

พริยญาณิพนธ์

โปรแกรมพจนานุกรมกริยาวลี

THE PHRASAL VERB DICTIONARY PROGRAM



พริยญาณิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์

ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง โปรแกรมพจนานุกรมกริยาวลี

THE PHRASAL VERB DICTIONARY PROGRAM

ผู้จัดทำ

1. นายชูกิจ ศรีใหม่
2. นายเชิงชาย ไชยชิน
3. นายวัชร ชพานนท์
4. นายอนุเชษฐ อ่อนคง

อาจารย์ที่ปรึกษา

ลงนาม

(อาจารย์วรวิทย์ สมหา)

ลงนาม

(อาจารย์สุชิน อาจหาญ)

ลงนาม

(อาจารย์สันติ ตันตระกูล)

หัวหน้าภาควิชา

ลงนาม

(ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมพจนานุกรมกริยาวลี

นายชูกิจ ศรีใหม่
นายเชิงชาย ไชยชิน
นายวัชระ ชพานนท์
นายอนุเชษฐ อ่อนคง

อาจารย์ที่ปรึกษา
อาจารย์วรวิทย์ สมหา
อาจารย์สุชิน อากาศ
อาจารย์สันติ ตันตระกูล
ปีการศึกษา 2537

บทคัดย่อ

ปริญญาานิพนธ์ ชุดนี้ได้นำเสนอโปรแกรมพจนานุกรมกริยาวลีรวมทั้งเพิ่มเติมสำนวน และ
สะแลง โปรแกรมประยุกต์ชุดนี้ใช้เพื่อศึกษาเรื่องต่างๆ ดังเช่น โปรแกรมประมวลคำภาษาไทยและ
ภาษาอังกฤษ ระบบฐานข้อมูลไบนารีทรี ระบบฐานข้อมูลเชิงสัมพันธ์ การใช้ระบบกราฟฟิกส์มาเป็น
ตัวกลางสื่อความหมายระหว่างผู้ใช้เครื่องคอมพิวเตอร์กับผู้ใช้เครื่องคอมพิวเตอร์ การออกแบบ
โปรแกรมช่วยสอนและโปรแกรมทดสอบการเรียนรู้คำศัพท์ เป็นต้น

ชุดโปรแกรมพจนานุกรมกริยาวลีนี้ สร้างขึ้นมาเพื่อใช้เรียนวิธีใช้งานคำกริยาวลี คำบุรพบท
สำนวนและสะแลง เพื่อนำไปใช้ในการสร้างรูปประโยคพร้อมกับทดสอบความรู้

THE PHRASAL VERB DICTIONARY PROGRAM

MR. CHOOKIT	SRIMAI
MR. CHERNGCHAI	CHAICHIN
MR. WATCHARA	CHAPARNON
MR. ANUCHAET	ONKONG

ADVISOR

MR. WORAVIT	SOMHA
MR. SUCHIN	ADHAN
MR. SUNTI	TUNTRAKUL

1994

ABSTRACT :

THIS THESIS PRESENTS THE PHRASAL VERB DICTIONARY PROGRAM INCLUDING VERBAL IDIOMS AND SLANG. FOR THE APPLICATION OF THE PHRASAL VERB DICTIONARY PROGRAM IS USED TO LEARN IN SOMETHING. FOR EXAMPLES ARE THAI/ENGLISH WORD PROCESSOR PROGRAM, BINARY TREE DATABASE SYSTEM, RELATIONAL DATABASE MANAGEMENT SYSTEM, GRAPHICS USER INTERFACE(GUI), COMPUTER PROGRAM, ETC.

THIS PROGRAM APPLICATION WAS CREATED FOR USING PHRASAL VERB, PREPOSITION, VERBAL IDIOMS AND SLANG IN ORDER TO BUILD ENGLISH SENTENCE AND KNOWLEDGE TESTING.

