,PANEL_TEXT,
            PANEL_LABEL_STRING, "Enter file name :",
            PANEL_NOTIFY_PROC,   notify_save,
            PANEL_LAYOUT,   PANEL_HORIZONTAL,
            PANEL_VALUE_DISPLAY_LENGTH,   20,
            NULL);
    (void)xv_create(panel_save,PANEL_BUTTON,
            PANEL_LABEL_STRING, "SAVE",
            PANEL_NOTIFY_PROC,   notify_save,
            NULL);
  /* ************************end of save*****************    */
    frame_load = (Frame)xv_create(frame,FRAME_CMD,
        FRAME_LABEL,    "SHEET: load",
        XV_WIDTH,    400,
        XV_HEIGHT,   100,
        NULL);
    panel_load = (Panel)xv_get(frame_load,FRAME_CMD_PANEL,
            PANEL_LAYOUT,    PANEL_VERTICAL,
            NULL);
     create_text_load = (Panel_item)xv_create(panel_load,PANEL_TEXT,
            PANEL_LABEL_STRING, "Enter file name :",
            PANEL_NOTIFY_PROC,   notify_load,
            PANEL_LAYOUT,   PANEL_HORIZONTAL,
            PANEL_VALUE_DISPLAY_LENGTH,   20,
            NULL);
    (void)xv_create(panel_load,PANEL_BUTTON,
            PANEL_LABEL_STRING, "LOAD",
            PANEL_NOTIFY_PROC,   notify_load,
            NULL);
  /* ************************end of load*****************    */
    frame_print = (Frame)xv_create(frame,FRAME_CMD,
        FRAME_LABEL,    "SHEET: print",
        XV_WIDTH,    400,
        XV_HEIGHT,   100,
        NULL);
    panel_print = (Panel)xv_get(frame_print,FRAME_CMD_PANEL,
            PANEL_LAYOUT,    PANEL_VERTICAL,
            NULL);
     create_text_print = (Panel_item)xv_create(panel_print,PANEL_TEXT,
            PANEL_LABEL_STRING, "Enter file name :",
            PANEL_NOTIFY_PROC,   notify_print,
            PANEL_LAYOUT,   PANEL_HORIZONTAL,
            PANEL_VALUE_DISPLAY_LENGTH,   20,
            NULL);
    (void)xv_create(panel_print,PANEL_BUTTON,
            PANEL_LABEL_STRING, "PRINT",
            PANEL_NOTIFY_PROC,   notify_print,
            NULL);
  /* ********************end of print*****************    */
    frame_format = (Frame)xv_create(frame,FRAME_CMD,
        FRAME_LABEL,    "SHEET: format",
        XV_WIDTH,    400,
        XV_HEIGHT,   100,
        NULL);
    panel_format = (Panel)xv_get(frame_format,FRAME_CMD_PANEL,
            PANEL_LAYOUT,    PANEL_VERTICAL,
            NULL);
     create_text_format = (Panel_item)xv_create(panel_format,PANEL_TEXT,
            PANEL_LABEL_STRING, "Enter the cell to format:",
            PANEL_NOTIFY_PROC,   notify_format,
            PANEL_LAYOUT,   PANEL_HORIZONTAL,
            PANEL_VALUE_DISPLAY_LENGTH,   20,
            NULL);
    (void)xv_create(panel_format,PANEL_BUTTON,
            PANEL_LABEL_STRING, "FORMAT",
            PANEL_NOTIFY_PROC,   notify_format,
            NULL);
  /* ********************end of format***************** */
    frame_goto = (Frame)xv_create(frame,FRAME_CMD,
        FRAME_LABEL,    "SHEET: goto",
        XV_WIDTH,    400,
        XV_HEIGHT,   100,
        NULL);
    panel_goto = (Panel)xv_get(frame_goto,FRAME_CMD_PANEL,
            PANEL_LAYOUT,    PANEL_VERTICAL,
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            NULL);
    create_text_goto = (Panel_item)xv_create(panel_goto,PANEL_TEXT,
            PANEL_LABEL_STRING, "Enter the cell to goto:",
            PANEL_NOTIFY_PROC,  notify_goto,
            PANEL_LAYOUT,  PANEL_HORIZONTAL,
            PANEL_VALUE_DISPLAY_LENGTH,  20,
            NULL); /* end of goto */
    (void)xv_create(panel_goto,PANEL_BUTTON,
            PANEL_LABEL_STRING, "GOTO",
            PANEL_NOTIFY_PROC,  notify_goto,
            NULL);
  /*  ******************* end of goto ********** */
    frame_width = (Frame)xv_create(frame,FRAME_CMD,
        FRAME_LABEL,    "SHEET: set width",
        XV_WIDTH,    400,
        XV_HEIGHT,   100,
        NULL);
    panel_width = (Panel)xv_get(frame_width,FRAME_CMD_PANEL,
            PANEL_LAYOUT,   PANEL_VERTICAL,
            NULL);
    create_text_width1 = (Panel_item)xv_create(panel_width,PANEL_TEXT,
            PANEL_LABEL_STRING, "Enter column to set :",
            PANEL_NOTIFY_PROC,  notify_width1,
            PANEL_LAYOUT,  PANEL_HORIZONTAL,
            PANEL_VALUE_DISPLAY_LENGTH,  10,
            NULL);
    create_text_width2 = (Panel_item)xv_create(panel_width,PANEL_TEXT,
            PANEL_LABEL_STRING, "Enter width of column :",
            PANEL_NOTIFY_PROC,  notify_width2,
            PANEL_LAYOUT,  PANEL_HORIZONTAL,
            PANEL_VALUE_DISPLAY_LENGTH,  10,
            NULL);
    (void)xv_create(panel_width,PANEL_BUTTON,
            PANEL_LABEL_STRING, "SELECT",
            PANEL_NOTIFY_PROC,  notify_width2,
            NULL);
  /*  ******************* end of set width ********** */
    frame_insert_col = (Frame)xv_create(frame,FRAME_CMD,
        FRAME_LABEL,    "SHEET: Insert column",
        XV_WIDTH,    400,
        XV_HEIGHT,   100,
        NULL);
    panel_insert_col = (Panel)xv_get(frame_insert_col,FRAME_CMD_PANEL,
            PANEL_LAYOUT,   PANEL_VERTICAL,
            NULL);
    create_text_insertcol = (Panel_item)xv_create(panel_insert_col,PANEL_TEXT,
            PANEL_LABEL_STRING, "Enter column to inserted(A-IV) :",
            PANEL_NOTIFY_PROC,  notify_insert_col,
            PANEL_LAYOUT,  PANEL_HORIZONTAL,
            PANEL_VALUE_DISPLAY_LENGTH,  10,
            NULL);
    (void)xv_create(panel_insert_col,PANEL_BUTTON,
            PANEL_LABEL_STRING, "Insert column",
            PANEL_NOTIFY_PROC,  notify_insert_col,
            NULL);
/* ************************end of insert column*****************    */
    frame_insert_row  = (Frame)xv_create(frame,FRAME_CMD,
        FRAME_LABEL,    "SHEET: Insert row",
        XV_WIDTH,    400,
        XV_HEIGHT,   100,
        NULL);
    panel_insert_row = (Panel)xv_get(frame_insert_row,FRAME_CMD_PANEL,
            PANEL_LAYOUT,   PANEL_VERTICAL,
            NULL);
    create_text_insertrow = (Panel_item)xv_create(panel_insert_row,PANEL_TEXT,
            PANEL_LABEL_STRING, "Enter row to be inserted:",
            PANEL_NOTIFY_PROC,  notify_insert_row,
            PANEL_LAYOUT,  PANEL_HORIZONTAL,
            PANEL_VALUE_DISPLAY_LENGTH,  10,
            NULL);
    (void)xv_create(panel_insert_row,PANEL_BUTTON,
            PANEL_LABEL_STRING, "Insert row",
            PANEL_NOTIFY_PROC,  notify_insert_row,
            NULL);
/* ************************end of inset row*****************    */
    frame_delete_col = (Frame)xv_create(frame,FRAME_CMD,
        FRAME_LABEL,    "SHEET: Delete column",
        XV_WIDTH,    400,
        XV_HEIGHT,   100,
        NULL);
    panel_delete_col = (Panel)xv_get(frame_delete_col,FRAME_CMD_PANEL,
            PANEL_LAYOUT,   PANEL_VERTICAL,
            NULL);
    create_text_deletecol = (Panel_item)xv_create(panel_delete_col,PANEL_TEXT,
            PANEL_LABEL_STRING, "Enter column to delete(A-IV) :",
            PANEL_NOTIFY_PROC,  notify_delete_col,
            PANEL_LAYOUT,  PANEL_HORIZONTAL,
            PANEL_VALUE_DISPLAY_LENGTH,  10,
            NULL);
    (void)xv_create(panel_delete_col,PANEL_BUTTON,
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น พร้อมทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            PANEL_LABEL_STRING, "Delete column",
            PANEL_NOTIFY_PROC,  notify_delete_col,
            NULL);
/* ********************end of delete column****************** */
  frame_delete_row = (Frame)xv_create(frame,FRAME_CMD,
      FRAME_LABEL,   "SHEET: Delete row",
      XV_WIDTH,    400,
      XV_HEIGHT,   100,
      NULL);
  panel_delete_row = (Panel)xv_get(frame_delete_row,FRAME_CMD_PANEL,
          PANEL_LAYOUT,   PANEL_VERTICAL,
          NULL);
   create_text_deleterow = (Panel_item)xv_create(panel_delete_row,PANEL_TEXT,
          PANEL_LABEL_STRING, "Enter row to be delete:",
          PANEL_NOTIFY_PROC,  notify_delete_row,
          PANEL_LAYOUT,  PANEL_HORIZONTAL,
          PANEL_VALUE_DISPLAY_LENGTH,  20,
          NULL);
  (void)xv_create(panel_delete_row,PANEL_BUTTON,
          PANEL_LABEL_STRING, "Delete row",
          PANEL_NOTIFY_PROC,  notify_delete_row,
          NULL);
/* ********************end of delete row****************** */
  frame_fill_data = (Frame)xv_create(frame,FRAME_CMD,
      FRAME_LABEL,   "SHEET: Fill data",
      XV_WIDTH,    400,
      XV_HEIGHT,   150,
      NULL);
  panel_fill_data = (Panel)xv_get(frame_fill_data,FRAME_CMD_PANEL,
          PANEL_LAYOUT,   PANEL_VERTICAL,
          NULL);
   create_text1 = (Panel_item)xv_create(panel_fill_data,PANEL_TEXT,
          PANEL_LABEL_STRING, "Row & Column to start:",
          PANEL_NOTIFY_PROC,  notify_fill_data1,
          PANEL_LAYOUT,  PANEL_HORIZONTAL,
          PANEL_VALUE_DISPLAY_LENGTH,  8,
          NULL);
   create_text2 = (Panel_item)xv_create(panel_fill_data,PANEL_TEXT,
          PANEL_LABEL_STRING, "End of  Row & Column :",
          PANEL_NOTIFY_PROC,  notify_fill_data2,
          PANEL_LAYOUT,  PANEL_HORIZONTAL,
          PANEL_VALUE_DISPLAY_LENGTH,  8,
          NULL);
   create_text3 = (Panel_item)xv_create(panel_fill_data,PANEL_TEXT,
          PANEL_LABEL_STRING, "First value to fill:",
          PANEL_NOTIFY_PROC,  notify_fill_data3,
          PANEL_LAYOUT,  PANEL_HORIZONTAL,
          PANEL_VALUE_DISPLAY_LENGTH,  10,
          NULL);
   create_text4 = (Panel_item)xv_create(panel_fill_data,PANEL_TEXT,
          PANEL_LABEL_STRING, "End value to fill:",
          PANEL_NOTIFY_PROC,  notify_fill_data4,
          PANEL_LAYOUT,  PANEL_HORIZONTAL,
          PANEL_VALUE_DISPLAY_LENGTH,  12,
          NULL);
   create_text5 = (Panel_item)xv_create(panel_fill_data,PANEL_TEXT,
          PANEL_LABEL_STRING, "                Step:",
          PANEL_NOTIFY_PROC,  notify_fill_data5,
          PANEL_LAYOUT,  PANEL_HORIZONTAL,
          PANEL_VALUE_DISPLAY_LENGTH,  9,
          NULL);
  (void)xv_create(panel_fill_data,PANEL_BUTTON,
          PANEL_LABEL_STRING, "Fill data",
          PANEL_NOTIFY_PROC,  notify_fill_data5,
          NULL);
/* ****************** End of fill data***************** */
  frame_copy = (Frame)xv_create(frame,FRAME_CMD,
      FRAME_LABEL,   "SHEET: Copy data",
      XV_WIDTH,    400,
      XV_HEIGHT,   150,
      NULL);
  panel_copy = (Panel)xv_get(frame_copy,FRAME_CMD_PANEL,
          PANEL_LAYOUT,   PANEL_VERTICAL,
          NULL);
   create_text_copy1=(Panel_item)xv_create(panel_copy,PANEL_TEXT,
          PANEL_LABEL_STRING, "From :",
          PANEL_NOTIFY_PROC,  notify_copy1,
          PANEL_LAYOUT,  PANEL_HORIZONTAL,
          PANEL_VALUE_DISPLAY_LENGTH,  20,
          NULL);
   create_text_copy2 = (Panel_item)xv_create(panel_copy,PANEL_TEXT,
          PANEL_LABEL_STRING, "To :",
          PANEL_NOTIFY_PROC,  notify_copy2,
          PANEL_LAYOUT,  PANEL_HORIZONTAL,
          PANEL_VALUE_DISPLAY_LENGTH,  20,
          NULL);
   create_text_copy3 = (Panel_item)xv_create(panel_copy,PANEL_TEXT,
          PANEL_LABEL_STRING, "Start row&col to be copied :",
          PANEL_NOTIFY_PROC,  notify_copy3,
          PANEL_LAYOUT,  PANEL_HORIZONTAL,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                PANEL_VALUE_DISPLAY_LENGTH,  20,
                NULL);
    (void)xv_create(panel_copy,PANEL_BUTTON,
                PANEL_LABEL_STRING, "Copy data",
                PANEL_NOTIFY_PROC,  notify_fill_data3,
                NULL);
  /* ****************** End of copy data***************** */
    frame_move = (Frame)xv_create(frame,FRAME_CMD,
        FRAME_LABEL,    "SHEET: Move data",
        XV_WIDTH,    400,
        XV_HEIGHT,   150,
        NULL);
    panel_move = (Panel)xv_get(frame_move,FRAME_CMD_PANEL,
              PANEL_LAYOUT,   PANEL_VERTICAL,
              NULL);
     create_text_move1=(Panel_item)xv_create(panel_move,PANEL_TEXT,
                PANEL_LABEL_STRING, "From :",
                PANEL_NOTIFY_PROC,  notify_move1,
                PANEL_LAYOUT,  PANEL_HORIZONTAL,
                PANEL_VALUE_DISPLAY_LENGTH,  20,
                NULL);
     create_text_move2 = (Panel_item)xv_create(panel_move,PANEL_TEXT,
                PANEL_LABEL_STRING, "To :",
                PANEL_NOTIFY_PROC,  notify_move2,
                PANEL_LAYOUT,  PANEL_HORIZONTAL,
                PANEL_VALUE_DISPLAY_LENGTH,  20,
                NULL);
     create_text_move3 = (Panel_item)xv_create(panel_move,PANEL_TEXT,
                PANEL_LABEL_STRING, "Start row&col to be moved :",
                PANEL_NOTIFY_PROC,  notify_move3,
                PANEL_LAYOUT,  PANEL_HORIZONTAL,
                PANEL_VALUE_DISPLAY_LENGTH,  20,
                NULL);
    (void)xv_create(panel_move,PANEL_BUTTON,
                PANEL_LABEL_STRING, "Move data",
                PANEL_NOTIFY_PROC,  notify_move3,
                NULL);
  /* ****************** End of move data***************** */
   canvas = xv_create(frame,CANVAS,
          CANVAS_REPAINT_PROC, canvas_repaint,
          CANVAS_X_PAINT_WINDOW,   TRUE,
          WIN_DYNAMIC_VISUAL,      use_dynamic,
          WIN_CMS_NAME,            "palette",
          WIN_CMS_DATA,            &cms_data,
          NULL);
   icon_image = (Server_image)xv_create(NULL,SERVER_IMAGE,
          XV_WIDTH,  100,
          XV_HEIGHT, 150,
          SERVER_IMAGE_DEPTH, 1,
          SERVER_IMAGE_BITS,  icon_bitmap_bits,
          NULL);
  /* associate icon with base frame */
  icon = (Icon)xv_create(NULL,ICON,
          ICON_IMAGE,  icon_image,
          WIN_CMS_NAME,  "palette",
          NULL);
  menu = (Menu)xv_create(NULL,MENU,
         MENU_TITLE_ITEM,  "MAINMENU",
  /* ******************* menu ****************** */
         MENU_ITEM,
           MENU_STRING,   "Spreadsheet",
           MENU_NOTIFY_PROC, my_notify_proc,
           MENU_PULLRIGHT,
             xv_create(canvas,MENU,
             MENU_GEN_PIN_WINDOW, frame,"Spreadsheet",
             MENU_ITEM,
               MENU_STRING, "Load...",
               MENU_NOTIFY_PROC, load,
               NULL,
             MENU_ITEM,
               MENU_STRING, "Save...",
               MENU_NOTIFY_PROC, save,
               NULL,
             MENU_ITEM,
                MENU_STRING, "Print...",
                MENU_NOTIFY_PROC, print,
                NULL,
             MENU_ITEM,
               MENU_STRING, "Clear sheet...",
               MENU_NOTIFY_PROC, clear,
               NULL,
             NULL),
          NULL,
        MENU_ITEM,
          MENU_STRING,   "Column",
          MENU_NOTIFY_PROC, my_notify_proc,
          MENU_PULLRIGHT,
            xv_create(canvas,MENU,
            MENU_GEN_PIN_WINDOW, frame,"Column",
            MENU_ITEM,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MENU_STRING, "Insert...",
            MENU_NOTIFY_PROC, insert_col,
            NULL,
          MENU_ITEM,
            MENU_STRING, "Delete...",
            MENU_NOTIFY_PROC, delete_col,
            NULL,
          MENU_ITEM,
            MENU_STRING, "Set Width...",
            MENU_NOTIFY_PROC,  wide,
            NULL,
          NULL),
        NULL,
      MENU_ITEM,
        MENU_STRING,  "Row",
        MENU_NOTIFY_PROC, my_notify_proc,
        MENU_PULLRIGHT,
          xv_create(canvas,MENU,
            MENU_GEN_PIN_WINDOW, frame,"Row",
            MENU_ITEM,
              MENU_STRING, "Insert...",
              MENU_NOTIFY_PROC, insert_row,
              NULL,
            MENU_ITEM,
              MENU_STRING, "Delete...",
              MENU_NOTIFY_PROC, delete_row,
              NULL,
            NULL),
        NULL,
      MENU_ITEM,
        MENU_STRING,  "Utility",
        MENU_NOTIFY_PROC, my_notify_proc,
        MENU_PULLRIGHT,
          xv_create(canvas,MENU,
          MENU_GEN_PIN_WINDOW, frame,"Utility",
          MENU_ITEM,
            MENU_STRING, "Recalc...",
            MENU_NOTIFY_PROC, recalc,
            NULL,
          MENU_ITEM,
            MENU_STRING, "Formula...",
            MENU_NOTIFY_PROC, formula,
            NULL,
          MENU_ITEM,
            MENU_STRING, "Graph...",
            MENU_NOTIFY_PROC,  play,
            NULL,
          MENU_ITEM,
            MENU_STRING, "Fill data...",
            MENU_NOTIFY_PROC, fill_data,
            NULL,
          NULL),
        NULL,
      MENU_ITEM,
        MENU_STRING, "Autocalc...",
        MENU_NOTIFY_PROC, autocalc,
        NULL,
      MENU_ITEM,
        MENU_STRING, "Copy...",
        MENU_NOTIFY_PROC, copy,
        NULL,
      MENU_ITEM,
        MENU_STRING, "Move...",
        MENU_NOTIFY_PROC, move,
        NULL,
      MENU_ITEM,
        MENU_STRING, "Quit...",
        MENU_NOTIFY_PROC,  quit,
        NULL,
      NULL);
  /* ****************** set input mask ***************** */
  xv_set(canvas_paint_window(canvas),
          WIN_CONSUME_EVENTS,
                  WIN_ASCII_EVENTS,KBD_USE,KBD_DONE,
                  LOC_DRAG,LOC_MOVE,LOC_WINENTER,LOC_WINEXIT,
                  WIN_MOUSE_BUTTONS,
                  NULL,
          WIN_EVENT_PROC,  event_proc,
          WIN_CLIENT_DATA,   menu,
          NULL);
  window_fit(frame);
  xv_set(frame,FRAME_ICON,icon,NULL);
  strcpy(kbd_msg,"Keyboard : ");
  strcpy(ptr_msg,"Pointer : ");
  strcpy(but_msg,"Button : ");

  display = (Display *)xv_get(frame,XV_DISPLAY);
  xid = (XID)xv_get(canvas_paint_window(canvas),XV_XID);
  font = (Xv_Font)xv_get(frame,XV_FONT);
  font_info = (XFontStruct *)xv_get(font,FONT_INFO);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
gc_val.font = (Font)xv_get(font,XV_XID);
gc_val.stipple =
        XCreateBitmapFromData(display,xid,stipple_bits,16,2);
/* gc_val.graphics_exposures = False;*/
gc = XCreateGC(display,xid,GCFont | GCStipple,&gc_val),

colors = (unsigned long *) xv_get(canvas,WIN_X_COLOR_INDICES);

xv_main_loop(frame);
} /* ***** main loop ******** */
/* ********** initialize value ************ */
void
initvars()
{
 int i,j;
 for (i=0;i<=MAXGCOL-1;i++)
 ROW[i] = NULL;
  for (j=0;j<=MAXGCOL-1;j++)
    COL[j]= NULL;

 leftcol=toprow=rightcol=bottomrow=m=n=e=f=0;
 wd = 100;
 for (i=0;i<=20;i++)
 buf[i] = '\0';
}
/* canvas_repaint_proc is called whenever repaint is needed */
void
canvas_repaint(pw)
Xv_Window  pw;
{
 w = xv_get(pw,XV_WIDTH); /* width of window */
 h = xv_get(pw,XV_HEIGHT);/* height of window */
 drawcol();
 drawrow();
 /* fill area of acrollbar */
 XSetForeground(display,gc,colors[GRAY]);
 XFillRectangle(display,xid,gc,w-29,10,29,h-50);
 XFillRectangle(display,xid,gc,10,h-29,w-50,29);
/* draw bar's mark */
 XSetForeground(display,gc,colors[BLUE]);
 XFillRectangle(display,xid,gc,w-28,0,28,10);/* vert->start */
 XFillRectangle(display,xid,gc,w-28,h-40,28,10);/* vert->end */
 XFillRectangle(display,xid,gc,0,h-28,10,28);/* horiz ->start */
 XFillRectangle(display,xid,gc,w-40,h-28,10,28);/* horiz->end */
/* draw bar */
 XSetForeground(display,gc,colors[WHITE]);
 XFillRectangle(display,xid,gc,w-30,h-30,30,30);
 XSetForeground(display,gc,colors[RED]);
 XFillRectangle(display,xid,gc,w-28,m+10,28,20);
 XFillRectangle(display,xid,gc,n+10,h-28,20,28);
 show_data();
}

/* ************draw colunm*********** */
void
drawcol()
/*  leftcol used to check location of col that displaying */
{
char   *colstr;
int    col;
int    pcol=40; /* as pixel */
int  text_width;
int  len;
col = leftcol;
rightcol = right_col();
 while ((pcol < (w-30)) && (col < rightcol+2))
 {
 wd =  getcolwidth(col); /* wd is measured in pixel */
 colstr = getcolstr(col);
 len = strlen(colstr);
 text_width = XTextWidth(font_info,colstr,len);
 XSetForeground(display,gc,colors[BLACK]);
 XDrawString(display,xid ,gc,(pcol-text_width+wd/2),15,colstr,
             strlen(colstr));
 XSetForeground(display,gc,colors[GREEN]);
 XDrawLine(display,xid,gc,pcol,0,pcol,h-30);
 pcol += wd;
 col++;
 }
}
/* ************draw row************** */
void
drawrow()
{
int row;
int bufrow;
char rowstr[];
int prow; /* as pixel */
row = toprow;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
prow = 20;
bottomrow = toprow+( (h-50)/20)-1;
XSetForeground(display,gc,colors[GREEN]);
XDrawLine(display,xid,gc,0,prow,w-30,prow);
while ((prow < (h-30)) && (row < (bottomrow+2)) )
 {
 bufrow = row;
 sprintf(rowstr,"%d",bufrow+1);
 XSetForeground(display,gc,colors[BLACK]);
 XDrawString(display,xid,gc,3,prow+20,rowstr,strlen(rowstr));
 XSetForeground(display,gc,colors[GREEN]);
 XDrawLine(display,xid,gc,0,prow,w-30,prow);
 prow +=20;
 row ++;
 }
}
/* This function is used to calculate column's width */
/* and return column's width. */
int  getcolwidth(col)
int  col;
{
struct width *ptr;
ptr = headptr_width;            /* start link of cell */
if (ptr == NULL)
  {
  wd = defaultcolw;
  return wd;
  }
 wd = 0;
 while((wd == 0) && (ptr != NULL) ){
   if (ptr->acol < col)
     ptr = ptr->wptr;
   else
     wd++;
 }
 if(ptr == NULL)
   wd = defaultcolw;
 else {
   if(ptr->acol == col)
     wd = ptr->pcolw;
   else
     wd = defaultcolw;
  }
  return wd;
}
/* ********* get column string to draw *************** */
char *getcolstr(col)
int col;
{
char colstr[1];
 if (col <26)
    colstr[0] = col+'A';
 else
  {
   colstr[0] = (col/26)-1+'A';
   colstr[1] = (col%26)+'A';
  }
  return  colstr;
}
/*  ************** event procedure ************ */
void
event_proc(pw,event)
Xv_Window pw;
Event   *event;
{
 char c;
 int x,y;
 x = event_x(event);
 y = event_y(event);
 if (event_is_up(event))
     return;
  if (event_is_ascii(event))
  {
  editcell(pw,event);
  } else
     switch (event_action(event))
     {
      case ACTION_SELECT :
      case MS_LEFT :
       xv_set(frame,FRAME_RIGHT_FOOTER,"                    ",NULL);
       /*check first row of sheet */
       if ( (x>=w-30)&&(x<=w-1)&&(y<10)&&(y>=0) )
           {
             toprow = 0;
             m = 0;
             XSetForeground(display,gc,colors[GRAY]);
             XFillRectangle(display,xid,gc,w-29,10,29,h-50);
             XSetForeground(display,gc,colors[RED]);
             XFillRectangle(display,xid,gc,w-28,10,28,20);
             XSetForeground(display,gc,colors[WHITE]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงลิขสิทธิ์ของเอกสารทุกครั้งที่มีการนำไปใช้

```
        XFillRectangle(display,xid,gc,0,0,w-30,h-30);
        drawcol();
        drawrow();
        show_data();
        return;
      } else
    /* check last row of sheet */
    if ( (x>=w-30)&&(x<=w-1)&&(y<=h-30)&&(y>=h-40) )
      {
        bottomrow = 2000;
        m = (h-70);
        toprow = bottomrow-((h-50)/20)-1;
        XSetForeground(display,gc,colors[GRAY]);
        XFillRectangle(display,xid,gc,w-29,10,29,h-50);
        XSetForeground(display,gc,colors[RED]);
        XFillRectangle(display,xid,gc,w-28,h-60,28,20);
        XSetForeground(display,gc,colors[WHITE]);
        XFillRectangle(display,xid,gc,0,0,w-30,h-30);
        drawcol();
        drawrow();
        show_data();
        return;
      } else
    /* check first column of sheet */
    if ( (x>=0)&&(x<=10)&&(y<h-1)&&(y>h-30) )
      {
        leftcol = 0;
        n = 0;
        XSetForeground(display,gc,colors[GRAY]);
        XFillRectangle(display,xid,gc,10,h-29,w-50,29);
        XSetForeground(display,gc,colors[RED]);
        XFillRectangle(display,xid,gc,10,h-28,20,28);
        XSetForeground(display,gc,colors[WHITE]);
        XFillRectangle(display,xid,gc,0,0,w-30,h-30);
        drawcol();
        drawrow();
        show_data();
        return;
      } else
    /* check last column of sheet */
    if ( (x>=w-40)&&(x<=w-30)&&(y<h-1)&&(y>h-30) )
      {
        rightcol = 500;
        leftcol = left_col();
        n = (w-70);
        XSetForeground(display,gc,colors[GRAY]);
        XFillRectangle(display,xid,gc,10,h-29,w-50,29);
        XSetForeground(display,gc,colors[RED]);
        XFillRectangle(display,xid,gc,w-60,h-28,20,28);
        XSetForeground(display,gc,colors[WHITE]);
        XFillRectangle(display,xid,gc,0,0,w-30,h-30);
        drawcol();
        drawrow();
        show_data();
        return;
      } else
    /* check the final of vertical scrollbar */
    if ( (x>=w-28)&&(x<=w-2)&&(y>=h-50)&&(y<=h-40) )
     {
      if (bottomrow < 2000)
        {
        toprow++;
        XSetForeground(display,gc,colors[RED]);
        XFillRectangle(display,xid,gc,w-28,h-50,28,10);
        XSetForeground(display,gc,colors[WHITE]);
        XFillRectangle(display,xid,gc,0,0,w-30,h-30);
        drawcol();
        drawrow();
        show_data();
        return;
        }
      return;
     }
     /* check the location of vertical scrollbar */
    else if ( (x>=w-28)&&(x<=w-2)&&(y>=m+10)&&(y<=m+20) )
     {
      if (m+10>h-50)
          return;
      XSetForeground(display,gc,colors[GRAY]);
      XFillRectangle(display,xid,gc,w-29,10,29,h-50);
      toprow = toprow+1;
      XSetForeground(display,gc,colors[WHITE]);
      XFillRectangle(display,xid,gc,0,0,w-30,h-30);
      drawcol();
      drawrow();
      XSetForeground(display,gc,colors[RED]);
      XFillRectangle(display,xid,gc,w-28,m+10,28,20);
      m = m+5;
      show_data();
     return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

```
}
/* check the final location of horizontal scrollbar */
else if ( (x>=w-50)&&(x<=w-40)&&(y>=h-28)&&(y<=h-2) )
   {
     if (rightcol<500)
     {
      leftcol++;
      XSetForeground(display,gc,colors[RED]);
      XFillRectangle(display,xid,gc,w-50,h-28,20,28);
      XSetForeground(display,gc,colors[WHITE]);
      XFillRectangle(display,xid,gc,0,0,w-30,h-30);
      drawcol();
      drawrow();
      show_data();
      return;
     }
    return;
   }
 /* check the location of horizontal scrollbar */
 else if ( (x>=n+10)&&(x<=n+20)&&(y>=h-28)&&(y<=h-2) )
  {
   if (n+10>w-50)
       return;
   XSetForeground(display,gc,colors[GRAY]);
   XFillRectangle(display,xid,gc,10,h-29,w-50,29);
   leftcol = leftcol+1;
   XSetForeground(display,gc,colors[WHITE]);
   XFillRectangle(display,xid,gc,0,0,w-30,h-30);
   drawcol();
   drawrow();
   XSetForeground(display,gc,colors[RED]);
   XFillRectangle(display,xid,gc,n+10,h-28,20,28);
   n = n+5;
   show_data();
   return;
  }
  else if( (x>w-30)&&(x<w)&&(y>10)&&(y<h-50) )
    {
       if ( bottomrow == 2000 )
           return;
       if (bottomrow >=2000-(h-50)/20)
         {
           bottomrow = 2000;
           toprow = bottomrow+((h-50)/20)+1;
           m = h-70;
           XSetForeground(display,gc,colors[GRAY]);
           XFillRectangle(display,xid,gc,w-29,10,29,h-50);
           XSetForeground(display,gc,colors[RED]);
           XFillRectangle(display,xid,gc,w-28,h-60,28,20);
           XSetForeground(display,gc,colors[WHITE]);
           XFillRectangle(display,xid,gc,0,0,w-30,h-30);
           drawcol();
           drawrow();
           show_data();
           return;
         }
         toprow = bottomrow;
         bottomrow = toprow+((h-50)/20)+1;
         /* m = m+5*( ((h-70)/20)-1);*/
         m = m+( (bottomrow-toprow-1)*(h-70) )/2000;
         XSetForeground(display,gc,colors[GRAY]);
         XFillRectangle(display,xid,gc,w-29,10,29,h-50);
         XSetForeground(display,gc,colors[RED]);
         XFillRectangle(display,xid,gc,w-28,m+10,28,20);
         XSetForeground(display,gc,colors[WHITE]);
         XFillRectangle(display,xid,gc,0,0,w-30,h-30);
         drawcol();
         drawrow();
         show_data();
         return;
    }
  else if( (x>10)&&(x<w-50)&&(y>h-30)&&(y<h) )
    {
       if ( rightcol == 500 )
           return;
       if ( rightcol >= 500-(w-70)/100 )
         {
           rightcol = 500;
           leftcol = left_col();
           n = w-70;
           XSetForeground(display,gc,colors[GRAY]);
           XFillRectangle(display,xid,gc,10,h-29,w-50,29);
           XSetForeground(display,gc,colors[RED]);
           XFillRectangle(display,xid,gc,w-70,h-28,20,28);
           XSetForeground(display,gc,colors[WHITE]);
           XFillRectangle(display,xid,gc,0,0,w-30,h-30);
           drawcol();
           drawrow();
           show_data();
           return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

```
            }
            leftcol  = rightcol;
            rightcol = right_col();
            n = n+( ( (w-70)*(rightcol-leftcol-1) )/500);
            XSetForeground(display,gc,colors[GRAY]);
            XFillRectangle(display,xid,gc,10,h-29,w-50,29);
            XSetForeground(display,gc,colors[RED]);
            XFillRectangle(display,xid,gc,n+10,h-28,20,28);
            XSetForeground(display,gc,colors[WHITE]);
            XFillRectangle(display,xid,gc,0,0,w-30,h-30);
            drawcol();
            drawrow();
            show_data();
            return;
        }
      else if ( (x>=0)&&(x<40)&&(y>=0)&&(y<20) )
        {
         if (e <= 3 )
           {
            switch (e)
             {
             case  0:
              /* expose_01();*/
               e ++;
               break;
             case 1:
               /* expose_02();*/
               e++;
               break;
             case 2:
               /* expose_03();*/
               e++;
               return;
             case 3:
               /* expose_04();*/
                e++;
                break;
               }
              return;
           }
         e = 0;
         XSetForeground(display,gc,colors[GRAY]);
         XFillRectangle(display,xid,gc,w-29,10,29,h-50);
         XSetForeground(display,gc,colors[WHITE]);
         XFillRectangle(display,xid,gc,0,0,w-30,h-30);
         drawcol();
         drawrow();
         return;
        }
        else
        displaycell(pw,event);
        break;
case ACTION_ADJUST :
case MS_MIDDLE :
    break;
case ACTION_MENU :
case MS_RIGHT :
 /* check the start of vertical scrollbar */
 if ( (x>=w-28)&&(x<=w-2) && (y>=10)&&(y<=20) )
   {
    if (toprow>0)
     {
      toprow--;
      XSetForeground(display,gc,colors[RED]);
      XFillRectangle(display,xid,gc,w-28,10,28,20);
      XSetForeground(display,gc,colors[WHITE]);
      XFillRectangle(display,xid,gc,0,0,w-30,h-30);
      drawcol();
      drawrow();
      show_data();
      return;
     }
    return;
   }
    /* check to backward  vertical */
  else if ( (x>=w-28)&&(x<=w-2)&&(y>=m+10) && (y<=m+20) )
    {
     m = m-5;
     toprow = toprow-1;
     XSetForeground(display,gc,colors[GRAY]);
     XFillRectangle(display,xid,gc,w-29,10,29,h-50);
     XSetForeground(display,gc,colors[WHITE]);
     XFillRectangle(display,xid,gc,0,0,w-30,h-30);
     drawcol();
     drawrow();
     XSetForeground(display,gc,colors[RED]);
     XFillRectangle(display,xid,gc,w-28,m+10,28,20);
     show_data();
     return;
    }
```

เอกสารนี้เป็นเอกสาร... ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ... เอกสารทุกครั้งที่มีการนำไปใช้

```
/* check the start of horizontal scrollbar */
else if ((x>=10)&&(x<=20)&&(y>=h-28)&&(y<=h-2))
    {
     if (leftcol>0)
      {
       leftcol--;
       XSetForeground(display,gc,colors[RED]);
       XFillRectangle(display,xid,gc,10,h-28,20,28);
       XSetForeground(display,gc,colors[WHITE]);
       XFillRectangle(display,xid,gc,0,0,w-30,h-30);
       drawcol();
       drawrow();
       show_data();
       return;
      }
     return;
     }
    else if ( (x>=n+10)&&(x<=n+20)&&(y>=h-28)&&(y<=h-2) )
     /* check backward of horizontal */
     {
      n = n-5;
      leftcol = leftcol-1;
      XSetForeground(display,gc,colors[GRAY]);
      XFillRectangle(display,xid,gc,10,h-29,w-50,29);
      XSetForeground(display,gc,colors[WHITE]);
      XFillRectangle(display,xid,gc,0,0,w-30,h-30);
      drawcol();
      drawrow();
      XSetForeground(display,gc,colors[RED]);
      XFillRectangle(display,xid,gc,n+10,h-28,20,28);
      show_data();
      return;
     }
     else if( (x>w-30)&&(x<w)&&(y>10)&&(y<h-50) )
      {
       if ( toprow  == 0 )
         return;
       if (toprow <= (h-50)/20)
        {
         toprow = 0;
         bottomrow = toprow+((h-50)/20)+1;
         m = 0;
         XSetForeground(display,gc,colors[GRAY]);
         XFillRectangle(display,xid,gc,w-29,10,29,h-50);
         XSetForeground(display,gc,colors[RED]);
         XFillRectangle(display,xid,gc,w-28,10,28,20);
         XSetForeground(display,gc,colors[WHITE]);
         XFillRectangle(display,xid,gc,0,0,w-30,h-30);
         drawcol();
         drawrow();
         show_data();
         return;
        }
         bottomrow  = toprow;
         toprow = bottomrow-((h-50)/20)-1;
         m = m-( (bottomrow-toprow-1)*(h-70) )/2000;
         /* m = m-5*( ( (h-70)/20)-1 );*/
         XSetForeground(display,gc,colors[GRAY]);
         XFillRectangle(display,xid,gc,w-29,10,29,h-50);
         XSetForeground(display,gc,colors[RED]);
         XFillRectangle(display,xid,gc,w-28,m+10,28,20);
         XSetForeground(display,gc,colors[WHITE]);
         XFillRectangle(display,xid,gc,0,0,w-30,h-30);
         drawcol();
         drawrow();
         show_data();
         return;
      }
else if( (x>10)&&(x<w-50)&&(y>h-30)&&(y<h) )
 {
    if ( leftcol == 0 )
      return;
    if ( rightcol <= (w-70)/100 )
      {
       leftcol = 0;
       rightcol = right_col();
       n = 0;
       XSetForeground(display,gc,colors[GRAY]);
       XFillRectangle(display,xid,gc,10,h-29,w-50,29);
       XSetForeground(display,gc,colors[RED]);
       XFillRectangle(display,xid,gc,10,h-28,20,28);
       XSetForeground(display,gc,colors[WHITE]);
       XFillRectangle(display,xid,gc,0,0,w-30,h-30);
       drawcol();
       drawrow();
       show_data();
       return;
      }
      rightcol = leftcol;
      leftcol = left_col();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    n = n-( ( (w-70)*(rightcol-leftcol-1) )/500);
                    XSetForeground(display,gc,colors[GRAY]);
                    XFillRectangle(display,xid,gc,10,h-29,w-50,29);
                    XSetForeground(display,gc,colors[RED]);
                    XFillRectangle(display,xid,gc,n+10,h-28,20,28);
                    XSetForeground(display,gc,colors[WHITE]);
                    XFillRectangle(display,xid,gc,0,0,w-30,h-30);
                    drawcol();
                    drawrow();
                    show_data();
                    return;
                }
                      else
                       {
                         menu = (Menu)xv_get(pw,WIN_CLIENT_DATA);
                         menu_show(menu,pw,event,NULL);
                       }
                     break;
                  case SHIFT_RIGHT :
                     break;
                  case SHIFT_LEFT :
                     break;
                  case SHIFT_LEFTCTRL :
                     break;
                  case SHIFT_RIGHTCTRL:
                     break;
                  case SHIFT_META :
                     break;
                  case SHIFT_ALT :
                     break;
                  case KBD_USE:
                     break;
                  case KBD_DONE:
                     break;
                  case LOC_MOVE:
                     break;
                  case LOC_DRAG:
                     break;
                  case LOC_WINENTER:
                     break;
                  case LOC_WINEXIT:
                     break;
                  default:
                     return;
               }/* switch */
}/* if */
/*  ********display cell of spreadsheet and report ***  */
void
displaycell(pw,event)
Xv_Window  pw;
Event   *event;
{
 char *colstr;
 y = 20*(srow-toprow+1);
 x = 40;
 curcol = leftcol;
 while (curcol < scol)
  {
   wd = getcolwidth(curcol);
   x += wd;
   curcol++;
  }
 XSetForeground(display,gc,colors[WHITE]);
 XFillRectangle(display,xid,gc,x+1,y+1,getcolwidth(scol)-1,19);
 XSetForeground(display,gc,colors[BLACK]);
 show_cell();
 srow = checkrow(event);
 scol = checkcol(event);
 y = 20*(srow-toprow+1);
 x = 40;
 curcol = leftcol;
 while (curcol < scol)
  {
   wd = getcolwidth(curcol);
   x += wd;
   curcol++;
  }
 first_site = x;
XSetForeground(display,gc,colors[WHITE]);
XFillRectangle(display,xid,gc,x+1,y+1,getcolwidth(scol)-1,19);
XSetForeground(display,gc,colors[RED]);
buffer[dex] = '\0';
XDrawRectangle(display,xid,gc,x+1,y+1,getcolwidth(scol)-2,18);
XSetForeground(display,gc,colors[ORCHID]);
XDrawLine(display,xid,gc,x+4,y+5,x+4,y+15);
dex = 0;
colstr = getcolstr(scol);
sprintf(buf,"%s,%d",colstr,srow+1);
xv_set(frame,FRAME_LEFT_FOOTER,buf,NULL);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานเมื่อมีการนำไปใช้