ปริญญานิพนธ์

เรื่อง โปรแกรมพจนานุกรมกริยาวลี THE PHRASAL VERB DICTIONARY PROGRAM

จุดประสงค์

1. เพื่อศึกษาการสร้างโปรแกรมประยุกต์พจนานุกรมกริยาวลี
2. เพื่อศึกษาการออกแบบระบบโปรแกรม TOP DOWN DESIGN
3. เพื่อศึกษานำภาษาซี มาประยุกต์ใช้งาน
4. เพื่อศึกษาการทำงานโปรแกรมประมวลผลคำภาษาไทยและภาษาอังกฤษ
5. เพื่อศึกษาระบบฐานข้อมูลแบบไม่นารีทรีและระบบจัดฐานข้อมูลเชิงสัมพันธ์
6. เพื่อศึกษานำระบบกราฟฟิกส์มาใช้สื่อความหมายระหว่างเครื่องคอมพิวเตอร์กับผู้ใช้คอมพิวเตอร์
7. เพื่อสร้างโปรแกรมช่วยสอน
8. เพื่อประมวลผลความรู้ที่ผู้เรียนได้เรียนรู้จากโปรแกรม
9. เพื่อนำไปใช้แก้ปัญหาการสร้างประโยคภาษาอังกฤษให้สมบูรณ์และเป็นที่ยอมรับใช้กันทั่วไป

ประโยชน์คาดว่าจะได้รับ

1. เพิ่มทักษะการเขียนโปรแกรมด้วยภาษาซี
2. มีทักษะในการสร้างประโยคภาษาอังกฤษให้ถูกต้องยิ่งขึ้น
3. เป็นการบุกเบิกการนำความรู้ระบบฐานข้อมูลระบบประมวลคำและระบบกราฟฟิกส์มาประยุกต์ใช้งานร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิติกรรมประกาศ

การที่ปริญญาานิพนธ์สำเร็จลุล่วงไปด้วยดีนั้น คณะผู้จัดทำปริญญาานิพนธ์เรื่อง
โปรแกรมพจนานุกรมกริยาวลี จึงเรียนมาเพื่อขอขอบพระคุณอาจารย์ที่ปรึกษาและรวมทั้งอาจารย์
ประจำภาควิชาครุศาสตร์ศึกษาศาสตร์ทุกท่านเป็นอย่างมาก ที่ได้ให้ข้อเสนอแนะและแนวทางการแก้
ปัญหาต่างๆ พร้อมทั้งขอขอบคุณเพื่อนๆ ครุศาสตร์ศึกษาศาสตร์รุ่น 15 ทุกท่านที่ได้ให้คำแนะนำและ
เป็นกำลังใจในการทำปริญญาานิพนธ์ในครั้งนี้และสุดท้ายนี้ขอขอบคุณเพื่อนร่วมงานทุกคนที่ได้ให้
ความร่วมมือเป็นอย่างดีจนกระทั่งปริญญาานิพนธ์นี้สำเร็จลุล่วงไปด้วยดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1. บทนำ	1
บทที่ 2. ทฤษฎีและหลักการ	2
2.1 ฐานข้อมูลขั้นพื้นฐาน	3
2.1.1 ฐานข้อมูล	3
2.1.2 องค์ประกอบข้อมูล	4
2.1.3 เพิ่ม	7
2.2 การออกแบบฐานข้อมูล	8
2.2.1 ขั้นตอนการออกแบบฐานข้อมูล	8
2.2.2 พิจารณาความต้องการขั้นพื้นฐานของฐานข้อมูล	8
2.2.3 กำหนดความต้องการเกี่ยวกับหน้าที่และความสามารถของฐานข้อมูล	9
2.2.4 แสดงหัวข้อต่างๆของข้อมูล	9
2.2.5 แยกองค์ประกอบข้อมูลออกจากเพิ่มข้อมูล	10
2.2.6 สร้างพจนานุกรมขององค์ประกอบข้อมูล	10
2.2.7 การกำหนดลักษณะเฉพาะหรือวิธีการเรียกใช้ข้อมูลของแต่ละไฟล์	11
2.2.8 บอกลักษณะตามสัมพันธ์ระหว่างไฟล์	12
2.3 ระบบจัดการฐานข้อมูล	13
2.3.1 DBMS	13
2.3.2 หลักการพื้นฐานของระบบจัดการฐานข้อมูล	14
2.4 ภาษาซีกกับภาษากำหนดลักษณะข้อมูล	16
2.4.1 แบบแผนฐานข้อมูล	16
2.4.2 โปรแกรมประยุกต์	16
2.5 ข้อมูลภาษาซี	21
2.5.1 ภาษากำหนดลักษณะข้อมูลภาษาซี	21
2.6 โครงสร้างข้อมูลแบบต้นไม้	23
2.6.1 นิยามโครงสร้างต้นไม้	23
2.6.2 ต้นไม้ที่มีแบบแผน	26
2.6.3 การแทนโครงสร้างต้นไม้ในคอมพิวเตอร์	27