```
 /* *********  check row and return row **********/
int
checkrow(event)
Event  *event;
{
 int y,currow,pre;
 currow = toprow;
 y = event_y(event);
 pre = 20;
 if (y<pre)
    return;
 pre += 20;
 while (y>pre)
 {
  currow++;
  pre +=20;
 }
 return currow;
}
/*  *** check column and return column *** */
int
checkcol(event)
Event   *event;
{
int x,curcol,pce,wd;
x = event_x(event);
curcol = leftcol;
wd = getcolwidth(curcol);
pce = 40;
if (x< pce)
   return;
pce+= wd;
while (x>pce)
 {
 curcol++;
 wd = getcolwidth(curcol);
 pce += wd;
 }
 return curcol;
}
/* **********edit cell ******* */
void
editcell(pw,event)
Xv_Window  pw;
Event   *event;
{
 char buf_scan[];
 c = (char )event_id(event);
  buffer[dex] = c;
  dex++;
  buffer[dex] = '\0';
  if (c == 7 || c == 127) /* backspace or delete */
    {
     if (x > first_site )
     {
      x -= XTextWidth(font_info,&buffer[dex-2],1);
     /*  x -= 4; */
     /* use XDrawImageString to overwrinte previous text */
      XSetForeground(display,gc,colors[WHITE]);
     /*  XDrawImageString(display,xid,gc,x+4,y+15," ",1);*/
      XFillRectangle(display,xid,gc,x-XTextWidth(font_info,&buffer[dex-2]-4,1),
            y+2,XTextWidth(font_info,&buffer[dex-2],1),16);
      dex -= 2;
      buffer[dex] = '\0';
     }
    }
     else if (c != '\r')
      {
       /* Advance x to next position, If over cell width. */
        XSetForeground(display,gc,colors[WHITE]);
        XDrawLine(display,xid,gc,x+4,y+5,x+4,y+15);
        XSetForeground(display,gc,colors[BLACK]);
        XDrawString(display,xid,gc,x+4,y+15,&c,1);
        x += XTextWidth(font_info,&c,1);
        XSetForeground(display,gc,colors[ORCHID]);
        XDrawLine(display,xid,gc,x+4,y+5,x+4,y+15);
      }
      else{
        dex--;
        buffer[dex] = '\0';
        alloctext(srow,scol,buffer);
        sprintf(buf_test,"%s","Text");
        xv_set(frame,FRAME_RIGHT_FOOTER,buf_test,NULL);
      }
}
/* ******     notify about menu      ****** */
void
my_notify_proc(menu,menu_item)
Menu  menu;
Menu_item  menu_item;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

```
{
 xv_set(frame,
    FRAME_LABEL, xv_get(menu_item,MENU_STRING),
    NULL);
}
/* define vareable */
char str1[6],str2[6],str3[6],str4[6],str5[6];
/* *****   load filename and notify event of load ***** */
void
load()
{
xv_set(frame_load,XV_SHOW,TRUE,NULL);
}
int
notify_load(item,event)
Panel_item   item;
Event   *event;
{
 char str[80];
 sprintf(str,"%s",(char *)xv_get(create_text_load,PANEL_VALUE));
/*  xv_set(create_text_load,PANEL_VALUE," ",NULL);*/
 xv_set(frame_load,XV_SHOW,FALSE,NULL);
 loadsheet(str);
 return XV_OK;
}
/* ***** clear all of sheet ***** */
void
clear()
{
  clearsheet();
  show_data();
}
/* ***** exit to unix system ***** */
void
system()
{
   xv_destroy_safe(frame);
}
/*  ***** delete column & notify event of delete column ***** */
void
delete_col()
{
xv_set(frame_delete_col,XV_SHOW,TRUE,NULL);
}

int
notify_delete_col(item,event)
Panel_item   item;
Event   *event;
{
 char str[2];
 int col;
 sprintf(str,"%s",(char *)xv_get(create_text_deletecol,PANEL_VALUE));
 col = col_to_int(str);
 xv_set(create_text_deletecol,PANEL_VALUE," ",NULL);
 xv_set(frame_delete_col,XV_SHOW,FALSE,NULL);
 deletecol(col);
 return XV_OK;
}
/*  ***** delete row & notify event of delete row ***** */
void
delete_row()
{
xv_set(frame_delete_row,XV_SHOW,TRUE,NULL);
}

int
notify_delete_row(item,event)
Panel_item   item;
Event   *event;
{
 char str[4];
 int row;
 sprintf(str,"%s",(char *)xv_get(create_text_deleterow,PANEL_VALUE));
 row = atoi(str)-1;
 xv_set(create_text_deleterow,PANEL_VALUE," ",NULL);
 xv_set(frame_delete_row,XV_SHOW,FALSE,NULL);
 deleterow(row);
 return XV_OK;
}
/* ***** insert row & notify event of insert row ***** */
void
insert_row()
{
xv_set(frame_insert_row,XV_SHOW,TRUE,NULL);
}
int
notify_insert_row(item,event)
Panel_item   item;
Event   *event;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
 char str[4];
 int row;
 sprintf(str,"%s",(char *)xv_get(create_text_insertrow,PANEL_VALUE));
 row = atoi(str) - 1;
 /* brint name to process */
 xv_set(create_text_insertrow,PANEL_VALUE," ",NULL);
 xv_set(frame_insert_row,XV_SHOW,FALSE,NULL);
 insertrow(row);
 return XV_OK;
}
/* ***** insert column & notify event of insert column ***** */
void
insert_col()
{
xv_set(frame_insert_col,XV_SHOW,TRUE,NULL);
}
int
notify_insert_col(item,event)
Panel_item   item;
Event   *event;
{
 char str[2];
 int col;
 sprintf(str,"%s",(char *)xv_get(create_text_insertcol,PANEL_VALUE));
 col = col_to_int(str);
 xv_set(create_text_insertcol,PANEL_VALUE," ",NULL);
 xv_set(frame_insert_col,XV_SHOW,FALSE,NULL);
 insertcol(col);
 return XV_OK;
}
void
recalc()
{
   xv_destroy_safe(frame);
}

void
autocalc()
{
   xv_destroy_safe(frame);
}
/* set column's width */
void
wide()
{
xv_set(frame_width,XV_SHOW,TRUE,NULL);
}
int
notify_width1(item,event)
Panel_item   item;
Event   *event;
{
 sprintf(str1,"%s",(char *)xv_get(create_text_width1,PANEL_VALUE));
 return XV_OK;
}
int
notify_width2(pw,item,event)
Xv_Window  pw;
Panel_item   item;
Event   *event;
{
 int col_width,col;
 col = col_to_int(str1);
 sprintf(str2,"%s",(char *)xv_get(create_text_width2,PANEL_VALUE));
 col_width = atoi(str2);
 xv_set(create_text_width2,PANEL_VALUE," ",NULL);
 xv_set(frame_width,XV_SHOW,FALSE,NULL);
 setcolwidth(pw,col,col_width);
 return XV_OK;
}
void
play()
{
 XSetForeground(display,gc,colors[WHITE]);
 XFillRectangle(display,xid,gc,0,0,w-30,h-30);
 /* expose_04();*/
}

void
edit()
{
   xv_destroy_safe(frame);
}
void
formula()
{
   xv_destroy_safe(frame);
}
/* ***** print sheet ***** */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void
print()
{
xv_set(frame_print,XV_SHOW,TRUE,NULL);
}

int
notify_print(item,event)
Panel_item   item;
Event   *event;
{
 char str[20];
 sprintf(str,"%s",(char *)xv_get(create_text_print,PANEL_VALUE));
 xv_set(create_text_print,PANEL_VALUE,"",NULL);
 xv_set(frame_print,XV_SHOW,FALSE,NULL);
 /* to mam */
 return XV_OK;
}
/* quit from spreadsheet */
int
quit()
{
   xv_destroy_safe(frame);
   return XV_OK;
}
/* to save sheet */
void
save()
{
xv_set(frame_save,XV_SHOW,TRUE,NULL);
}
Xv_opaque
notify_save(item,event,client,status)
Panel_item   item;
Event   *event;
Notify_client  client;
Destroy_status  status;
{
 char str[MAXINPUT+1];
 int  result;
 int xx,yy;
 Panel panel  =(Panel)xv_get(item,PANEL_PARENT_PANEL);
 Window win;
 Window root = (Window)xv_get(xv_get(panel,XV_ROOT),XV_XID);
 sprintf(str,"%s",(char *)xv_get(create_text_save,PANEL_VALUE));
 if (!access(str,0))
 {
  XTranslateCoordinates(xv_get(panel,XV_DISPLAY),
      xv_get(panel,XV_XID),root,event_x(event),event_y(event),
      &xx,&yy,&win);
  result = notice_prompt(panel,NULL,
      NOTICE_MESSAGE_STRINGS, "File Exists * Overwrite it?",NULL,
      NOTICE_FOCUS_XY,  xx,yy,
      NOTICE_BUTTON_YES, "SAVE",
      NOTICE_BUTTON_NO,  "CANCEL",
      NULL);
    if (result == NOTICE_YES)
      savesheet(str);
  /* xv_set(create_text_save,PANEL_VALUE," ",NULL);*/
  xv_set(frame_save,XV_SHOW,FALSE,NULL);
  return XV_OK;
 }
 else
   savesheet(str);
}
void
format()
{
xv_set(frame_format,XV_SHOW,TRUE,NULL);
}
int
notify_format(item,event)
Panel_item   item;
Event   *event;
{
 char str[20];
 sprintf(str,"%s",(char *)xv_get(create_text_format,PANEL_VALUE));
 xv_set(create_text_format,PANEL_VALUE," ",NULL);
 xv_set(frame_format,XV_SHOW,FALSE,NULL);
 /* to mam */
 return XV_OK;
}
/* go to specified cell  & notify event of go to cell */
void
go_to()
{
xv_set(frame_goto,XV_SHOW,TRUE,NULL);
}
int
notify_goto(item,event)
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Panel_item   item;
Event   *event;
{
 char str[6];
 sprintf(str,"%s",(char *)xv_get(create_text_goto,PANEL_VALUE));
 parse(str);
 xv_set(create_text_goto,PANEL_VALUE," ",NULL);
 xv_set(frame_goto,XV_SHOW,FALSE,NULL);
 /* print name to process */
 return XV_OK;
}
void
/*  ***** routine for fill data ***** */
fill_data()
{
xv_set(frame_fill_data,XV_SHOW,TRUE,NULL);
}
int
notify_fill_data1(item,event)
Panel_item   item;
Event   *event;
{
 sprintf(str1,"%s",(char *)xv_get(create_text1,PANEL_VALUE));
 return XV_OK;
}
int
notify_fill_data2(item,event)
Panel_item   item;
Event   *event;
{
 sprintf(str2,"%s",(char *)xv_get(create_text2,PANEL_VALUE));
 return XV_OK;
}
int
notify_fill_data3(item,event)
Panel_item   item;
Event   *event;
{
 sprintf(str3,"%s",(char *)xv_get(create_text3,PANEL_VALUE));
 return XV_OK;
}
int
notify_fill_data4(item,event)
Panel_item   item;
Event   *event;
{
 sprintf(str4,"%s",(char *)xv_get(create_text4,PANEL_VALUE));
 return XV_OK;
}
int
notify_fill_data5(item,event)
Panel_item   item;
Event   *event;
{
 int col[3],i,j,first_val,end_val;
 int row[3];
 int range;
 char str[6];
 char buf_row[6];
 sprintf(str5,"%s",(char *)xv_get(create_text5,PANEL_VALUE));
 first_val = atoi(str3);
 end_val = atoi(str4);
 range = atoi(str5);
 col[0]=parse(str1);
 col[1]=parse(str2);
 for (i=0;i< 2;i++)
 {
   switch (i)
   {
   case 0:
     strcpy(str,str1);
     break;
   case 1:
     strcpy(str,str2);
     break;
   };
 if (col[i]>25)
   {
   if (str[2]==NULL)
     {
      sprintf(buf_error,"%c", "Specified Error!");
      xv_set(frame,FRAME_RIGHT_FOOTER,buf_error,NULL);
      return;
     }
   else if (str[3]== NULL)
        {
         sprintf(buf_row,"%c",str[2]);
         row[i] = atoi(buf_row)-1;
        }
        else if (str[4] == NULL)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
             sprintf(buf_row,"%c%c",str[2],str[3]);
             row[i] = atoi(buf_row)-1;
            }
            else if (str[5]== NULL)
                {
                 sprintf(buf_row,"%c%c%c",str[2],str[3],str[4]);
                 row[i] = atoi(buf_row)-1;
                }
                else
                 {
                 sprintf(buf_row,"%c%c%c%c",str[2],str[3],str[4],str[5]);
                 row[i] = atoi(buf_row)-1;
                 }
  }
    else if (str[2] == NULL)
          {
           sprintf(buf_row,"%c%c",str[1],str[2]);
           row[i] = atoi(buf_row)-1;
          }
          else if (str[3] == NULL)
              {
               sprintf(buf_row,"%c%c%c%c",str[1],str[2],str[3]);
               row[i] = atoi(buf_row)-1;
              }
              else if (str[4] == NULL)
                  {
                   sprintf(buf_row,"%c%c%c%c",str[1],str[2],str[3],str[4]);
                   row[i] = atoi(buf_row)-1;
                  }
                  else if (str[5]== NULL)
                   {
                   sprintf(buf_row,"%c%c%c%c",str[1],str[2],str[3],str[4]);
                   row[i] = atoi(buf_row)-1;
                   }
                   else
                     {
                     sprintf(buf_row,"%c%c%c%c%c",str[1],str[2],str[3],str[4],str[5]);
                     row[i] = atoi(buf_row)-1;
                     }
   }
  xv_set(create_text5,PANEL_VALUE," ",NULL);
  xv_set(frame_fill_data,XV_SHOW,FALSE,NULL);
  fill(col[0],row[0],col[1],row[1],first_val,end_val,range);
  return XV_OK;
}
void
copy()
{
xv_set(frame_copy,XV_SHOW,TRUE,NULL);
}
/* ****** routine for copy data ***** */
int
notify_copy1(item,event)
Panel_item   item;
Event   *event;
{
 sprintf(str1,"%s",(char *)xv_get(create_text_copy1,PANEL_VALUE));

 /* xv_set(create_text_copy1,PANEL_VALUE," ",NULL);*/
 return XV_OK;
}
int
notify_copy2(item,event)
Panel_item   item;
Event   *event;
{
 sprintf(str2,"%s",(char *)xv_get(create_text_copy2,PANEL_VALUE));
/*  xv_set(create_text_copy2,PANEL_VALUE," ",NULL);*/
 return XV_OK;
}
int
notify_copy3(item,event)
Panel_item   item;
Event   *event;
{
 int col[3],i,j;
 int row[3];
 char str[6];
 char buf_row[6];
 sprintf(str3,"%s",(char *)xv_get(create_text_copy3,PANEL_VALUE));
 col[0]=parse(str1);
 col[1]=parse(str2);
 col[2]=parse(str3);
 for (i=0;i< 3;i++)
 {
   switch (i)
   {
   case 0:
     strcpy(str,str1);
```

เอกสารนี้ใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;
      case 1:
        strcpy(str,str2);
        break;
      case 2:
        strcpy(str,str3);
        break;
      };
    if (col[i]>25)
      {
      if (str[2]==NULL)
        {
         sprintf(buf_error,"%c", "Specified Error!");
         xv_set(frame,FRAME_RIGHT_FOOTER,buf_error,NULL);
         return;
        }
      else if (str[3]== NULL)
           {
            sprintf(buf_row,"%c",str[2]);
            row[i] = atoi(buf_row)-1;
           }
           else if (str[4] == NULL)
              {
               sprintf(buf_row,"%c%c",str[2],str[3]);
               row[i] = atoi(buf_row)-1;
              }
              else if (str[5]== NULL)
                 {
                  sprintf(buf_row,"%c%c%c",str[2],str[3],str[4]);
                  row[i] = atoi(buf_row)-1;
                 }
                 else
                  {
                  sprintf(buf_row,"%c%c%c%c",str[2],str[3],str[4],str[5]);
                  row[i] = atoi(buf_row)-1;
                  }
    }
      else if (str[2] == NULL)
           {
            sprintf(buf_row,"%c%c",str[1],str[2]);
            row[i] = atoi(buf_row)-1;
           }
           else if (str[3] == NULL)
              {
               sprintf(buf_row,"%c%c%c%c",str[1],str[2],str[3]);
               row[i] = atoi(buf_row)-1;
              }
              else if (str[4] == NULL)
                 {
                  sprintf(buf_row,"%c%c%c%c",str[1],str[2],str[3],str[4]);
                  row[i] = atoi(buf_row)-1;
                 }
                 else if (str[5]== NULL)
                  {
                  sprintf(buf_row,"%c%c%c%c",str[1],str[2],str[3],str[4]);
                  row[i] = atoi(buf_row)-1;
                  }
                  else
                    {
                    sprintf(buf_row,"%c%c%c%c%c",str[1],str[2],str[3],str[4],str[5]);
                    row[i] = atoi(buf_row)-1;
                    }
    }
   xv_set(create_text_copy3,PANEL_VALUE," ",NULL);
   xv_set(frame_copy,XV_SHOW,FALSE,NULL);
   copydata(col[0],row[0],col[1],row[1],col[2],row[2]);
   return XV_OK;
  } /* end of copy */
  /* ******  routine for move data ******* */
  void
  move()
  {
  xv_set(frame_move,XV_SHOW,TRUE,NULL);
  }
  int
  notify_move1(item,event)
  Panel_item   item;
  Event   *event;
  {
   sprintf(str1,"%s",(char *)xv_get(create_text_move1,PANEL_VALUE));
   return XV_OK;
  }
  int
  notify_move2(item,event)
  Panel_item   item;
  Event   *event;
  {
   sprintf(str2,"%s",(char *)xv_get(create_text_move2,PANEL_VALUE));
   return XV_OK;
  }
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int
notify_move3(item,event)
Panel_item   item;
Event   *event;
{
 int col[3],i,j;
 int row[3];
 char str[6];
 char buf_row[6];
 sprintf(str3,"%s",(char *)xv_get(create_text_move3,PANEL_VALUE));
 col[0]=parse(str1);
 col[1]=parse(str2);
 col[2]=parse(str3);
 for (i=0;i< 3;i++)
 {
   switch (i)
   {
   case 0:
     strcpy(str,str1);
     break;
   case 1:
     strcpy(str,str2);
     break;
   case 2:
     strcpy(str,str3);
     break;
   };
 if (col[i]>25)
   {
   if (str[2]==NULL)
     {
      sprintf(buf_error,"%c", "Specified Error!");
      xv_set(frame,FRAME_RIGHT_FOOTER,buf_error,NULL);
      return;
     }
   else if (str[3]== NULL)
        {
         sprintf(buf_row,"%c",str[2]);
         row[i] = atoi(buf_row)-1;
        }
        else if (str[4] == NULL)
           {
            sprintf(buf_row,"%c%c",str[2],str[3]);
            row[i] = atoi(buf_row)-1;
           }
           else if (str[5]== NULL)
              {
               sprintf(buf_row,"%c%c%c",str[2],str[3],str[4]);
               row[i] = atoi(buf_row)-1;
              }
              else
               {
               sprintf(buf_row,"%c%c%c%c",str[2],str[3],str[4],str[5]);
               row[i] = atoi(buf_row)-1;
               }
 }
   else if (str[2] == NULL)
        {
         sprintf(buf_row,"%c%c",str[1],str[2]);
         row[i] = atoi(buf_row)-1;
        }
        else if (str[3] == NULL)
           {
            sprintf(buf_row,"%c%c%c%c",str[1],str[2],str[3]);
            row[i] = atoi(buf_row)-1;
           }
           else if (str[4] == NULL)
              {
               sprintf(buf_row,"%c%c%c%c",str[1],str[2],str[3],str[4]);
               row[i] = atoi(buf_row)-1;
              }
              else if (str[5]== NULL)
               {
               sprintf(buf_row,"%c%c%c%c",str[1],str[2],str[3],str[4]);
               row[i] = atoi(buf_row)-1;
               }
               else
                 {
                  sprintf(buf_row,"%c%c%c%c%c",str[1],str[2],str[3],str[4],str[5]
                  row[i] = atoi(buf_row)-1;
                 }
   }
 xv_set(create_text_move3,PANEL_VALUE," ",NULL);
 xv_set(frame_move,XV_SHOW,FALSE,NULL);
 move_data(col[0],row[0],col[1],row[1],col[2],row[2]);
 return XV_OK;
}/*  end of move */

void
delete_all()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
/* delete all cell */
xv_destroy_safe(frame);
}
void
vert_normal(event)
Event  *event;
{
}
void
horiz_normal(event)
Event  *event;
{

}
void
scrollbar_default_compute_scroll_pro()
{
}
int
find_x(col)
{
 x = 40;
 curcol = leftcol;
 while (curcol < col)
  {
   wd = getcolwidth(curcol);
   x += wd;
   curcol++;
  }
 return x;
}

void show_data()
{
 int row,col,xx,yy,out,len,text_width;
 char bufint[MAXINPUT+1];
struct node *bufinfo;
/*int rightcol = 6;*/
/* int bottomrow = 26;*/
  /*find where data is */
  for (col= leftcol;col<=rightcol;col++)
  {
   for (row = toprow; row<= bottomrow;row++)
   {
     xx = find_x(col);
     yy = 20*(row-toprow+1);
     XSetForeground(display,gc,colors[WHITE]);
     XFillRectangle(display,xid,gc,xx+1,yy+1,getcolwidth(col)-2,18);
     bufinfo = search(row,col);
        if (bufinfo != NULL)
        {
         switch( bufinfo->attrib ){
        case 'T':
            XSetForeground(display,gc,colors[BLACK]);
            XDrawString(display,xid,gc,xx+4,yy+15,bufinfo->v.text,
                strlen(bufinfo->v.text));
             break;
        case 'V':
            XSetForeground(display,gc,colors[BLACK]);
            sprintf(bufint,"%d",bufinfo->v.value);
            len = strlen(bufint);
            wd = getcolwidth(col);
            text_width = XTextWidth(font_info,bufint,len);
            XDrawString(display,xid,gc,xx+(wd-text_width)-2,yy+15,
                bufint,len);
             break;
        case 'F':
            XSetForeground(display,gc,colors[BLACK]);
            XDrawString(display,xid,gc,xx+4,yy+15,bufinfo->v.f.fvalue,
                strlen(bufinfo->v.f.fvalue));
             break;
        default:
             break;
        }
     }
    }
  }
}
int
right_col()
{
 int col,sum_wd;
 col = leftcol;
 sum_wd =0;
 while (sum_wd <= (w-70))
 {
  sum_wd += getcolwidth(col);
  col++;
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    col -= 2;
    return col;
}
/*  ********** */
int left_col()
{
    int col,sum_wd;
    col = rightcol;
    sum_wd = 0;
    while (sum_wd <= (w-70))
    {
     sum_wd += getcolwidth(col);
     col--;
    }
     col+=2;
     return col;
}

void show_cell()
{
 int xx,yy,out,len,text_width;
 char bufint[MAXINPUT+1];
struct node *bufinfo;
  /*find where data is */
     xx = find_x(scol);
     yy = 20*(srow-toprow+1);
     XSetForeground(display,gc,colors[WHITE]);
     XFillRectangle(display,xid,gc,xx+1,yy+1,getcolwidth(scol)-2,18);
     bufinfo = search(srow,scol);
        if (bufinfo != NULL)
        {
         switch( bufinfo->attrib ){
        case 'T':
            XSetForeground(display,gc,colors[BLACK]);
            XDrawString(display,xid,gc,xx+4,yy+15,bufinfo->v.text,
                strlen(bufinfo->v.text));
             break;
        case 'V':
            XSetForeground(display,gc,colors[BLACK]);
            sprintf(bufint,"%d",bufinfo->v.value);
            len = strlen(bufint);
            wd = getcolwidth(scol);
            text_width = XTextWidth(font_info,bufint,len);
            XDrawString(display,xid,gc,xx+(wd-text_width)-2,yy+15,
                bufint,len);
             break;
        case 'F':
            XSetForeground(display,gc,colors[BLACK]);
            XDrawString(display,xid,gc,xx+4,yy+15,bufinfo->v.f.fvalue,
                strlen(bufinfo->v.f.fvalue));
             break;
        default:
            break;
        }
  }
}

struct node *search(row,col)
int row,col;
{
  struct group *ptr;
  struct node *bufinfo;
  int out;
     ptr = ROW[row/FROW];
     out = 0;
     while( (ptr != NULL) && ( out != 1) )
     {
        if ( ptr->cid < (col/FCOL) )
          ptr = ptr->right;
        else
          out = 1;
     }
        /*get data to show*/
      if (ptr != NULL) {
        if (ptr->cid == (col/FCOL))
          bufinfo = ptr->info[row%FROW][col%FCOL];
        else
          bufinfo = NULL;
      }
     else
        bufinfo = NULL;
    return bufinfo;
}

/* ***** end ***** */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* color.h -to define color */

#define WHITE   0
#define RED     1
#define GREEN   2
#define BLUE    3
#define ORANGE  4
#define AQUA    5
#define PINK    6
#define BLACK   7

#define GRAY    8
#define BROWN   9
#define GOLD    10
#define YELLOW    11
#define MAROON 12
#define VIOLET 13
#define ORCHID 14
#define MAXCOLORS 15

/* ***** end ***** */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* ********Heading.h - include file and global varible ******** */
#include <xview/xview.h>
#include <X11/Xlib.h>
#include <X11/Xutil.h>
#include <X11/Xos.h>
#include <X11/X.h>
#include <xview/canvas.h>
#include <xview/cms.h>
#include <xview/xv_xrect.h>
#include <xview/scrollbar.h>
#include <xview/notice.h>
#include <xview/panel.h>
#include <xview/svrimage.h>
#include <xview/icon.h>
#include <xview/font.h>
#include <ctype.h>
#include <stdio.h>
#include <malloc.h>
#include <memory.h>
#include <fcntl.h>
#include <math.h>
#include <string.h>
#include "woman"
#include "color.h"

#define MAXGROW 800
#define MAXGCOL 260
#define FROW 10
#define FCOL 10
#define MAXCOL  MAXGCOL*FCOL
#define MAXROW  MAXGROW*FROW
#define defaultcolw 100

#define S_IREAD 0x0100
#define S_IWRITE 0x0080
#define MAXINPUT 80
#define NO 0
#define YES 1

char buffer[MAXINPUT+1];
float pv;
char *buff;
int yrow,ycol;
char attr;
char *lexbuf;

/* defined data structure */
struct node{
  char attrib;
  union{
    char text[MAXINPUT+1];
    float value;
    struct{
      float fvalue;
      char formula[MAXINPUT+1];
    }f;
  }v;
};
struct group{
  struct node *info[FROW][FCOL];
  struct group *down;
  struct group *right;
  int rid,cid;
};
struct group *ROW[MAXGROW];
struct group *COL[MAXGCOL];

struct width
  {
   int  acol;          /*  column available */
   int  pcolw;         /*  Show pixel of cell */
   struct width *wptr; /* WPTR is link */
  };
struct width *headptr_width;

Frame  frame,frame_save,frame_load,frame_print,frame_format,frame_goto,
       frame_insert_col,frame_insert_row,frame_delete_col,frame_width,
       frame_delete_row,frame_fill_data,frame_copy,frame_move;

Panel  panel,panel_save,panel_load,panel_print,panel_format,panel_goto,
       panel_insert_col,panel_insert_row,panel_delete_row,panel_width,
       panel_delete_col,panel_fill_data,panel_copy,panel_move;
Panel_item create_text_insertrow,create_text,create_textI,create_text2,
       create_text3,create_text4,create_text5,
       create_text_insertcol,create_text_deletecol,create_text_deleterow,
       create_text_format,create_text_print,create_text_save,create_text_goto,
       create_text_load,create_text_print,create_text_save,create_text_goto,
       create_text3,create_text_width1,create_text_width2,
       create_text_move1,create_text_move2,create_text_move3,
       create_text_copy1,create_text_copy2,create_text_copy3;
```

```
GC    gc;
Xv_Font font;
Xv_Window pw;
Menu     menu;
Display  *display;
XID   xid;
Icon     icon;
Scrollbar  vert_scrollbar,horiz_scrollbar;
unsigned long *colors;
XFontStruct  *font_info;
char kbd_msg[128],ptr_msg[128],but_msg[128];
int  m,n,w,h,e,f;
int toprow,leftcol,rightcol,bottomrow;
int first_site;
char buf[20];
char buf_test[1024];
char buf_error[80];
void show_cell();
void build_new();
int  check_built();
void insert_col();
void insert_row();
void delete_col();
void delete_row();
void fill_data();
void alloctext();
void insertcol();
void insertrow();
void deletecol();
void deleterow();
void fill();
void allocvalue();
void allocspace();
void allocformula();
void copy();
void wide();
void move();
void copydata();
void move_data();
void show_data();
void draw_cell();
int  editstring();
void writef_num();
void initvars();
void redrawscreen();
void  drawcol();
void  drawrow();
void clearsheet();
char  *centercolstring();
char  *getcolstr();
struct node *search();
int   getcolwidth();
int   right_col();
void  displaycell();
int  checkrow(),checkcol();
void editcell();
void initptr();
int  parse(),compare();
int  col_to_int();
void     my_notify_proc();
void    initvars();
void     load(),wide(),edit(),recalc(),formula();
void     my_resize_proc(),play();
void    delete(),autocalc(),insert(),print(),values(),find();
void     save(),clear();
void  savesheet();
void loadsheet();
char  cut_string();
int   string_to_int();
int   notify_proc();
int   quit();
void  draw_text();
void  create_read_stipple();
void  expose_01();
void  expose_02();
void  expose_03();
void  expose_04();
int   notify_load(),notify_print(),notify_goto(),
      notify_format(),notify_insert_col(),notify_insert_row(),
      notify_delete_col(),notify_delete_row(),notify_fill_data1(),
      notify_fill_data2(),notify_fill_data3(),notify_fill_data4(),
      notify_fill_data5(),
      notify_copy1(),notify_copy2(),notify_copy3(),notify_move1(),
      notify_move2(),notify_move3(),notify_width1(),notify_width2();
Xv_opaque   notify_save();
void  format(),go_to(),reply(),delete_all(),set_wide();
extern void exit();
char lyparser();
char *parstxt();
char *parsfor();
```

เอกสารนี้เ... ...ไม่ว่ากร... ...ก่อนการนำไปใช้

```
float parsval();
/* *************** end ************* */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*  Util.c -  utility  routine   */
#include "heading.h"
#define  MAXSTR 6
Frame  frame;
int num_col,row;
char buf_error[80];
/* ****************convert string to int************ */
int
parse(str)
char str[MAXSTR];
{
 char tem1,tem2;
 int i;
 char new_str[];
 tem1  = toupper(str[0]);
 num_col=0;
 if (isdigit(str[1]))
   {
   if (str[1]==NULL)
     {
      sprintf(buf_error,"%s","Specied Error!");
      xv_set(frame,FRAME_RIGHT_FOOTER,buf_error,NULL);
      return;
     }
   else if (str[2] == NULL)
        {
         num_col = tem1-'A';
         return num_col;
        }
        else if (str[3] == NULL)
           {
            num_col = tem1-'A';
            return num_col;
           }
           else if (str[4] == NULL)
              {
               num_col = tem1-'A';
               return num_col;
              }
              else if (str[5]== NULL)
               {
               num_col = tem1-'A';
               return num_col;
               }
               else
                {
                num_col = tem1-'A';
                return num_col;
                }
   }
   else if (str[2]==NULL)
     {
      sprintf(buf_error,"%c", "Specified Error!");
      xv_set(frame,FRAME_RIGHT_FOOTER,buf_error,NULL);
      return;
     }
   else if (str[3]== NULL)
        {
         tem2 = toupper(str[1]);
         num_col = compare(tem1,tem2);
         return num_col;
        }
        else if (str[4] == NULL)
           {
            tem2 = toupper(str[1]);
            num_col =compare(tem1,tem2);
            return num_col;
           }
           else if (str[5]== NULL)
              {
               tem2 = toupper(str[1]);
               num_col = compare(tem1,tem2);
               return num_col;
              }
              else
               {
               tem2 = toupper(str[1]);
               num_col =compare(tem1,tem2);
               return num_col;
               }
}
int
compare(st1,st2)
/* char st1[1],st2[1];*/
{
int col;
 if (st1 == 'A')
  {
   col = 26+((st1-'A')+(st2-'A'));
   return col;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    else if (st1 == 'B')
        {
         col = 52+((st1-'A'-1)+(st2-'A'));
         return col;
        }
        else if (st1 == 'C')
         {
          col = 78+((st1-'A'-2)+(st2-'A'));
          return col;
         }
         else if (st1 == 'D')
           {
            col = 104+((st1-'A'-3)+(st2-'A'));
            return col;
           }
           else  if (st1 == 'E')
            {
             col = 130+((st1-'A'-4)+(st2-'A'));
             return col;
            }
            else if (st1 == 'F')
             {
              col = 156+((st1-'A'-5)+(st2-'A'));
              return col;
             }
             else if (st1 == 'G')
              {
              col = 182+((st1-'A'-6)+(st2-'A'));
              return col;
              }
              else if (st1 == 'H')
              {
               col = 208+((st1-'A'-7)+(st2-'A'));
               return col;
              }
              else
               {
                col = 234+((st1-'A'-8)+(st2-'A'));
                return col;
               }
}
int
col_to_int(str)
char str[2];
{
 int col;
 char tem1,tem2;
 if( str[0] == ' '){
   str[0] = str[1];
   str[1] = str[2];
   str[2] = '\0';
 }
 tem1 = toupper(str[0]);
 tem2 = toupper(str[1]);
 if ((isdigit(str[0])) || (isdigit(str[1])) || (str[2] != NULL))
  {
   sprintf(buf_error,"%s","Input Error.");
   xv_set(frame,FRAME_RIGHT_FOOTER,buf_error,NULL);
   return;
  }
  else if (str[1] == NULL)
     {
      col = tem1-'A';
      return col;
     }
     else
      {
       col = compare(tem1,tem2);
       return col;
      }
}
/* char
cut_string(string,c)
char string[];
char c;
{
char new_string[6];
int len;
char *ptr;
strcpy(new_string,string);
ptr = strchr(new_string,c);
if (ptr)
  {
  strcpy((new_string+(ptr-new_string)),(new_string+(ptr-new_string)+1));
  return new_string;
  }
else
  return 0;
} */
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void
setcolwidth(pw,col,wd)
Xv_Window  pw;
int col,wd;
{
struct width *avail,*save;
save = headptr_width;
if (save == NULL)
 {
  save = (struct width *)malloc(sizeof(struct width));
  save->acol = col;
  save->pcolw = wd;
  save->wptr = NULL;
  headptr_width = save;
 XSetForeground(display,gc,colors[WHITE]);
 XFillRectangle(display,xid,gc,0,0,w-30,h-30);
 drawcol();
 drawrow();
 show_data();
  return;
 }
avail = (struct width *)malloc(sizeof(struct width));
avail->acol = col;
avail->pcolw = wd;
avail->wptr = NULL;
if ( (avail->acol) < (save->acol) ) /* the first col to be set */
 {
 headptr_width =avail;
 avail->wptr = save;
 XSetForeground(display,gc,colors[WHITE]);
 XFillRectangle(display,xid,gc,0,0,w-30,h-30);
 drawcol();
 drawrow();
 show_data();
 return;
 }
while ((save->wptr != NULL) && ((save->wptr)->acol<=col))
 save = save->wptr;
if(avail->acol == save->acol){
 save->pcolw = wd;
 free(avail);
 avail = NULL;
 XSetForeground(display,gc,colors[WHITE]);
 XFillRectangle(display,xid,gc,0,0,w-30,h-30);
 drawcol();
 drawrow();
 show_data();
}
else{
 avail->wptr = save->wptr;
 save->wptr = avail;
 XSetForeground(display,gc,colors[WHITE]);
 XFillRectangle(display,xid,gc,0,0,w-30,h-30);
 drawcol();
 drawrow();
 show_data();
 return;
}
}
/* ***** end ***** */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include "heading.h"
void initptr() /* initial pointer for each group row and column */
{
int i,j;
  for(i=0; i<=MAXGROW-1; i++)
    ROW[i] = NULL;
  for(j=0; j<=MAXGCOL-1; j++)
    COL[j] = NULL;
}

check_built(grow,gcol) /* see if there exists the specified group */
int grow,gcol;
{
int built;
struct group *ptr;
  built = NO;
  /* if pointer for that group row or group column is null,
     there does not exsist that group. */
  if( (ROW[grow] == NULL) || (COL[gcol] == NULL) )
    built = NO;
  else{
    ptr = ROW[grow]; /* get the first pointer in that group row */
    /* shift until match that group or
       there are no more group in that group row */
    while( (ptr != NULL) && (built == NO) ){
         if( ptr->cid == gcol)
           built = YES;
         else
           ptr = ptr->right;
    }
  }
  return(built);
}

void build_new(grow,gcol) /* build new group */
 int grow,gcol;
{
struct group *temp;
struct group *ptr;
int i,j;
  ptr = (struct group *)malloc( sizeof(struct group) );
  for(i=0; i<=FROW-1; i++){
    for(j=0; j<=FCOL-1; j++)
      ptr->info[i][j] = NULL;
  }
  /* assign col,row and pointer for new group */
  ptr->cid = gcol;
  ptr->rid = grow;
  temp = ROW[grow];
  if(temp == NULL){ /* Is it the first group in this row ? */
    ROW[grow] = ptr;
    ptr->right = temp;
  }
  else{
    i = 0;
    while( temp != NULL) {
      if ( temp->cid < gcol) {
         temp = temp->right;
         i++;
      }
      else temp = NULL;
    }
    i--;
    temp = ROW[grow];
    if(temp->cid > ptr->cid){ /* Is this the first group in this column ? */
      ROW[grow] = ptr;
      ptr->right = temp;
    }
    else{
      for( j=0; j<i; j++)
         temp = temp->right;
      ptr->right = temp->right;
      temp->right = ptr;
    }
  }
  temp = COL[gcol];
  if(temp == NULL){
    COL[gcol] = ptr;
    ptr->down = temp;
  }
  else{
    i = 0;
    while (temp != NULL) {
      if ( temp->rid < grow) {
         temp = temp->down;
         i++;
      }
      else temp = NULL;
    }
    i--;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        temp = COL[gcol];
        if(temp->rid > ptr->rid){
          COL[gcol] = ptr;
          ptr->down = temp;
        }
        else{
          for( j=0; j<i; j++)
             temp = temp->down;
          ptr->down = temp->down;
          temp->down = ptr;
        }
      }
    }

    void insertcol(col) /* insert an empty column */
    int col;
    {
    int i,j,bufid,lastgrow;
    struct node *buf[FCOL],*buf1[FCOL];
    struct group *temp;
    struct group *ptr,*bufptr;
      /* find out the last row to shift data */
      lastgrow = MAXGROW - 1;
      while( (ROW[lastgrow] == NULL) && (lastgrow > 0) )
        lastgrow--;
      if((lastgrow <= 0) && (ROW[0] == NULL) )
      /* there is still no data in sheet ,insert nothing */
        return;
      ptr = COL[col/FCOL];
      temp = ptr;
      bufptr = ptr;
      /* build new last column to copy last data to */
      while( temp->rid <= lastgrow + 1){
        while( temp->right != NULL)
          temp = temp->right;
        build new(temp->rid,temp->cid + 1);
        temp = bufptr;
        /* copy all last datas in that group row */
        for(i=0;i<=FROW-1;i++)
          buf[i] = temp->info[i][FCOL-1];
        /* clear column to be inserted */
        for( i=0; i<=FROW-1; i++){
          for( j=FCOL-1; j>col%FCOL; j--)
            temp->info[i][j] = temp->info[i][j-1];
          temp->info[i][col%FCOL] = NULL;
        }
        /* begin shifting data */
        temp = bufptr->right;
        while( temp != NULL ){
          bufid = bufptr->cid;
          if(temp->cid != bufid+1)
            build new(ptr->rid,bufid+1);
          for(i=0;i<=FROW-1;i++)
            buf1[i] = temp->info[i][FCOL-1];
          for( i=0; i<=FROW-1; i++){
            for( j=FCOL-1; j>=0; j--)
              temp->info[i][j] = temp->info[i][j-1];
            temp->info[i][0] = buf[i];
            buf[i] = buf1[i];
          }
          bufptr = temp;
          temp = bufptr->right;
        }
        bufptr = ptr->down;
        if( bufptr == NULL)
          build new( ptr->rid + 1,ptr->cid );
        else if( bufptr->rid != ptr->rid + 1 )
          build new( ptr->rid + 1,ptr->cid );
        ptr = ptr->down;
        bufptr = ptr;
        temp = ptr;
      }
     show_data();
    }

    void insertrow(row)
    int row;
    {
    int i,j,bufid,lastgcol;
    struct node *buf[FROW],*buf1[FROW];
    struct group *temp;
    struct group *ptr,*bufptr;
      /* find out the last column to shift data */
      lastgcol = MAXGCOL - 1;
      while( (COL[lastgcol] == NULL) && (lastgcol > 0) )
        lastgcol--;
      if( (lastgcol <= 0) && (COL[0] == NULL) )
      /* there is still no data in sheet, insert nothing */
        return;
      ptr = ROW[row/FROW];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    temp = ptr;
    bufptr = ptr;
    /* build new last row to copy data to */
    while( temp->cid <= lastgcol + 1){
      while( temp->down != NULL)
        temp = temp->down;
      build new(temp->rid + 1,temp->cid );
      temp = bufptr;
      /* copy all last datas in that group column */
      for(j=0;j<=FCOL-1;j++)
        buf[j] = temp->info[FROW-1][j];
      /* clear row to be inserted */
      for( j=0; j<=FCOL-1; j++){
        for( i=FROW-1; i>row%FROW; i--)
          temp->info[i][j] = temp->info[i-1][j];
        temp->info[row%FROW][j] = NULL;
      }
      /* begin shifting data */
      temp = bufptr->down;
      while( temp != NULL ){
        bufid = bufptr->rid;
        if(temp->rid != bufid+1)
          build new(bufid+1,ptr->cid);
        for(j=0;j<=FCOL-1;j++)
          buf1[j] = temp->info[FROW-1][j];
        for( j=0; j<=FCOL-1; j++){
          for( i=FROW-1; i>=0; i--)
            temp->info[i][j] = temp->info[i-1][j];
          temp->info[0][j] = buf[j];
          buf[j] = buf1[j];
        }
        bufptr = temp;
        temp = bufptr->down;
      }
      bufptr = ptr->right;
      if( bufptr == NULL)
        build new( ptr->rid ,ptr->cid + 1);
      else if( bufptr->cid != ptr->cid + 1 )
        build new( ptr->rid ,ptr->cid + 1 );
      ptr = ptr->right;
      bufptr = ptr;
      temp = ptr;
    }
   show_data();
}