เรื่อง	หน้า
2.6.4 ต้นไม้แบบไบนารี	29
2.7 เพิ่มข้อมูล	29
2.7.1 ความหมายของเพิ่มข้อมูลและลักษณะการประมวลผลเพิ่มข้อมูล	29
2.7.2 ฟังก์ชันที่ใช้ในการประมวลผลเพิ่มข้อมูล	32
2.7.3 ชนิดของเพิ่มข้อมูล	32
2.7.4 การเปิดเพิ่มข้อมูล	33
2.8 สถาปัตยกรรมของ IBM PC และแผนการจัดการหน่วยความจำภาษาซี	33
2.9 โครงสร้างหน่วยความจำของโปรแกรมภาษาซี	35
2.10 โครงสร้างหน่วยความจำและขนาด stack	36
2.11 CRC	38
2.11.1 คุณภาพของ CRC - 32	39
2.11.2 CRCMAN	40
2.11.3 อธิบายโปรแกรม	41
2.11.4 การสร้างไฟล์เก็บ CRC	42
2.11.5 การคำนวณ CRC - 32 ในโปรแกรม	43
2.11.6 การตรวจสอบไฟล์	43
2.11.7 การใช้โปรแกรม	43
2.12 สอนภาษาซีให้รู้จักภาษาไทย	44
2.12.1 สร้างคำสั่ง	45
2.13 ทฤษฎีภาษาอังกฤษ	46
2.13.1 วลี	46
2.13.2 วลีและประโยคทำหน้าที่คล้ายคำวิเศษณ์	46
2.13.3 คำบุรพบท	47
บทที่ 3. การออกแบบและการสร้าง	50
3.1 ด้านเนื้อหาภาษาอังกฤษ	50
3.2 ด้านเทคนิควิธีเขียนโปรแกรม	50
3.2.1 ศึกษากลุ่มเป้าหมายที่ใช้โปรแกรม	50
3.2.2 ออกแบบระบบ	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง	หน้า
3.2.3 คำเนิการเขียนโปรแกรม	50
3.2.4 ทดสอบความถูกต้อง	50
3.2.5 นำออกไปใช้งาน	50
3.3 ด้านฐานข้อมูล	50
บทที่ 4. การใช้งานโปรแกรมพจนานุกรม กริยาวลี	63
4.1 ขั้นตอนการทำโปรแกรมโดยใช้เทอร์โบซี	63
4.2 การใช้งานโปรแกรมพจนานุกรม กริยาวลี	64
4.2.1 ส่วนช่วยเหลือจากโปรแกรม	65
4.2.2 ฉับ	66
4.2.3 คัพท	68
4.2.4 ปรับความเร็วเมาส์	81
4.2.5 ซ่อมแซม	81
4.2.6 ระบบคอส	82
4.2.7 ออกจากโปรแกรม	82
บทที่ 5. บทสรุปและวิจารณ์	83
5.1 ปัญหาที่พบในการทำโครงการ	83
5.2 แก้ไขปัญหา	83
5.3 แนวทางการพัฒนา	83
5.4 สรุปผลของโครงการ	84
ภาคผนวก	
ภาคผนวก ก โฟลว์ชาร์ทการทำงานของโปรแกรมพจนานุกรม กริยาวลี	85
ภาคผนวก ข แสดงปุมฟังก์ชันคีย์ควบคุมการทำงานโปรแกรมพจนานุกรม กริยาวลี	99
ภาคผนวก ค แสดงการจัดกลุ่มไฟล์ตามหน้าที่การทำงาน	101
ภาคผนวก ง โปรแกรมพจนานุกรมกริยาวลี	103
บรรณานุกรม	375