void deletecol(col) /* delete a specified column */
int col;
{
int i,j,lastgrow,mkrow;
struct group *temp;
struct group *ptr;
  j = col%FCOL;
  /* find out lastrow to shift data */
  lastgrow = MAXGROW - 1;
  while( (ROW[lastgrow] == NULL) && (lastgrow >0) )
    lastgrow--;
  if( (lastgrow <= 0) && (ROW[0] == NULL) )
  /* there is still no data in sheet, nothing to be deleted */
    return;
  lastgrow++;
  temp = COL[col/FCOL];
  while( temp != NULL ){     /* delete all datas in that column */
    for( i=0; i<=FROW-1; i++){
      if(temp->info[i][j] != NULL){
        free(temp->info[i][j]);
        temp->info[i][j] = NULL;
      }
    }
    temp = temp->down;
  } /* end delete all datas */
  for( mkrow=0; mkrow<=lastgrow; mkrow++){  /* start shifting */
    ptr = ROW[mkrow];
    /* find out if this group needs shifting */
    i = 0;
    j = 0;
    while( (ptr != NULL) && (i != 3) ){
      if(ptr->cid < (col/FCOL)){
        j++;
        ptr = ptr->right;
      }
      else
        i = 3;
    }
    j--;
    if( ptr != NULL){  /* this group needs shifting */
      if( ptr->cid == (col/FCOL) ){
        for(i=0; i<=FROW-1; i++){
          for(j=(col%FCOL); j<FCOL-1 ; j++)
            ptr->info[i][j] = ptr->info[i][j+1];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ptr->info[i][FCOL-1] = NULL;
          }
          temp = ptr;
          ptr = ptr->right;
        }
        else{
          temp = ROW[mkrow];
          while( j >  0){
            temp = temp->right;
            j--;
          }
        }
        while( ptr != NULL ){
          if( (check_built(ptr->rid,ptr->cid - 1)) == NO ){
            build_new(ptr->rid,ptr->cid - 1);
            if(ptr != temp)
              temp = temp->right;
            else
              temp = ROW[mkrow];
          }
          for(i=0; i<=FROW-1; i++)
            temp->info[i][FCOL-1] = ptr->info[i][0];
          for(i=0; i<=FROW-1; i++){
            for(j=0; j<FCOL-1; j++)
              ptr->info[i][j] = ptr->info[i][j+1];
            ptr->info[i][FCOL-1] = NULL;
          }
          temp = ptr;
          ptr = ptr->right;
        }
      } /* end one group */
    } /* end shifting */
    show_data();
  } /* end deletecol */

  void deleterow(row) /* delete a specified row */
  int row;
  {
  int i,j,lastgcol,mkcol;
  struct group *temp;
  struct group *ptr;
    i = row%FROW;
    /* find out lastrow to shift data */
    lastgcol = MAXGCOL - 1;
    while( (COL[lastgcol] == NULL) && (lastgcol > 0) )
      lastgcol--;
    if( (lastgcol <= 0) && (COL[0] == NULL) )
    /* there is still no data in sheet, nothing to be deleted */
      return;
    lastgcol++;
    temp = ROW[row/FROW];
    while( temp != NULL ){     /* delete all datas in that row */
      for( j=0; j<=FCOL-1; j++){
        if(temp->info[i][j] != NULL){
          free(temp->info[i][j]);
          temp->info[i][j] = NULL;
        }
      }
      temp = temp->right;
    } /* end delete all datas */
    for( mkcol=0; mkcol<=lastgcol; mkcol++){  /* start shifting */
      ptr = COL[mkcol];
      j = 0;
      i = 0;
      /* find out if this group needs shifting */
      while( (ptr != NULL) && (j != 3) ){
        if(ptr->rid < (row/FROW)){
          i++;
          ptr = ptr->down;
        }
        else
          j = 3;
      }
      i--;
      if( ptr != NULL){  /* this group needs shifting */
        if( ptr->rid == (row/FROW) ){
          for(j=0; j<=FCOL-1; j++){
            for(i=(row%FROW); i<FROW-1 ; i++)
              ptr->info[i][j] = ptr->info[i+1][j];
            ptr->info[FROW-1][j] = NULL;
          }
          temp = ptr;
          ptr = ptr->down;
        }
        else {
          temp = COL[mkcol];
          while( i >  0){
            temp = temp->down;
            i--;
          }
```

เอกสารนี้เป็นเอกสาร... เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        while( ptr != NULL ){
          if( (check_built(ptr->rid - 1,ptr->cid )) == NO ){
            build_new(ptr->rid - 1,ptr->cid );
            if( ptr != temp)
              temp = temp->down;
            else
              temp = COL[mkcol];
          }
          for(j=0; j<=FCOL-1; j++)
            temp->info[FROW-1][j] = ptr->info[0][j];
          for(j=0; j<=FCOL-1; j++){
            for(i=0; i<FROW-1; i++)
              ptr->info[i][j] = ptr->info[i+1][j];
            ptr->info[FROW-1][j] = NULL;
          }
          temp = ptr;
          ptr = ptr->down;
        }
      } /* end one group */
    } /* end shifting */
    show_data();
  } /* end deleterow */

  void allocspace(row,col) /* allocate an empty cell */
  int row,col;
  {
  struct group *ptr;
    if( (check_built(row/FROW,col/FCOL) ) != NO ){
      ptr = ROW[row/FROW];
      while( ptr->cid < (col/FCOL) )
        ptr = ptr->right;
      if(ptr->cid == (col/FCOL) ){
        free(ptr->info[row%FROW][col%FCOL]);
        ptr->info[row%FROW][col%FCOL] = NULL;
      }
    }
  }

  void alloctext(row,col,data) /* allocate a text cell */
  int row,col;
  char data[MAXINPUT+1];
  {
  struct group *ptr;
  struct node *bufinfo;
    if( (check_built(row/FROW,col/FCOL) ) == NO )
      build_new(row/FROW,col/FCOL);
    /*find where to store data */
    ptr = ROW[row/FROW];
    while( ptr->cid < (col/FCOL) )
      ptr = ptr->right;
    bufinfo = ptr->info[row%FROW][col%FCOL];
    if( bufinfo == NULL){
      bufinfo = (struct node *)malloc( sizeof(struct node) );
      ptr->info[row%FROW][col%FCOL] = bufinfo;
    }
    strcpy ( bufinfo->v.text , data );
    bufinfo->attrib = 'T';
  }

  void allocvalue(row,col,data) /* allocate a value cell */
  int row,col;
  float data;
  {
  struct group *ptr;
  struct node *bufinfo;
    if( (check_built(row/FROW,col/FCOL) ) == NO )
      build_new(row/FROW,col/FCOL);
    /*find where to store data */
    ptr = ROW[row/FROW];
    while( ptr->cid < (col/FCOL) )
      ptr = ptr->right;
    bufinfo = ptr->info[row%FROW][col%FCOL];
    if( bufinfo == NULL){
      bufinfo = (struct node *)malloc( sizeof(struct node) );
      ptr->info[row%FROW][col%FCOL] = bufinfo;
    }
    bufinfo->v.value = data;
    bufinfo->attrib = 'V';
  }

  void allocformula(row,col,dataf,datav) /* allocate a formula cell */
  int row,col;
  float datav;
  char dataf[MAXINPUT+1];
  {
  struct group *ptr;
  struct node *bufinfo;
    if( (check_built(row/FROW,col/FCOL) ) == NO )
      build_new(row/FROW,col/FCOL);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  /*find where to store data */
  ptr = ROW[row/FROW];
  while( ptr->cid < (col/FCOL) )
    ptr = ptr->right;
  bufinfo = ptr->info[row%FROW][col%FCOL];
  if( bufinfo == NULL){
    bufinfo = (struct node *)malloc( sizeof(struct node) );
    ptr->info[row%FROW][col%FCOL] = bufinfo;
  }
  strcpy(bufinfo->v.f.formula,dataf);
  bufinfo->v.f.fvalue = datav;
  bufinfo->attrib = 'F';
}


void fill(col1,row1,col2,row2,start,last,step) /* fill a range of value */
 int col1,row1,col2,row2,start,last,step;
{
int row,col;
   /* check and swap */
  if(row1 > row2){
    row = row1;
    row1 = row2;
    row2 = row;
  }
  if(col1 > col2){
    col = col1;
    col1 = col2;
    col2 =col;
  }
  if( ((start > last) && (step > 0)) || ((start < last) && (step < 0)) ){
      xv_set(frame,FRAME_RIGHT_FOOTER," ERROR !!!!!!!!",NULL);
      exit(1);
  }
  /* fill data */
  for( col=col1; col<=col2; col++){
    for( row=row1; row<=row2; row++){
      if( !(( (start < last) && (step <= 0) )|| ( (start >last) && (step >= 0) ) )
        allocvalue(row,col,start);
        start = start + step;
      }
    }
  }/*end filling data */
  show_data();
} /* end fill */

void copydata(fcol,frow,lcol,lrow,bcol,brow)
int bcol,brow,fcol,frow,lcol,lrow;
{
int i,j,k,l;
struct group *from,*fptr;
struct node *frominfo;
  /* check and swap */
  if(frow > lrow){
    i = frow;
    frow = lrow;
    lrow = i;
  }
  if(fcol > lcol){
    j = lcol;
    fcol = lcol;
    lcol = j;
  }
    if( ((frow <= brow) && (brow <= lrow)) && ((fcol <= bcol) && (bcol <= lcol)) )
    {
      xv_set(frame,FRAME_RIGHT_FOOTER,"Can't copy to itself.",NULL);
    exit(1);
    }
  for(j=fcol/FCOL; j<=lcol/FCOL; j++){
    for(i=frow/FROW; i<=frow/FROW; i++){
      if( check_built(i,j) == NO )
        build_new(i,j);
    }
  }
  /* where to begin */
  from = ROW[frow/FROW];
  while( from->cid != fcol/FCOL )
    from = from->right;
  /* copy data */
  l = bcol;
  for(j=fcol;j<=lcol;j++){
    fptr = from;
    k = brow;
    for(i=frow;i<=lrow;i++){
      frominfo = fptr->info[i%FROW][j%FCOL];
      if( frominfo != NULL ){
        switch(frominfo->attrib){
          case 'T': alloctext(k,l,frominfo->v.text);
                    break;
          case 'V': allocvalue(k,l,frominfo->v.value);
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    break;
            case 'F': allocformula(k,l,frominfo->v.f.formula,frominfo->v.f.fvalue);
                    break;
            default : break;
          }/* end switch */
        }/* end if */
        else allocspace(k,l);
        if( i%FROW == FROW - 1 )    /* check limit FROW */
          fptr = fptr->down;
        k = k + 1;
      }/* end for i */
      if( j%FCOL == FCOL - 1 )  /* check limit FCOL */
        from = from->right;
      l = l + 1;
    }/* end for j */
    show_data();
}/* end copydata */

void move_data(fcol,frow,lcol,lrow,bcol,brow)
int bcol,brow,fcol,frow,lcol,lrow;
{
int i,j,k,l;
struct group *from,*fptr;
struct node *frominfo;
  /* check and swap */
  if(frow > lrow){
    i = frow;
    frow = lrow;
    lrow = i;
  }
  if(fcol > lcol){
    j = lcol;
    fcol = lcol;
    lcol = j;
  }
  if( ((frow <= brow) && (brow <= lrow)) && ((fcol <= bcol) && (bcol <= lcol))) {
   xv_set(frame,FRAME_RIGHT_FOOTER,"Can't move to itself.",NULL);
   return;
  }
  for(j=fcol/FCOL; j<=lcol/FCOL; j++){
    for(i=frow/FROW; i<=frow/FROW; i++){
      if( check_built(i,j) == NO )
        build_new(i,j);
    }
  }
  /* where to begin */
  from = ROW[frow/FROW];
  while( from->cid != fcol/FCOL )
    from = from->right;
  /* move data */
  l = bcol;
  for(j=fcol;j<=lcol;j++){
    fptr = from;
    k = brow;
    for(i=frow;i<=lrow;i++){
      frominfo = fptr->info[i%FROW][j%FCOL];
      if( frominfo != NULL ){
        switch(frominfo->attrib){
          case 'T': alloctext(k,l,frominfo->v.text);
                    free(fptr->info[i%FROW][j%FCOL]);
                    fptr->info[i%FROW][j%FCOL] = NULL;
                    break;
          case 'V': allocvalue(k,l,frominfo->v.value);
                    free(fptr->info[i%FROW][j%FCOL]);
                    fptr->info[i%FROW][j%FCOL] = NULL;
                    break;
          case 'F': allocformula(k,l,frominfo->v.f.formula,frominfo->v.f.fvalue);
                    free(fptr->info[i%FROW][j%FCOL]);
                    fptr->info[i%FROW][j%FCOL] = NULL;
                    break;
          default : break;
        }/* end switch */
      }/* end if */
      else allocspace(k,l);
      if( i%FROW == FROW - 1 )    /* check limit FROW */
        fptr = fptr->down;
      k = k + 1;
    }/* end for i */
    if( j%FCOL == FCOL - 1 )  /* check limit FCOL */
      from = from->right;
    l = l + 1;
  }/* end for j */
  show_data();
}/* end move_data */


void savesheet(filename)
char filename[MAXINPUT+1];
{
char *name = "spreadsheet";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลง อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  int  i,j,size,next,file;
  int  gcol,grow,lastgcol,lastgrow,col,row;
  int sci = sizeof(int);
  int scf = sizeof(float);
  int count = 0;
  /* int EOF = ??? */
  struct group *ptr;
  struct node *bufinfo;
    if(!access(filename,0)){
      if( ( file  = open(filename,O_WRONLY || O_CREAT ,S_IREAD || S_IWRITE)) == -1){
       sprintf(buf_error,"%s","Could not open file.");
       xv_set(frame,FRAME_RIGHT_FOOTER,buf_error,NULL);
        return;
      }
    }
    else{
      if( (file  = creat(filename,0666)) == -1){
        sprintf(buf_error,"%s","Could not open file.");
        xv_set(frame,FRAME_RIGHT_FOOTER,buf_error,NULL);
        return;
      }
    }
    /* write header to declare this is a spreadsheet file */
    write(file,name,strlen(name) + 1 );
    lastgcol = MAXGCOL - 1;
    while( (COL[lastgcol] == NULL) && (lastgcol > 0) )
      lastgcol--;
    if( (lastgcol <= 0) && (COL[0] == NULL) ){
    /* there is nothing to save,exit */
      close(file);
      return;
    }
    lastgrow = MAXGROW - 1;
    while( (ROW[lastgrow] == NULL) && (lastgrow > 0) )
      lastgrow--;
    if( (lastgrow <= 0) && (ROW[0] == NULL) ){
    /* there is nothing to save, exit */
      close(file);
      return;
    }
    /* reserve a place to write size of sheet */
    size = MAXGCOL * FCOL ;
    write(file,(char *)&size,sci);
    /* begin saving cell by cell */
    for(grow=0;grow<=lastgrow;grow++){
      ptr = ROW[grow];
      while(ptr != NULL ){
        for(i=0;i<FROW;i++){
          for(j=0;j<FCOL;j++){
            bufinfo = ptr->info[i][j];
            next = 0;
            if(bufinfo != NULL){
              switch(bufinfo->attrib){
                case 'T':size = strlen(bufinfo->v.text) + (4 - strlen(bufinfo->v.text
                         break;
                case 'V':size = scf + 4;
                         break;
                case 'F':size = strlen(bufinfo->v.f.formula) + (4 - strlen(bufinfo->v
%4) + scf +4;
                         break;
                default : next = 1;
                         break;
              } /*switch*/
            }
            else
              next = 1;
            if(next == 0 ){
              col = (ptr->cid)*FCOL + j;
              row = (ptr->rid)*FROW + i;
              write(file,(char *)&col,sci);
              write(file,(char *)&row,sci);
              write(file,(char *)&size,sci);
              write(file,(char *)bufinfo,size);
              bufinfo = NULL;
              count = count + size;
            } /*endif*/
          }/* end for j */
        }/* end for i */
        ptr = ptr->right;
      }/* end while */
    }/* end for grow */
    size = strlen(name) +1;
    /* seek for the reserved place to write size of sheet */
    /*write(file,(char *)&EOF,sci); */
    lseek(file,size,0);
    write(file,(char *)&count,sci);
    close(file);
  }/* end savesheet */

  void loadsheet(filename)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  char filename[MAXINPUT+1];
  {
  int size,file,lastgcol,lastgrow,col,row;
  char check[81];
  char *name = "spreadsheet";
  int sci = sizeof(int);
  int count;
  struct node bufinfo;
    if(access(filename,0)){
    sprintf(buf_error,"Could not find file %s .",filename);
    xv_set(frame,FRAME_RIGHT_FOOTER,buf_error,NULL);
      return;
    }
    if((file = open(filename,O_RDWR )) == -1){
        sprintf(buf_error,"%s","Could not open file.");
    xv_set(frame,FRAME_RIGHT_FOOTER,buf_error,NULL);
      return;
    }
    read(file,check,strlen(name) + 1);
    if(strcmp(check,name) != 0){ /*Is this a spreadsheet file ? */
      sprintf(buf_error,"%s","NO FILE !!!!!!!!");
      xv_set(frame,FRAME_RIGHT_FOOTER,buf_error,NULL);
      close(file);
      return;
    }
    clearsheet();
    /* read size of sheet */
    read(file,(char *)&count,sci);
    /* begin loading cell by cell */
    do{
      if( (read(file,(char *)&col,sci) <= 0) || (count <= 0) )
        break;
      read(file,(char *)&row,sci);
      read(file,(char *)&size,sci);
      read(file,(char *)&bufinfo,size);
      switch(bufinfo.attrib){
        case 'T':alloctext(row,col,bufinfo.v.text);
                 count = count - size;
                 break;
        case 'V':allocvalue(row,col,bufinfo.v.value);
                 count = count - size;
                 break;
        case 'F':allocformula(row,col,bufinfo.v.f.formula,bufinfo.v.f.fvalue);
                 count = count - size;
                 break;
        default :break;
      }/* end switch */
    }
    while(1);
    close(file);
    show_data();
  }/* end loadsheet */

  void clearsheet()
  {
    int k,i,j;
    struct group *bufcol;
    for(k=0;k<MAXGCOL;k++){
      while ( COL[k] != NULL ){
        bufcol = COL[k];
        for( i=0; i<FROW; i++){
          for( j=0; j<FCOL; j++){
            if ( bufcol->info[i][j] != NULL){
               free(bufcol->info[i][j]);
               bufcol->info[i][j] = NULL;
            }
          }/* end for j */
        } /* end for i */
        COL[k] = COL[k]->down;
        free(bufcol);
      }/*end while COL */
    } /* end for k(col) */
    /* inititate ROW */
    for( k=0; k<MAXGROW; k++)
      ROW[k] = NULL;
  } /* end clearsheet */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

(ข)

คำสั่งฐานิกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
spell filename          #check correct spelling word

nohup command&          #do command even logout

nice command            #decrease priority

^old^new      # Editing command

more [b]      # backward by screen

head [-,+n]     # show the beginning of file [at line n]
tail [-,+n]     # show the beginning of file [at line n]

File descriptor
        0       stdin
        1       stdout
        2       stderr

Job Control
        CTRL-Z   suspend curent process
        bg [%n]  resume stopped job no.n in background
        fg [%n]  resume stopped job no.n in foreground
        jobs     list job
        kill [%n] terminate job
        stop[%n]  stop execution of job

echo mesg,$var          # show mesg or value of shell variable
                          the CTRL-G character in echo will ring bell

finger [name]           #show status of usr name

file [filename]         #show status of file

![n,-n]             # redoing history command n
!command            # redoing the last command beginning with command
!!                  # redo previous command
!!:p                # display previous command
!:n                 # nth argument
!*                  # all argument
!$                  # last argument
!^                  # first argument

~                   # abreviating login directory

find [path] -name [filename] -print

VI EDITOR
        vi + filename   #to last line
        a               #append
        i               #insert
        A               #append at end of current line
        I               #insert at beginning of current line
        O               #insert line above
        o               #insert line below

        :w              #save
        :q              #quit
        :q!             #quit not save

        l       #move right
        h       #move left
        j       #move down
        k       #move up
        ^       #move to beginning of line
        $       #move to end of line
        H       #move to top of screen
        M       #move to middle of screen
        L       #move to bottom of screen

        w       #move to next word
        e       #move to end of word
        b       #move back to beginning of word
        )       #move to start of next sentence
        (       #move back to start of sentence
        }       #move to start of next paragraph
        {       #move back to start of paragraph
        ]]      #move to start of next section
        [[      #move back to start of section

        [n]CTRL-F       #move forward one screen
        [n]CTRL-D       #move forward one half screen
        [n]CTRL-B       #move back one screen
        [n]CTRL-U       #move back one half screen
        [n]G            #move to end of file or line n
        :n              #move to line n
        :n1,n2          #from line n1 to n2
        :n1,n2s/old/new/g
                        #replace old to new (g is global)
        :n1,n2s/.old/new        #replace ???old to new
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
s       #change character
cw      #change word
cc      #change line
c$      #change to end of line

[n]x    #delete current character
[n]dd   #delete line
[n]dw   #delete word
[n]d$   #delete to end of line

[n]yy   #yacc line
[n]yw   #yacc word
[n]y$   #yacc to end of line

u       #undo the most recent change
U       #undo all of change in line
:e!     #undo all change [edit again!]

p       #put content in buffer to the right of cursor
P       #put content in buffer to the left of cursor

/string     #search string
//,n        #search next occurrence of string
?string     #search backward
??,N        #search backward next occurrence of string
"           #stop searching

"[a-z]d,y,c      #delete yacc or change then place
                  the content to[a-z]buffer
"[n][a-z]p,P     #place the content in [a-z]buffer to text

:r !command      #inserting output from shell command
                  to the current line
:[n]j            #combind line
:sh              #go to shell,press CTRL-D to return
:! cmd           #run command from vi

editing multiple file:
        vi file1 file2 file3    #edit in sequence
        :n                      #begin edit next file
        #the benefit of editing several files in one editing
         session is that you can move text between file by
         named buffer.

grep string filename     #find string in file
grep "many strings" filename
grep string *            #using grep to locate file
grep -v string filename     #find line that do not match string

script [filename]      #copy everything displayed on terminal to file

SHELL PROGRAMMING

        sh filename      #to execute shell script
        (cmd;cmd;...)    #to run sequence cmd(command) in subshell
        {cmd;...;cmd;}   #to run sequence cmd(command) in current shell
        #                #comment
        shell positional parameter
                cmd     arg1    arg2 ...   arg9
                $0       $1       $2   ...    $9
                $*   all argument

        \        #has continue line
        $#       #contains the number of positional parameter
        $?       #return the number of success or fail command
        $!       #contain the process ID of the last background process

        ${variable:-word}        # if variable is undefiled,use word
            but it does not set or change the value of the variable.

        ${variable:=word}        # if variable is undefiled,use word
                                   and set variable to word.
        ${variable:?message}     # if variable is undefiled,print mesg
                                   on stderr.
        variable=.....           # assign value to variable and to
                                   invoke variable use $variable
        echo ...                 # write to stdout
        read variable            # read one line from stdin then
                                   assign to variable
        here document...
                <<word           or      <<-delimiting word eg. !
                .                        .
                .                        .
                word                     del. word
        # to provide multi-line input to command within shell script

    BATCH COMMAND
        batch cmd        #to run cmd but don't care when executed

    AT COMMAND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        at [-f filename] time
        at time << (filename,here document)
        # read command for later execution
        # time may be:
                hour alone
                hour:minute
                can specify am, pm, zulu(Greenwich time)
                day name
                month name day number,year number
                noon,midnight,now,today,tomorrow,next,minute(s),
                  hour(s),week(s),month(s),year(s)
                increment a specified time eg. now + 1 week

        at -l   #list all scheduled at job
        at -r job ID    #remove at job

set 'cmd'       #set take the output of cmd and assign the shell
                 positional parameter to each of output word

trap command interrupt-nomber #command to be executed when interrupt
                               signal are recieved
        command - 'sequence command'
        interrupt-number
                0       shell exit
                1       hangup
                2       interrupt,DEL
                3       quit
                9       kill(cannot be trap)
                15      terminate

exit 0                  #program terminated normally
exit 1                  #program terminated abnormally
exit 2                  #usage or command line error by user

 IF COMMAND
        if command
        then commands
        fi
                #the command following the if is executed.
        If it completes successfully,the command following the then
        are executed.

 TEST COMMAND
        test expression         #test evaluates an expression;
                                 return 0 if expression is true
                                return non-0 if expression is not true
        more information see man test

IF..THEN..ELSE
        if command
        then commands
        else commands
        fi
                #if the if command is true,the then command is executed
                 if the if command is false,the else command is executed

IF..THEN..ELIF
        if command
        then commands
        elif command
        then commands
        else commands
        fi

CASE COMMAND
        case string
        in
          pattern-list)
                        command line
                        command line
                        ...
                        ;;

          pattern-list)
                        command line
                        command line
                        ...
                        ;;
        esac
THE && and || OPERATOR
        command1 && command2
        #if cmd1 return true then cmd2 executed
         and && return true if both command return true

        command1 || command2
        #if cmd1 return false then cmd2 executed
         and || return false if both command return false

FOR LOOP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        for i
           in list
        do
           commands
        done
                #If omit 'in list' line command will be executed
                 once for each positional parameter

   WHILE
        while commands1
        do

                commands2
        done

   UNTIL
        until commands1
        do
                commands2
        done

   BREAK
        break   #exits from immediately loop
        break n #exits from immediately n loop

   CONTINUE
        continue        #go back to top of smallest loop
        continue n      #go back to top of smallest nth loop

   expr argument        #expr take the arguments as expression,
                         evaluates them and print on stdout

   sh -x filename       #tell shell to run script and print each
                         command and its argument as it is executed

NETWORKING**********
        /etc/host       #contain nachine name and address on network
        /etc/hasts.equiv  #contain machine trust name

        ping machine-name       #to check the remote machine is up
        ruptime   #to obtain information of all machine on the net work
        finger usr@mach-name    #obtain information of usr on mach-name
        rwho            #print information of each user on network
        rusers

        rlogin machine          #same user name
        rlogin machine -l user  #different user name

        CTRL-D,~.               #disconnect rlogin
        ~~.                     #disconnect intermidiate rlogin
        ~ CTRL-Z                #suspend rlogin
        ~~ CTRL-Z               #suspend intermidiate rlogin
        %                       #to resume rlogin

        REMOTE FILE COPY
                rcp machine:file directory      #from another machine
                rcp file machine:directory      #to another machine
                rcp -r                          #copy directory

        rsh machine command     #execute command on another machine

        COPY FROM ANOTHER MACHINE-NOT UNIX SYSTEM
                ftp
                ftp>open machine-name
                name:username
                passwd:
                ftp>get file directory  #copy from
                ftp>send directory file #copy to
                ftp>quit

SYSTEM BOOT
        L1-A            #to boot system
        >b              #boot
        >b -s           #boot to single user(superuser)

HALTING SYSTEM
        shutdown -h now #halt in root logname
        su              #become superuser then
        /usr/etc/halt   #halt in other logname

HALT HANG-SYSTEM
        L1-A
        >n
        ok sync         #to prevent losing data

CHANGE ROOT PASSWORD AT BOOT
        L1-A
        >b -s
        passwd
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
REINDIRECTING I/O
        >       #re output
        <       #re input
        >!      #re existing output(replace old data)
        >>,>>!  #append to output
        >&      #re error file

PIPE
        |

MAIL AND MESSAGE
        mail user[@machine]     #send mail
                ~v      #use vi editor
                CTRL-D  #end of text
                CTRL-C  #interrupt CTRL-C again to kill letter
                s       #save
                d       #delete
                q       #quit
                r       #reply
                ~r file #reply by insert file

        mail            #read mail
        mail -f file    #read mail that saved in file

        talk user[@machine]     #use CTRL-C to disconnect
        write user              #use CTRL-D to disconnect
        wall                    #broadcast

KILL COMMAND
        kill process-ID
        kill -9 process-ID      #sure kill
        kill 0                  #kill all process

TAPE and DISK
        fdformat        #format diskette
        /dev/rst0       #tape name
        /dev/rfd0       #disk name

        -tvf            #to read
        -cvf            #to overwrite
        -rvf            #to append
        -xvf            #to extract
        -xvpf           #to extract by check directory

        tar [option] [device name] filename
                        #filename: if use [.] mean all file in current directory
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

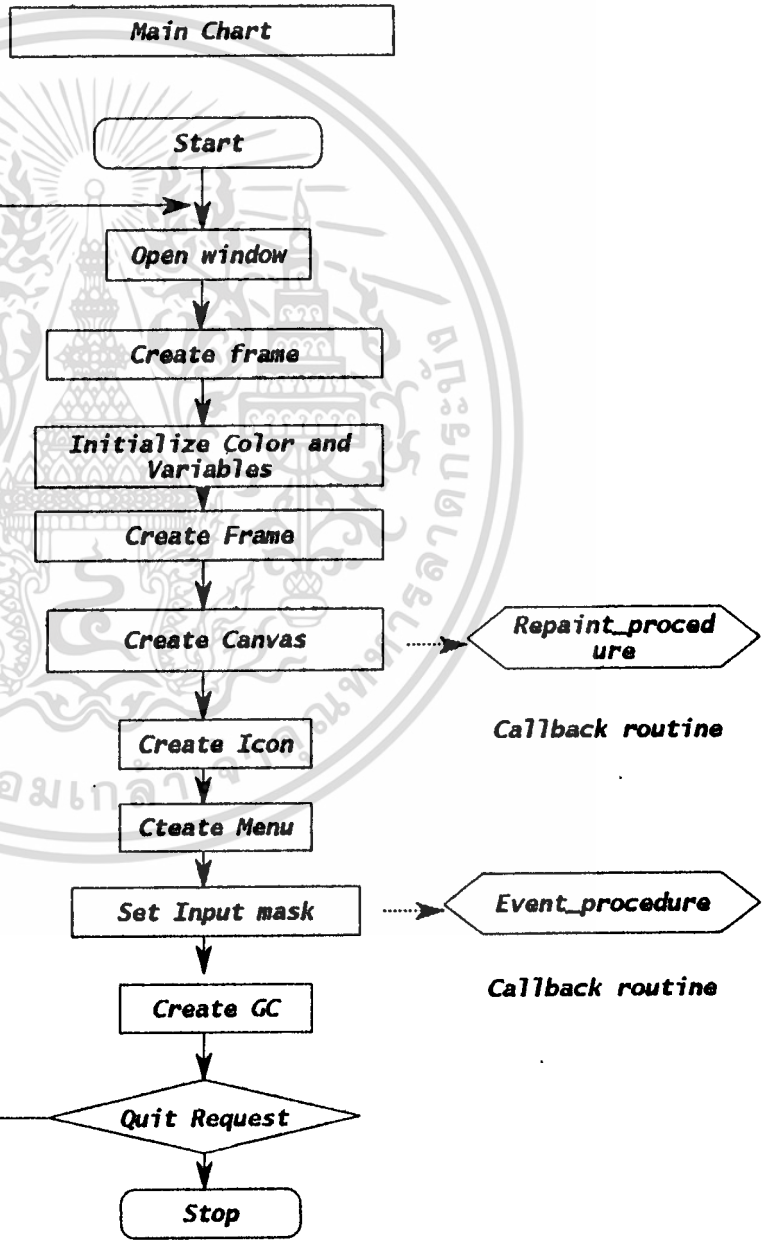
(ค)

โฟว์ชาร์ต

TopChart

Sunday, March 15, 1992

12:16 PM



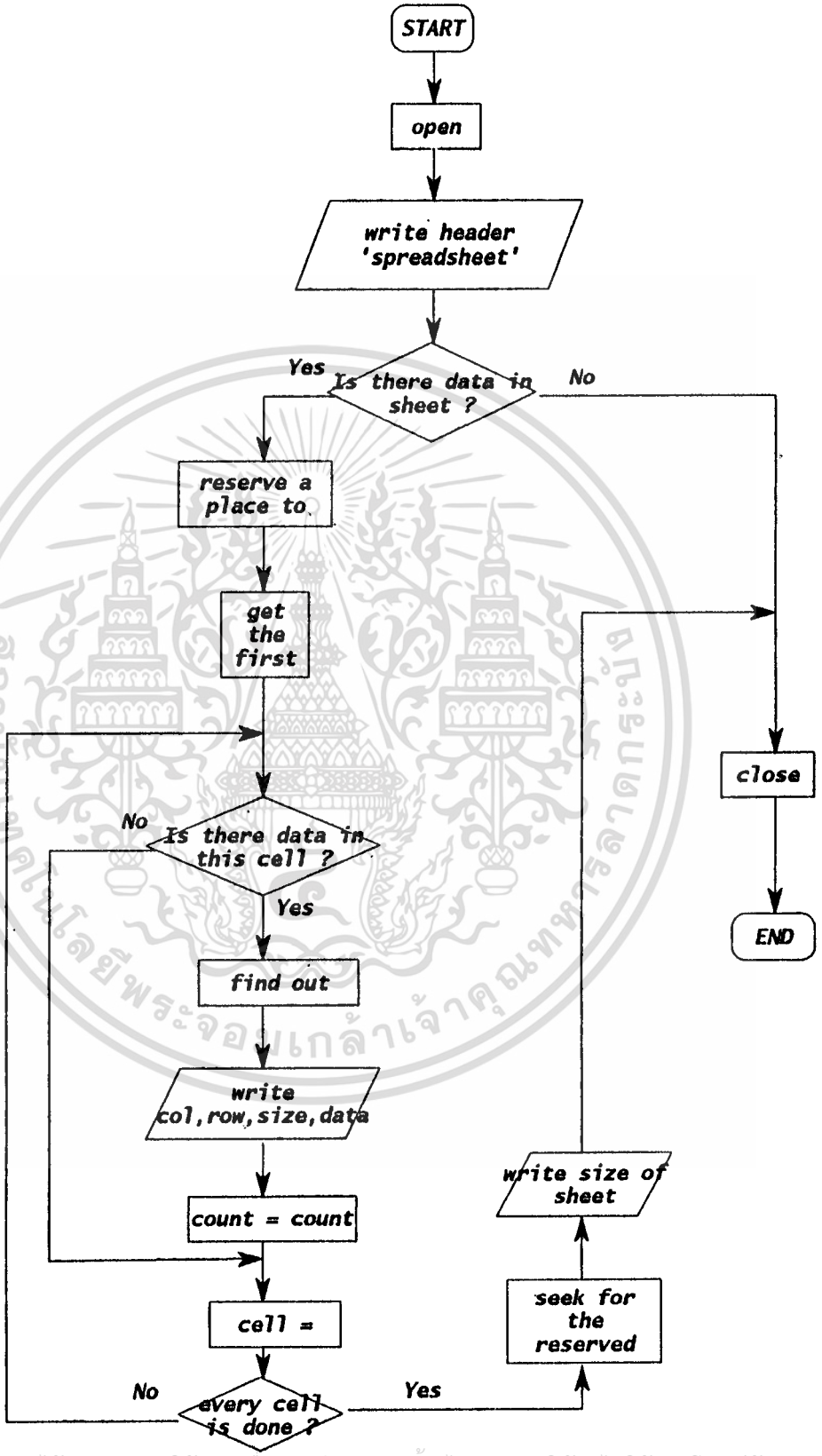
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TopChart

Saturday, March 14, 1992

2:48 PM

Save sheeet

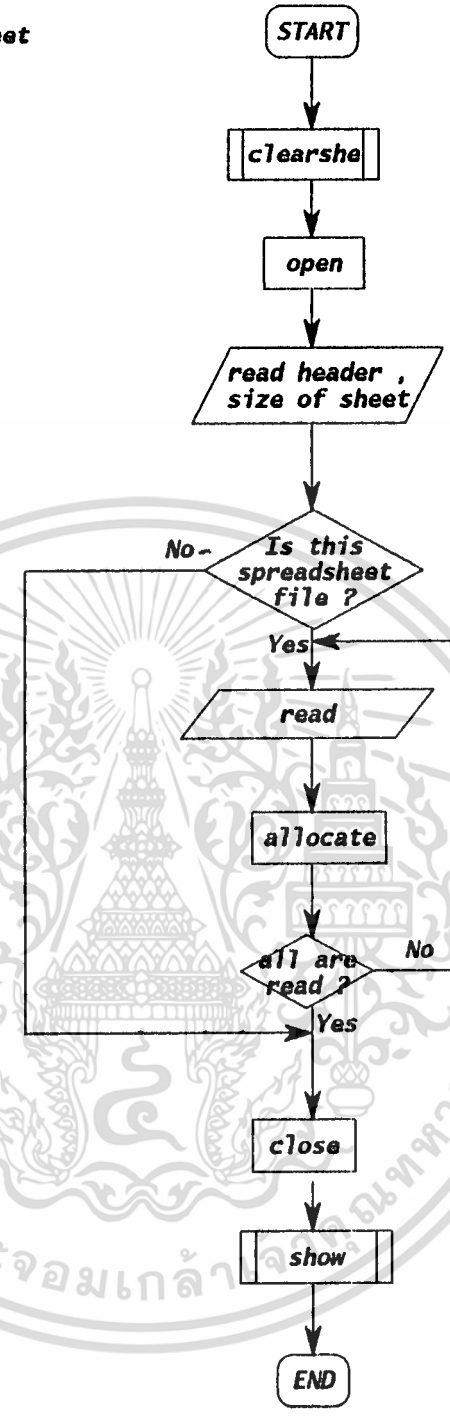


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TopChart

Saturday, March 14, 1992 Load sheet

3:57 PM

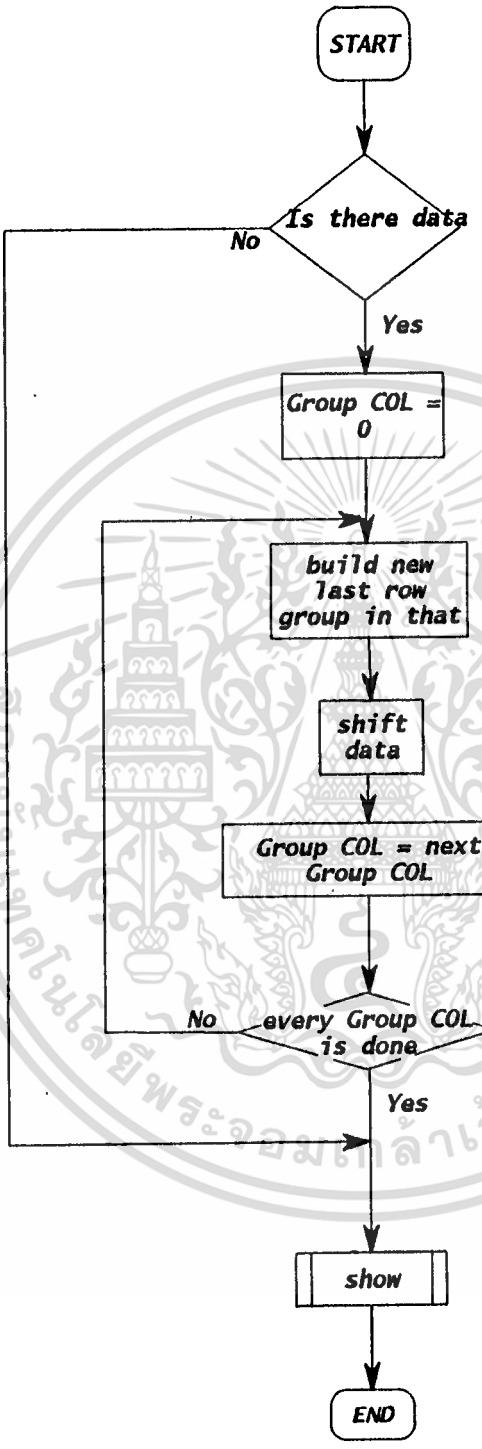


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TopChart

Saturday, March 14, 1992

1:17 PM

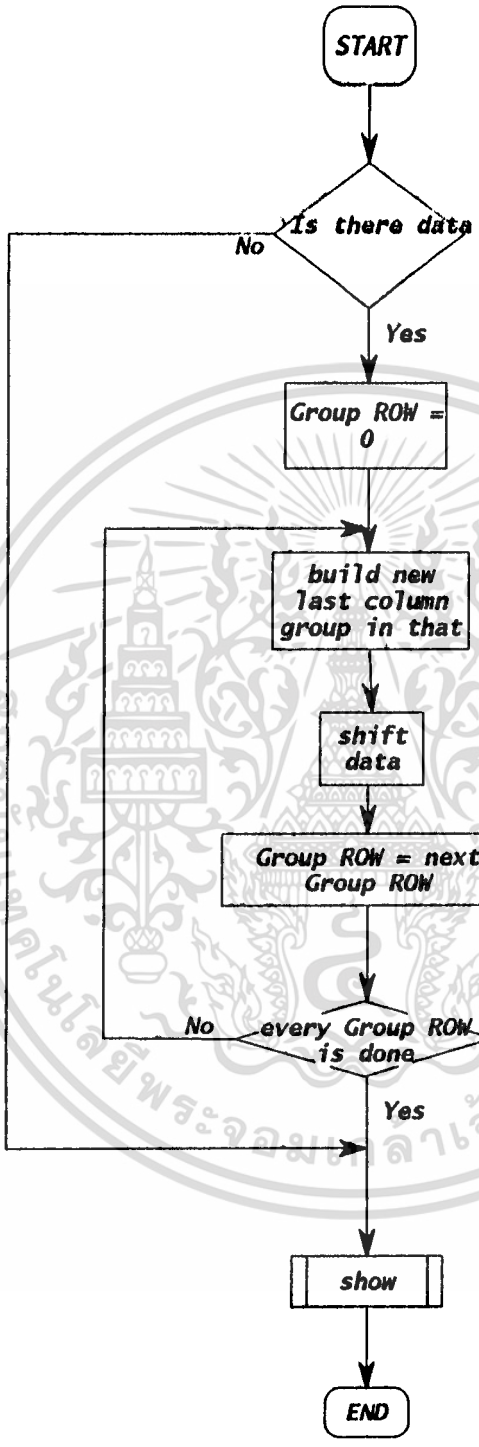


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TopChart

Saturday, March 14, 1992

1:17 PM

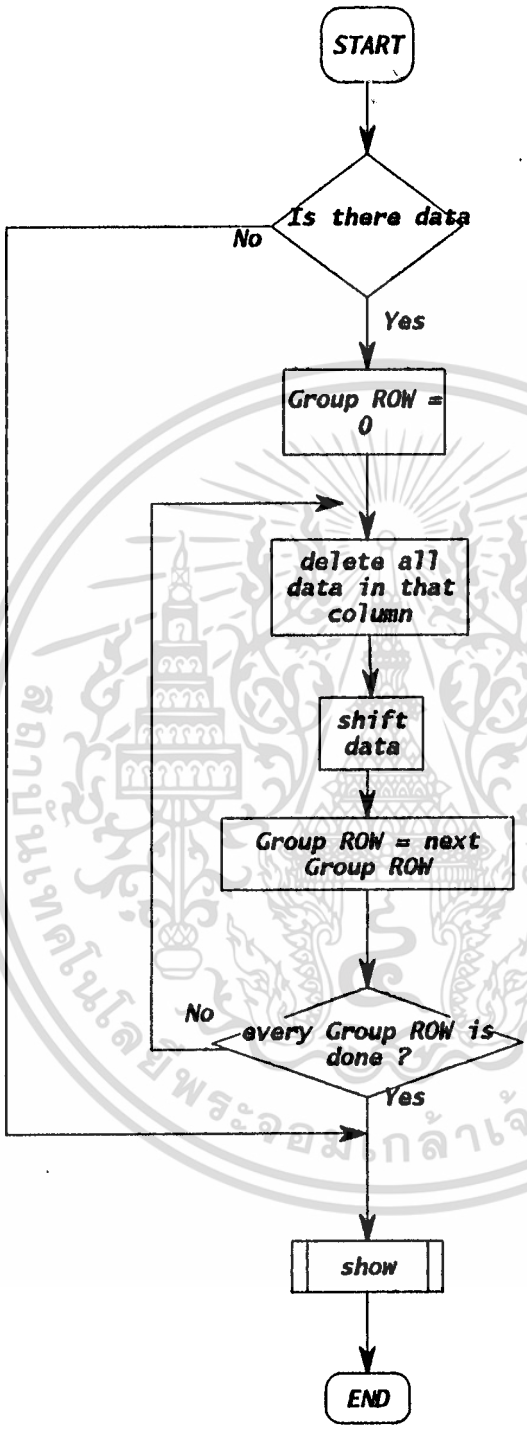


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TopChart

Saturday, March 14, 1992

1:17 PM

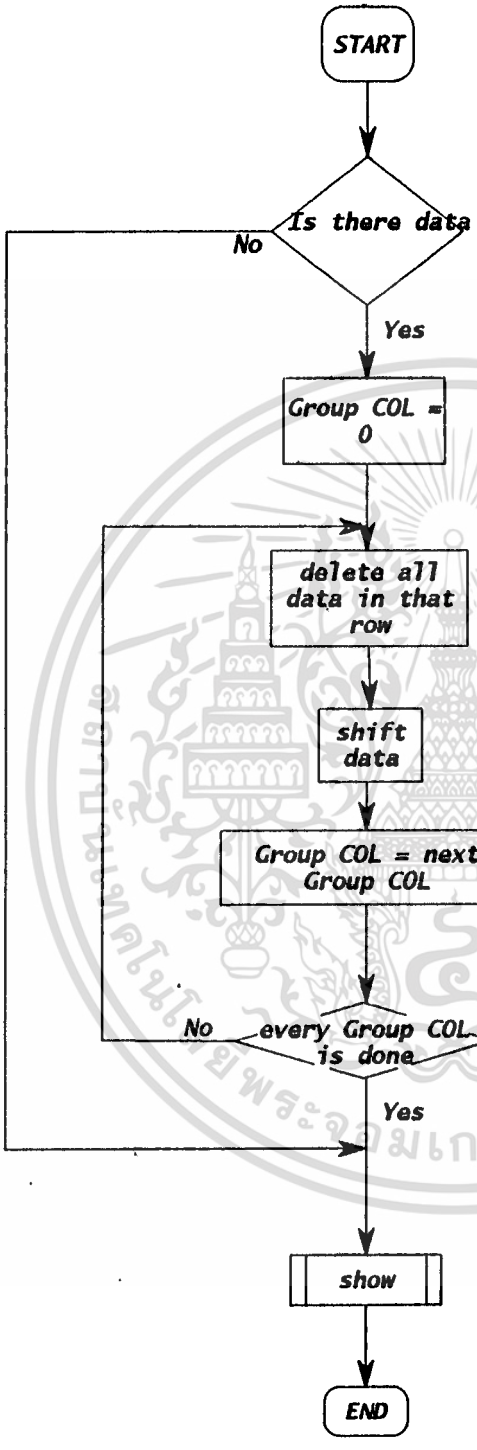


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TopChart

Saturday, March 14, 1992

1:17 PM

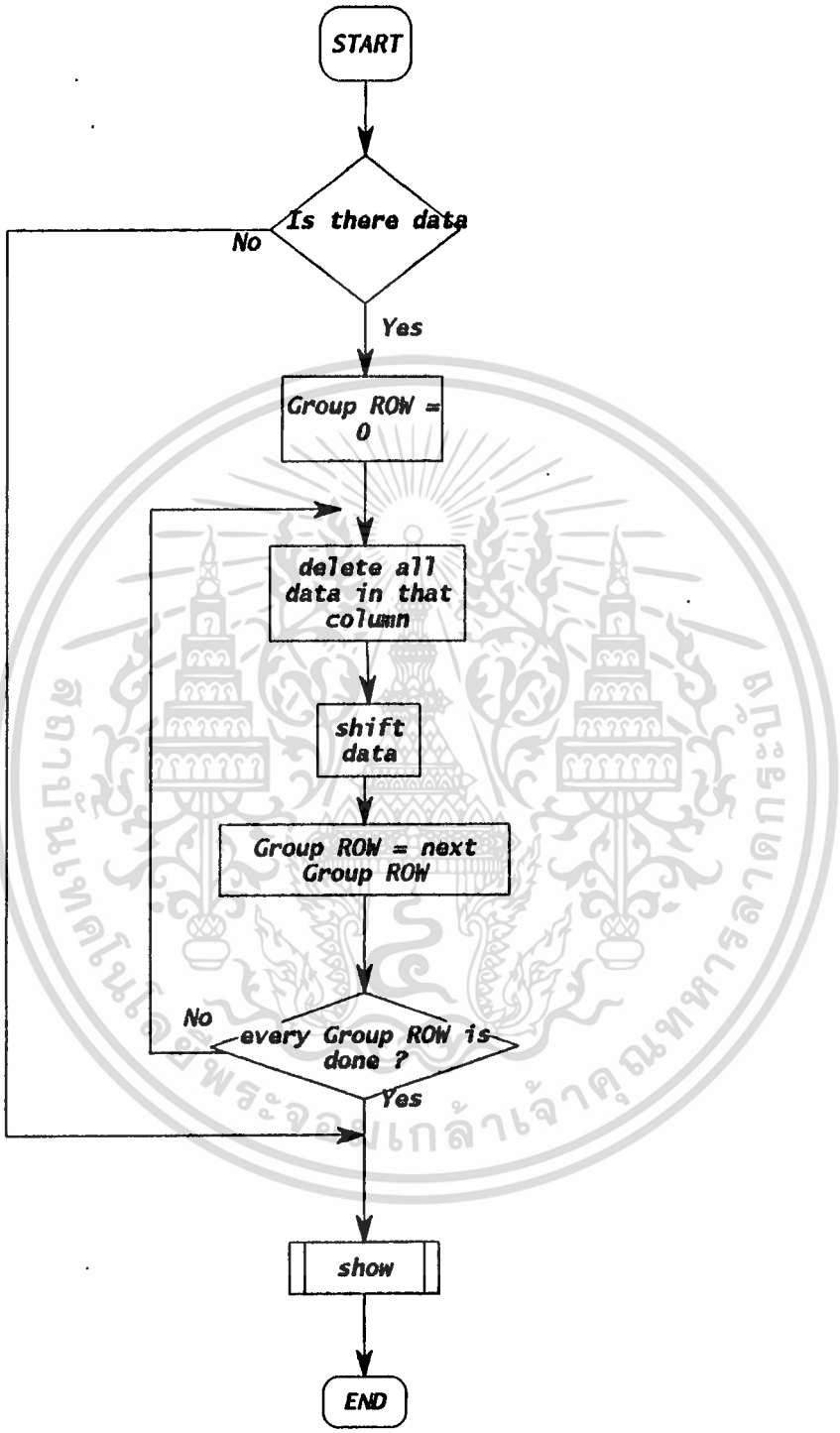


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TopChart

Saturday, March 14, 1992

1:17 PM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม parser.l

```
%{
#include"y.tab.h"
extern float pv;
extern char *buff;
extern int yrow,ycol;
extern char attr;

%}
NUM     {DIG}+("."{DIG}+)?([E|e][+|-]?{DIG}+)?
ALP      [A-Za-z]
DIG      [0-9]
ALN      [0-9A-Za-z]
WHI    [ \t\n]
%%
{ALP}{ALP}?{DIG}{DIG}?{DIG}?    {
                                  if(isupper(yytext[0]))
                                     ycol = yytext[0]-'A';
                                  else
                                  ycol = yytext[0]-'a';
                                  yrow =atof(yytext+1)-1;
                                  return CELL;
                                 }
{NUM}                {
                                  pv = atof(yytext);
                                  return NUMBER;
                     }
"$"                  { return FT; }
\+                   { return PLUS; }
\-                   { return MINUS; }
\*                   { return MULT; }
\/                   { return DIV; }
\(                   { return OB; }
\)                   { return CB; }
\$                   { return DOLSIG; }
\%                   { return PERC; }
\@                   { return FS; }
{ALP}+{DIG}*         { buff = yytext; return TXT1;   }
\'{ALN}*              { buff = yytext + 1; return TXT2; }
\^{ALN}*              { buff = yytext + 1; return TXT3;   }
\"{ALN}*             { buff = yytext + 1; return TXT4;   }
{WHI}                { return WHI; }
%%
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม parser.y

```
%{
#define YYSTYPE float
extern float pv;
extern char attr;
%}

%token  FT CELL NUMBER PLUS MULT WHI MINUS DIV ALP DOLSIG
%token  OB CB PERC FS TXT1 TXT2 TXT3 TXT4
%left '+' '-'
%left '*' '/'
%left  UMINUS
%start  expr
%%
expr       : line WHI     { return pv; }
           ;
line       : function     { attr = 'F'; }
           | text         { attr = 'T'; }
           | arg          { attr = 'V'; }
           ;
function   : /*empty*/
           | function PLUS function    { $$ = $1 + $3; }
           | function MULT function    { $$ = $1 * $3; }
           | function MINUS function   { $$ = $1 - $3; }
           | function DIV function     { $$ = $1 / $3; }
           | OB function CB           { $$ = $2; }
           | FS function
           | PLUS function
           | MINUS function
           | CELL
           ;
text       : /* -- empty -- */
           | TXT1
           | TXT2
           | TXT3
           | TXT4
           ;
arg        : /* empty */
           | arg PLUS arg     { $$ = pv = $1 + $3; }
           | arg MINUS arg    { $$ = pv = $1 - $3; }
           | arg MULT arg     { $$ = pv = $1 * $3; }
           | arg DIV arg      { $$ = pv = $1 / $3; }
           | DOLSIG arg       { $$ = pv = $2; }
           | arg PERC         { $$ = pv = $1 / 100; }
           | OB arg CB    { $$ = pv = $2; }
           | PLUS arg         { $$ = pv = $2; }
           | MINUS arg        { $$ = pv = -$2; }
           | NUMBER           { $$ = pv;  }
           ;
%%
yyerror()
     {
          return(1);
     }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม y.tab.h

```
# define FT 257
# define CELL 258
# define NUMBER 259
# define PLUS 260
# define MULT 261
# define WHI 262
# define MINUS 263
# define DIV 264
# define ALP 265
# define DOLSIG 266
# define OB 267
# define CB 268
# define PERC 269
# define FS 270
# define UMINUS 271
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม parser.c

```
#include"heading.h"
#include"pyacc.h"
#include"plex.h"
char *lexbuf;
char lyparser(parsbuf)
  char parsbuf[MAXINPUT+1];
  {

     yyin=fopen("lexbuf","w");
     fprintf(yyin,"%s",parsbuf);
    fclose(yyin);
    yyin=fopen("lexbuf","r");
     yyparse();

    fclose(yyin);
     return attr;

  }
/* ----- function to return value ------*/
float parsval(parsbuf)
 char parsbuf[MAXINPUT+1];
    {
     yyin=fopen("lexbuf","w");
     fprintf(yyin,"%s",parsbuf);
    fclose(yyin);
    yyin=fopen("lexbuf","r");
     yyparse();

    fclose(yyin);
      return pv;
    }

/* ------ function to return formula ----*/
char *parsfor(parsbuf)
 char parsbuf[MAXINPUT+1];
    {
     yyin=fopen("lexbuf","w");
     fprintf(yyin,"%s",parsbuf);
    fclose(yyin);
    yyin=fopen("lexbuf","r");
     yyparse();

       fclose(yyin);
       return buff;
    }
/* ------ function to return text ------*/
char *parstxt(parsbuf)
 char parsbuf[MAXINPUT+1];
    {
     yyin=fopen("lexbuf","w");
     fprintf(yyin,"%s",parsbuf);
    fclose(yyin);
    yyin=fopen("lexbuf","r");
     yyparse();
       yyparse();
       fclose(yyin);
       return buff;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม y.tab.c

```
# line 2 "pum.y"
#define YYSTYPE float
extern float pv;
extern char attr;
# define FT 257
# define CELL 258
# define NUMBER 259
# define PLUS 260
# define MULT 261
# define WHI 262
# define MINUS 263
# define DIV 264
# define ALP 265
# define DOLSIG 266
# define OB 267
# define CB 268
# define PERC 269
# define FS 270
# define TXT1 271
# define TXT2 272
# define TXT3 273
# define TXT4 274
# define UMINUS 275
#define yyclearin yychar = -1
#define yyerrok yyerrflag = 0
extern int yychar;
extern short yyerrflag;
#ifndef YYMAXDEPTH
#define YYMAXDEPTH 150
#endif
#ifndef YYSTYPE
#define YYSTYPE int
#endif
YYSTYPE yylval, yyval;
# define YYERRCODE 256

# line 49 "pum.y"

yyerror()
     {
          return(1);
     }



short yyexca[] ={
-1, 0,
        269, 20,
        -2, 5,
-1, 1,
        0, -1,
        -2, 0,
-1, 6,
        269, 20,
        -2, 5,
-1, 8,
        269, 20,
        -2, 5,
-1, 9,
        269, 20,
        -2, 5,
        };
# define YYNPROD 31
# define YYLAST 101
short yyact[]={

  10,  16,   8,  18,  19,   9,  20,  21,  15,   6,
  17,  49,   7,  11,  12,  13,  14,  10,  16,   8,
  18,  19,   9,  20,  21,  15,   6,   4,  10,   7,
  31,   2,   1,  32,   0,  22,  24,  30,  23,  25,
   7,   0,   0,  50,  26,  22,  24,   0,  23,  25,
   0,  16,  39,   0,  26,  40,  28,   5,  15,  38,
   0,   0,   0,   0,   0,  34,  36,  27,   3,   0,
   0,   0,  37,   0,   0,  29,  33,  35,   0,  45,
  46,  47,  48,   0,   0,   0,  41,  42,  43,  44,
   0,   0,   0,   0,   0,   0,  34,  36,   0,  33,
  35 };
short yypact[]={

-258,-1000,-252,-240,-1000,-215,-241,-230,-241,-241,
-1000,-1000,-1000,-1000,-1000,-208,-1000,-1000,-230,-230,
-230,-230,-208,-208,-208,-208,-1000,-257,-225,-240,
-230,-230,-230,-240,-215,-240,-215,-215,-208,-208,
-208,-240,-240,-240,-240,-215,-215,-215,-215,-1000,
-1000 };
short yypgo[]={

   0,  32,  31,  67,  27,  56 };
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ต้องอ้างอิงเอกสารทุกครั้งที่มีการนำไปใช้

```
short yyr1[]={

    0,    1,    2,    2,    2,    3,    3,    3,    3,    3,
    3,    3,    3,    3,    3,    4,    4,    4,    4,    4,
    5,    5,    5,    5,    5,    5,    5,    5,    5,    5,
    5 };
short yyr2[]={

    0,    2,    1,    1,    1,    0,    3,    3,    3,    3,
    3,    2,    2,    2,    1,    0,    1,    1,    1,    1,
    0,    3,    3,    3,    3,    2,    2,    3,    2,    2,
    1 };
short yychk[]={

-1000,   -1,   -2,   -3,   -4,   -5,  267,  270,  260,  263,
 258,  271,  272,  273,  274,  266,  259,  262,  260,  261,
 263,  264,  260,  263,  261,  264,  269,   -3,   -5,   -3,
 267,  260,  263,   -3,   -5,   -3,   -5,   -5,  267,  260,
 263,   -3,   -3,   -3,   -3,   -5,   -5,   -5,   -5,  268,
 268 };
short yydef[]={

   -2,   -2,    0,    2,    3,    4,   -2,    5,   -2,   -2,
   14,   16,   17,   18,   19,   20,   30,    1,    5,    5,
    5,    5,   20,   20,   20,   20,   26,    0,    0,   11,
    5,    5,    5,   12,   28,   13,   29,   25,   20,   20,
   20,    6,    7,    8,    9,   21,   22,   23,   24,   10,
   27 };
#ifndef lint
static  char yaccpar_sccsid[] = "@(#)yaccpar 1.6 88/02/08 SMI"; /* from UCB 4.1 83/02/11 */
#endif

#
# define YYFLAG -1000
# define YYERROR goto yyerrlab
# define YYACCEPT return(0)
# define YYABORT return(1)

/*      parser for yacc output  */

#ifdef YYDEBUG
int yydebug = 0; /* 1 for debugging */
#endif
YYSTYPE yyv[YYMAXDEPTH]; /* where the values are stored */
int yychar = -1; /* current input token number */
int yynerrs = 0;  /* number of errors */
short yyerrflag = 0;  /* error recovery flag */

yyparse() {

        short yys[YYMAXDEPTH];
        short yyj, yym;
        register YYSTYPE *yypvt;
        register short yystate, *yyps, yyn;
        register YYSTYPE *yypv;
        register short *yyxi;

        yystate = 0;
        yychar = -1;
        yynerrs = 0;
        yyerrflag = 0;
        yyps= &yys[-1];
        yypv= &yyv[-1];

 yystack:    /* put a state and value onto the stack */

#ifdef YYDEBUG
        if( yydebug  ) printf( "state %d, char 0%o\n", yystate, yychar );
#endif
                if( ++yyps>= &yys[YYMAXDEPTH] ) { yyerror( "yacc stack overflow" ); return(
 }
                *yyps = yystate;
                ++yypv;
                *yypv = yyval;

 yynewstate:

        yyn = yypact[yystate];

        if( yyn<= YYFLAG ) goto yydefault; /* simple state */

        if( yychar<0 ) if( (yychar=yylex())<0 ) yychar=0;
        if( (yyn += yychar)<0 || yyn >= YYLAST ) goto yydefault;

        if( yychk[ yyn=yyact[ yyn ] ] == yychar ){ /* valid shift */
                yychar = -1;
                yyval = yylval;
                yystate = yyn;
                if( yyerrflag > 0 ) --yyerrflag;
                goto yystack;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }

yydefault:
        /* default state action */

        if( (yyn=yydef[yystate]) == -2 ) {
                if( yychar<0 ) if( (yychar=yylex())<0 ) yychar = 0;
                /* look through exception table */

                for( yyxi=yyexca; (*yyxi!= (-1)) || (yyxi[1]!=yystate) ; yyxi += 2 ) ; /* VOID
*/

                while( *(yyxi+=2) >= 0 ){
                        if( *yyxi == yychar ) break;
                        }
                if( (yyn = yyxi[1]) < 0 ) return(0);   /* accept */
                }

        if( yyn == 0 ){ /* error */
                /* error ... attempt to resume parsing */

                switch( yyerrflag ){

                case 0:   /* brand new error */

                        yyerror( "syntax error" );
                yyerrlab:
                        ++yynerrs;

                case 1:
                case 2: /* incompletely recovered error ... try again */

                        yyerrflag = 3;

                        /* find a state where "error" is a legal shift action */

                        while ( yyps >= yys ) {
                           yyn = yypact[*yyps] + YYERRCODE;
                           if( yyn>= 0 && yyn < YYLAST && yychk[yyact[yyn]] == YYERRCODE ){
                              yystate = yyact[yyn];  /* simulate a shift of "error" */
                              goto yystack;
                              }
                           yyn = yypact[*yyps];

                           /* the current yyps has no shift onn "error", pop stack */

#ifdef YYDEBUG
                           if( yydebug ) printf( "error recovery pops state %d, uncovers %d\n"
, *yyps, yyps[-1] );
#endif

                           --yyps;
                           --yypv;
                           }

                        /* there is no state on the stack with an error shift ... abort */

        yyabort:
                        return(1);


                case 3:  /* no shift yet; clobber input char */

#ifdef YYDEBUG
                        if( yydebug ) printf( "error recovery discards char %d\n", yychar );
#endif

                        if( yychar == 0 ) goto yyabort; /* don't discard EOF, quit */
                        yychar = -1;
                        goto yynewstate;   /* try again in the same state */

                        }

                }

        /* reduction by production yyn */

#ifdef YYDEBUG
                if( yydebug ) printf("reduce %d\n",yyn);
#endif
                yyps -= yyr2[yyn];
                yypvt = yypv;
                yypv -= yyr2[yyn];
                yyval = yypv[1];
                yym=yyn;
                        /* consult goto table to find next state */
                yyn = yyr1[yyn];
                yyj = yypgo[yyn] + *yyps + 1;
                if( yyj>=YYLAST || yychk[ yystate = yyact[yyj] ] != -yyn ) yystate = yyact[yyp
go[yyn]];
                switch(yym){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case 1:
# line 14 "pum.y"
{ return pv; } break;
case 2:
# line 16 "pum.y"
{ attr = 'F';  } break;
case 3:
# line 17 "pum.y"
{ attr = 'T';  } break;
case 4:
# line 18 "pum.y"
{ attr = 'V';  } break;
case 6:
# line 21 "pum.y"
{ yyval = yypvt[-2] + yypvt[-0]; } break;
case 7:
# line 22 "pum.y"
{ yyval = yypvt[-2] * yypvt[-0]; } break;
case 8:
# line 23 "pum.y"
{ yyval = yypvt[-2] - yypvt[-0]; } break;
case 9:
# line 24 "pum.y"
{ yyval = yypvt[-2] / yypvt[-0]; } break;
case 10:
# line 25 "pum.y"
{ yyval = yypvt[-1]; } break;
case 21:
# line 38 "pum.y"
{ yyval = pv = yypvt[-2] + yypvt[-0]; } break;
case 22:
# line 39 "pum.y"
{ yyval = pv = yypvt[-2] - yypvt[-0]; } break;
case 23:
# line 40 "pum.y"
{ yyval = pv = yypvt[-2] * yypvt[-0]; } break;
case 24:
# line 41 "pum.y"
{ yyval = pv = yypvt[-2] / yypvt[-0]; } break;
case 25:
# line 42 "pum.y"
{ yyval = pv = yypvt[-0]; } break;
case 26:
# line 43 "pum.y"
{ yyval = pv = yypvt[-1] / 100; } break;
case 27:
# line 44 "pum.y"
{ yyval = pv = yypvt[-1]; } break;
case 28:
# line 45 "pum.y"
{ yyval = pv = yypvt[-0]; } break;
case 29:
# line 46 "pum.y"
{ yyval = pv = -yypvt[-0]; } break;
case 30:
# line 47 "pum.y"
{ yyval = pv;  } break;
                }
                goto yystack;  /* stack new state and value */

        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ผลลัพธ์ lex.yy.c

```
# include "stdio.h"
# define U(x) x
# define NLSTATE yyprevious=YYNEWLINE
# define BEGIN yybgin = yysvec + 1 +
# define INITIAL 0
# define YYLERR yysvec
# define YYSTATE (yyestate-yysvec-1)
# define YYOPTIM 1
# define YYLMAX BUFSIZ
# define output(c) putc(c,yyout)
# define input() (((yytchar=yysptr>yysbuf?U(*--yysptr):getc(yyin))==10?(yylineno++,yytchar):yy
tchar)==EOF?0:yytchar)
# define unput(c) {yytchar= (c);if(yytchar=='\n')yylineno--;*yysptr++=yytchar;}
# define yymore() (yymorfg=1)
# define ECHO fprintf(yyout, "%s",yytext)
# define REJECT { nstr = yyreject(); goto yyfussy;}
int yyleng; extern char yytext[];
int yymorfg;
extern char *yysptr, yysbuf[];
int yytchar;
FILE *yyin, *yyout = {stdout};
extern int yylineno;
struct yysvf {
        struct yywork *yystoff;
        struct yysvf *yyother;
        int *yystops;};
struct yysvf *yyestate;
extern struct yysvf yysvec[], *yybgin;
#include"y.tab.h"
extern float pv;
extern char *buff;
extern int yrow,ycol;
extern char attr;

# define YYNEWLINE 10
yylex(){
int nstr; extern int yyprevious;
while((nstr = yylook()) >= 0)
yyfussy: switch(nstr){
case 0:
if(yywrap()) return(0); break;
case 1:
  {
                    if(isupper(yytext[0]))
                      ycol = yytext[0]-'A';
                    else
                    ycol = yytext[0]-'a';
                    yrow =atof(yytext+1)-1;
                    return CELL;
                  }

break;
case 2:
        {
                  pv = atof(yytext);
                  return NUMBER;

            }
break;
case 3:
         {  return FT; }
break;
case 4:
           { return PLUS; }
break;
case 5:
           { return MINUS; }
break;
case 6:
           { return MULT; }
break;
case 7:
           { return DIV; }
break;
case 8:
           { return OB; }
break;
case 9:
           { return CB; }
break;
case 10:
           { return DOLSIG; }
break;
case 11:
           { return PERC; }
break;
case 12:
           { return FS; }
break;
case 13:
   { buff = yytext; return TXT1;  }
break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case 14:
        { buff = yytext + 1; return TXT2; }
break;
case 15:
        { buff = yytext + 1; return TXT3;  }
break;
case 16:
        { buff = yytext + 1; return TXT4;  }
break;
case 17:
          {  return WHI; }
break;
case -1:
break;
default:
fprintf(yyout,"bad switch yylook %d",nstr);
} return(0); }
/* end of yylex */

int yyvstop[] = {
0,

17,
0,

16,
0,

10,
0,

11,
0,

14,
0,

8,
0,

9,
0,

6,
0,

4,
0,

5,
0,

7,
0,

2,
0,

12,
0,

13,
0,

15,
0,

3,
0,

1,
13,
0,

13,
0,

2,
0,

2,
0,

1,
13,
0,

13,
0,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1,
13,
0,

13,
0,
0};
# define YYTYPE char
struct yywork { YYTYPE verify, advance; } yycrank[] = {
0,0,	0,0,	1,0,	1,0,
1,0,	1,0,	1,0,	1,0,
1,0,	1,0,	1,3,	0,0,
1,0,	1,0,	1,0,	1,0,
1,0,	1,0,	1,0,	1,0,
1,0,	1,0,	1,0,	1,0,
1,0,	1,0,	1,0,	1,0,
1,0,	1,0,	1,0,	1,0,
1,0,	0,0,	1,0,	1,4,
1,0,	1,5,	1,6,	1,0,
1,7,	1,8,	1,9,	1,10,
1,11,	1,0,	1,12,	1,13,
1,14,	1,15,	0,0,	0,0,
0,0,	0,0,	0,0,	0,0,
13,19,	0,0,	0,0,	1,0,
1,0,	1,0,	1,0,	1,0,
1,0,	1,16,	1,17,	0,0,
0,0,	0,0,	1,17,	20,24,
20,24,	20,24,	20,24,	20,24,
20,24,	20,24,	20,24,	20,24,
20,24,	22,27,	22,27,	22,27,
22,27,	22,27,	22,27,	22,27,
22,27,	22,27,	22,27,	0,0,
1,0,	1,0,	1,0,	1,18,
1,0,	1,0,	0,0,	0,0,
0,0,	0,0,	0,0,	0,0,
0,0,	24,21,	25,26,	25,26,
25,26,	25,26,	25,26,	25,26,
25,26,	25,26,	25,26,	25,26,
0,0,	0,0,	0,0,	0,0,
0,0,	0,0,	0,0,	0,0,
1,0,	1,0,	1,0,	1,0,
1,0,	2,0,	2,0,	2,0,
2,0,	2,0,	2,0,	2,0,
2,0,	24,21,	0,0,	2,0,
2,0,	2,0,	2,0,	2,0,
2,0,	2,0,	2,0,	2,0,
2,0,	2,0,	2,0,	2,0,
2,0,	2,0,	2,0,	2,0,
2,0,	2,0,	2,0,	2,0,
24,21,	2,0,	2,4,	2,0,
2,5,	2,6,	2,0,	2,7,
2,8,	2,9,	2,10,	0,0,
2,0,	2,12,	2,13,	2,14,
4,4,	4,4,	4,4,	4,4,
4,4,	4,4,	4,4,	4,4,
4,4,	4,4,	2,0,	2,0,
2,0,	2,0,	2,0,	2,0,
2,16,	4,4,	4,4,	4,4,
4,4,	4,4,	4,4,	4,4,
4,4,	4,4,	4,4,	4,4,
4,4,	4,4,	4,4,	4,4,
4,4,	4,4,	4,4,	4,4,
4,4,	4,4,	4,4,	4,4,
4,4,	4,4,	4,4,	2,0,
2,0,	2,0,	2,18,	2,0,
2,0,	4,4,	4,4,	4,4,
4,4,	4,4,	4,4,	4,4,
4,4,	4,4,	4,4,	4,4,
4,4,	4,4,	4,4,	4,4,
4,4,	4,4,	4,4,	4,4,
4,4,	4,4,	4,4,	4,4,
4,4,	4,4,	4,4,	2,0,
2,0,	2,0,	2,0,	2,0,
7,7,	7,7,	7,7,	7,7,
7,7,	7,7,	7,7,	7,7,
7,7,	7,7,	0,0,	0,0,
0,0,	0,0,	0,0,	0,0,
0,0,	7,7,	7,7,	7,7,
7,7,	7,7,	7,7,	7,7,
7,7,	7,7,	7,7,	7,7,
7,7,	7,7,	7,7,	7,7,
7,7,	7,7,	7,7,	7,7,
7,7,	7,7,	7,7,	0,0,
0,0,	0,0,	0,0,	0,0,
0,0,	7,7,	7,7,	7,7,
7,7,	7,7,	7,7,	7,7,
7,7,	7,7,	7,7,	7,7,
7,7,	7,7,	7,7,	7,7,
7,7,	7,7,	7,7,	7,7,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
7,7,    7,7,    7,7,    7,7,
7,7,    7,7,    7,7,    15,20,
0,0,    15,15,  15,15,  15,15,
15,15,  15,15,  15,15,  15,15,
15,15,  15,15,  15,15,  27,29,
27,29,  27,29,  27,29,  27,29,
27,29,  27,29,  27,29,  27,29,
27,29,  0,0,    15,21,  21,25,
0,0,    21,25,  0,0,    0,0,
21,26,  21,26,  21,26,  21,26,
21,26,  21,26,  21,26,  21,26,
21,26,  21,26,  29,30,  29,30,
29,30,  29,30,  29,30,  29,30,
29,30,  29,30,  29,30,  29,30,
0,0,    0,0,    0,0,    0,0,
0,0,    0,0,    15,21,  0,0,
0,0,    0,0,    0,0,    0,0,
0,0,    17,22,  17,22,  17,22,
17,22,  17,22,  17,22,  17,22,
17,22,  17,22,  17,22,  0,0,
0,0,    0,0,    0,0,    0,0,
0,0,    15,21,  17,23,  17,23,
17,23,  17,23,  17,23,  17,23,
17,23,  17,23,  17,23,  17,23,
17,23,  17,23,  17,23,  17,23,
17,23,  17,23,  17,23,  17,23,
17,23,  17,23,  17,23,  17,23,
21,25,  0,0,    0,0,    0,0,
0,0,    0,0,    17,23,  17,23,
17,23,  17,23,  17,23,  17,23,
17,23,  17,23,  17,23,  17,23,
17,23,  17,23,  17,23,  17,23,
17,23,  17,23,  17,23,  17,23,
17,23,  17,23,  17,23,  17,23,
17,23,  17,23,  17,23,  17,23,
18,18,  18,18,  18,18,  18,18,
18,18,  18,18,  18,18,  18,18,
18,18,  18,18,  0,0,    0,0,
0,0,    0,0,    0,0,    0,0,
0,0,    18,18,  18,18,  18,18,
18,18,  18,18,  18,18,  18,18,
18,18,  18,18,  18,18,  18,18,
18,18,  18,18,  18,18,  18,18,
18,18,  18,18,  18,18,  18,18,
18,18,  18,18,  18,18,  0,0,
0,0,    0,0,    0,0,    0,0,
0,0,    18,18,  18,18,  18,18,
18,18,  18,18,  18,18,  18,18,
18,18,  18,18,  18,18,  18,18,
18,18,  18,18,  18,18,  18,18,
18,18,  18,18,  18,18,  18,18,
18,18,  18,18,  18,18,  18,18,
18,18,  18,18,  18,18,  23,28,
23,28,  23,28,  23,28,  23,28,
23,28,  23,28,  23,28,  23,28,
23,28,  23,28,  23,28,  23,28,
23,28,  23,28,  23,28,  23,28,
23,28,  23,28,  23,28,  23,28,
23,28,  23,28,  23,28,  23,28,
23,28,  0,0,    0,0,    0,0,
0,0,    0,0,    0,0,    23,28,
23,28,  23,28,  23,28,  23,28,
23,28,  23,28,  23,28,  23,28,
23,28,  23,28,  23,28,  23,28,
23,28,  23,28,  23,28,  23,28,
23,28,  23,28,  23,28,  23,28,
23,28,  23,28,  23,28,  23,28,
23,28,  28,30,  28,30,  28,30,
28,30,  28,30,  28,30,  28,30,
28,30,  28,30,  28,30,  0,0,
0,0,    0,0,    0,0,    0,0,
0,0,    0,0,    28,28,  28,28,
28,28,  28,28,  28,28,  28,28,
28,28,  28,28,  28,28,  28,28,
28,28,  28,28,  28,28,  28,28,
28,28,  28,28,  28,28,  28,28,
28,28,  28,28,  28,28,  28,28,
0,0,    0,0,    0,0,    0,0,
0,0,    0,0,    28,28,  28,28,
28,28,  28,28,  28,28,  28,28,
28,28,  28,28,  28,28,  28,28,
28,28,  28,28,  28,28,  28,28,
28,28,  28,28,  28,28,  28,28,
28,28,  28,28,  28,28,  28,28,
0,0};
struct yysvf yysvec[] = {
```

เอกสารนี้เป็นเอกสารสงวนสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า กรณีใช้งานอื่น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0,      0,      0,
yycrank+-1,     0,              0,
yycrank+-128,   yysvec+1,       0,
yycrank+0,      0,              yyvstop+1,
yycrank+128,    0,              yyvstop+3,
yycrank+0,      0,              yyvstop+5,
yycrank+0,      0,              yyvstop+7,
yycrank+208,    0,              yyvstop+9,
yycrank+0,      0,              yyvstop+11,
yycrank+0,      0,              yyvstop+13,
yycrank+0,      0,              yyvstop+15,
yycrank+0,      0,              yyvstop+17,
yycrank+0,      0,              yyvstop+19,
yycrank+10,     0,              0,
yycrank+0,      0,              yyvstop+21,
yycrank+285,    0,              yyvstop+23,
yycrank+0,      0,              yyvstop+25,
yycrank+345,    0,              yyvstop+27,
yycrank+420,    0,              yyvstop+29,
yycrank+0,      0,              yyvstop+31,
yycrank+23,     0,              0,
yycrank+312,    0,              0,
yycrank+33,     0,              yyvstop+33,
yycrank+478,    yysvec+17,      yyvstop+36,
yycrank+36,     yysvec+20,      yyvstop+38,
yycrank+58,     0,              0,
yycrank+0,      yysvec+25,      yyvstop+40,
yycrank+295,    0,              yyvstop+42,
yycrank+553,    0,              yyvstop+45,
yycrank+322,    0,              yyvstop+47,
yycrank+0,      yysvec+29,      yyvstop+50,
0,      0,      0};
struct yywork *yytop = yycrank+675;
struct yysvf *yybgin = yysvec+1;
char yymatch[] = {
00  ,01  ,01  ,01  ,01  ,01  ,01  ,01  ,
01  ,011 ,011 ,01  ,01  ,01  ,01  ,01  ,
01  ,01  ,01  ,01  ,01  ,01  ,01  ,01  ,
01  ,01  ,01  ,01  ,01  ,01  ,01  ,01  ,
011 ,01  ,01  ,01  ,01  ,01  ,01  ,01  ,
01  ,01  ,01  ,'+' ,01  ,'+' ,01  ,01  ,
'0' ,'0' ,'0' ,'0' ,'0' ,'0' ,'0' ,'0' ,
'0' ,'0' ,01  ,01  ,01  ,01  ,01  ,01  ,
01  ,'A' ,'A' ,'A' ,'A' ,'E' ,'A' ,'A' ,
'A' ,'A' ,'A' ,'A' ,'A' ,'A' ,'A' ,'A' ,
'A' ,'A' ,'A' ,'A' ,'A' ,'A' ,'A' ,'A' ,
'A' ,'A' ,'A' ,01  ,01  ,01  ,01  ,01  ,
01  ,'A' ,'A' ,'A' ,'A' ,'E' ,'A' ,'A' ,
'A' ,'A' ,'A' ,'A' ,'A' ,'A' ,'A' ,'A' ,
'A' ,'A' ,'A' ,'A' ,'A' ,'A' ,'A' ,'A' ,
'A' ,'A' ,'A' ,01  ,'|' ,01  ,01  ,01  ,
0};
char yyextra[] = {
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0};
#ifndef lint
static  char ncform_sccsid[] = "@(#)ncform 1.6 88/02/08 SMI"; /* from S5R2 1.2 */
#endif

int yylineno =1;
# define YYU(x) x
# define NLSTATE yyprevious=YYNEWLINE
char yytext[YYLMAX];
struct yysvf *yylstate [YYLMAX], **yylsp, **yyolsp;
char yysbuf[YYLMAX];
char *yysptr = yysbuf;
int *yyfnd;
extern struct yysvf *yyestate;
int yyprevious = YYNEWLINE;
yylook(){
        register struct yysvf *yystate, **lsp;
        register struct yywork *yyt;
        struct yysvf *yyz;
        int yych, yyfirst;
        struct yywork *yyr;
# ifdef LEXDEBUG
        int debug;
# endif
        char *yylastch;
        /* start off machines */
# ifdef LEXDEBUG
        debug = 0;
# endif
        yyfirst=1;
        if (!yymorfg)
                yylastch = yytext;
        else {
                yymorfg=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ทำซ้ำ ดัดแปลง หรือนำไปเผยแพร่ต่อ ทั้งนี้ห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                yylastch = yytext+yyleng;
                }
        for(;;){
                lsp = yylstate;
                yyestate = yystate = yybgin;
                if (yyprevious==YYNEWLINE) yystate++;
                for (;;){
# ifdef LEXDEBUG
                        if(debug)fprintf(yyout,"state %d\n",yystate-yysvec-1);
# endif
                        yyt = yystate->yystoff;
                        if(yyt == yycrank && !yyfirst){  /* may not be any transitions */
                                yyz = yystate->yyother;
                                if(yyz == 0)break;
                                if(yyz->yystoff == yycrank)break;
                                }
                        *yylastch++ = yych = input();
                        yyfirst=0;
                tryagain:
# ifdef LEXDEBUG
                        if(debug){
                                fprintf(yyout,"char ");
                                allprint(yych);
                                putchar('\n');
                                }
# endif
                        yyr = yyt;
                        if ( (int)yyt > (int)yycrank){
                                yyt = yyr + yych;
                                if (yyt <= yytop && yyt->verify+yysvec == yystate){
                                        if(yyt->advance+yysvec == YYLERR)       /* error tra
itions */
                                                {unput(*--yylastch);break;}
                                        *lsp++ = yystate = yyt->advance+yysvec;
                                        goto contin;
                                        }
                                }
# ifdef YYOPTIM
                        else if((int)yyt < (int)yycrank) {              /* r < yycrank */
                                yyt = yyr = yycrank+(yycrank-yyt);
# ifdef LEXDEBUG
                                if(debug)fprintf(yyout,"compressed state\n");
# endif
                                yyt = yyt + yych;
                                if(yyt <= yytop && yyt->verify+yysvec == yystate){
                                        if(yyt->advance+yysvec == YYLERR)       /* error tra
itions */
                                                {unput(*--yylastch);break;}
                                        *lsp++ = yystate = yyt->advance+yysvec;
                                        goto contin;
                                        }
                                yyt = yyr + YYU(yymatch[yych]);
# ifdef LEXDEBUG
                                if(debug){
                                        fprintf(yyout,"try fall back character ");
                                        allprint(YYU(yymatch[yych]));
                                        putchar('\n');
                                        }
# endif
                                if(yyt <= yytop && yyt->verify+yysvec == yystate){
                                        if(yyt->advance+yysvec == YYLERR)       /* error tra
ition */
                                                {unput(*--yylastch);break;}
                                        *lsp++ = yystate = yyt->advance+yysvec;
                                        goto contin;
                                        }
                                }
                        if ((yystate = yystate->yyother) && (yyt= yystate->yystoff) != yycra
){
# ifdef LEXDEBUG
                                if(debug)fprintf(yyout,"fall back to state %d\n",yystate-yys
c-1);
# endif
                                goto tryagain;
                                }
# endif
                        else
                                {unput(*--yylastch);break;}
                contin:
# ifdef LEXDEBUG
                        if(debug){
                                fprintf(yyout,"state %d char ",yystate-yysvec-1);
                                allprint(yych);
                                putchar('\n');
                                }
# endif
                        }
                }
# ifdef LEXDEBUG
                if(debug){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

```
                        fprintf(yyout,"stopped at %d with ",*(lsp-1)-yysvec-1);
                        allprint(yych);
                        putchar('\n');
                        }
# endif
                while (lsp-- > yylstate){
                        *yylastch-- = 0;
                        if (*lsp != 0 && (yyfnd= (*lsp)->yystops) && *yyfnd > 0){
                                yyolsp = lsp;
                                if(yyextra[*yyfnd]){            /* must backup */
                                        while(yyback((*lsp)->yystops,-*yyfnd) != 1 && lsp > y
lstate){
                                                lsp--;
                                                unput(*yylastch--);
                                                }
                                        }
                                yyprevious = YYU(*yylastch);
                                yylsp = lsp;
                                yyleng = yylastch-yytext+1;
                                yytext[yyleng] = 0;
# ifdef LEXDEBUG
                                if(debug){
                                        fprintf(yyout,"\nmatch ");
                                        sprint(yytext);
                                        fprintf(yyout," action %d\n",*yyfnd);
                                        }
# endif
                                return(*yyfnd++);
                                }
                        unput(*yylastch);
                        }
                if (yytext[0] == 0  /* && feof(yyin) */)
                        {
                        yysptr=yysbuf;
                        return(0);
                        }
                yyprevious = yytext[0] = input();
                if (yyprevious>0)
                        output(yyprevious);
                yylastch=yytext;
# ifdef LEXDEBUG
                if(debug)putchar('\n');
# endif
                }
        }
yyback(p, m)
        int *p;
{
if (p==0) return(0);
while (*p)
        {
        if (*p++ == m)
                return(1);
        }
return(0);
}
        /* the following are only used in the lex library */
yyinput(){
        return(input());
        }
yyoutput(c)
  int c; {
        output(c);
        }
yyunput(c)
   int c; {
        unput(c);
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้