สารบัญรูปภาพ

รูปภาพ	หน้า
รูปที่ 2.1 ระบบฐานข้อมูล DBMS	13
รูปที่ 2.2 โครงสร้างภาษากำหนดลักษณะข้อมูล	15
รูปที่ 2.3 ฐานข้อมูลแบบลำดับขั้น	15
รูปที่ 2.4 ฐานข้อมูลแบบเครือข่าย	16
รูปที่ 2.5 รูปแบบแผนข้อมูล	17
รูปที่ 2.6 ลักษณะโครงสร้างต้นไม้	23
รูปที่ 2.7 แสดงลักษณะของต้นไม้ตามนิยามแบบรีเคอร์ซีฟ	24
รูปที่ 2.8 การเรียกชื่อส่วนต่างๆของโครงสร้างต้นไม้	24
รูปที่ 2.9 ป่าที่ประกอบด้วยต้นไม้ A,B และ X	26
รูปที่ 2.10 ต้นไม้ A และ B จะแตกต่างกันถ้ากำหนดเป็นต้นไม้ที่มีแบบแผน	26
รูปที่ 2.11 การแทนต้นไม้	28
รูปที่ 2.12 การแทนโครงสร้างต้นไม้	28
รูปที่ 2.13 ต้นไม้ไบนารี และไม่ใช่ต้นไม้ไบนารี	30
รูปที่ 2.14 โครงสร้างต้นไม้ไบนารี LSON	30
รูปที่ 2.15 โครงสร้างหน่วยความจำของโมโครซอฟท์ซี โมเดล small	36
รูปที่ 2.16 โครงสร้างหน่วยความจำของ เทอร์โบซี โมเดล small	37
รูปที่ 2.17 โครงสร้างหน่วยความจำแบบ large	37
รูปที่ 4.1 การแปลโปรแกรม	63
รูปที่ 4.2 การเชื่อมโยง	64
รูปที่ 4.3 โลโก้ของโปรแกรม	64
รูปที่ 4.4 เมนูหลักของโปรแกรม	65
รูปที่ 4.5 ส่วนช่วยเหลือของโปรแกรม	66
รูปที่ 4.6 เมนูในส่วนของฉัน	66
รูปที่ 4.7 แสดงชื่อของผู้ให้คำปรึกษาในการจัดทำโปรแกรม	67
รูปที่ 4.8 แสดงชื่อและหน้าที่ของผู้จัดทำโปรแกรม	67
รูปที่ 4.9 เริ่มต้นเข้าสู่เมนูศัพท์	68
รูปที่ 4.10 เมนูของศัพท์	68

รูปภาพ	หน้า
รูปที่ 4.11 เมนูในส่วนค้นหาศัพท์	69
รูปที่ 4.12 การค้นหาศัพท์ในกรณีที่ศัพท์มีอยู่ในโปรแกรม	70
รูปที่ 4.13 อธิบายความหมายศัพท์	70
รูปที่ 4.14 ตัวอักษรแบบมาตรฐาน	71
รูปที่ 4.15 ตัวอักษรแบบตัวเอียง	71
รูปที่ 4.16 ตัวอักษรแบบสองชั้น	72
รูปที่ 4.17 ตัวอักษรแบบตัวเล็ก	72
รูปที่ 4.18 ค้นหาศัพท์ในกรณีศัพท์ในโปรแกรมไม่มี	73
รูปที่ 4.19 บันทึกไฟล์คำศัพท์	74
รูปที่ 4.20 ต้องการออกจากเมนูค้นหาศัพท์หรือไม่	74
รูปที่ 4.21 เมนูศัพท์	75
รูปที่ 4.22 เมนูเพิ่มศัพท์	75
รูปที่ 4.23 กรณีคำศัพท์ที่ไม่สามารถเพิ่มศัพท์	76
รูปที่ 4.24 คำถามของการเพิ่มศัพท์	76
รูปที่ 4.25 กรณีมีศัพท์อยู่ในโปรแกรม	77
รูปที่ 4.26 ออกจากเมนูการเพิ่มศัพท์	78
รูปที่ 4.27 เมื่อเริ่มเข้าเมนูเรียนรู้	78
รูปที่ 4.28 หน้าต่างให้ตัดสินใจ	79
รูปที่ 4.29 คำศัพท์จะแสดงออกมา	79
รูปที่ 4.30 เมnulงทะเบียน	80
รูปที่ 4.31 เมนูเลือกตัวอักษร	80
รูปที่ 4.32 เมนูของแม่เ้า	81
รูปที่ 4.33 ความพร้อมของการซ่อมแซม	82
รูปที่ 4.34 กำลังซ่อมแซม	82

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 ตารางพจนานุกรมองค์ประกอบข้อมูล	5
ตารางที่ 2.2 การสร้างพจนานุกรมองค์ประกอบข้อมูล	22
ตารางที่ 2.3 โครงสร้างโหนดของต้นไม้ทั่วไป	27
ตารางที่ 2.4 ลักษณะการประมวลเพิ่มข้อมูล	31
ตารางที่ 2.5 ลักษณะของเพิ่มข้อมูล	33



บทที่ 1

บทนำ

ในปัจจุบันนี้ซึ่งเข้าสู่ยุคโลกาภิวัตน์จำเป็นจะต้องเตรียมความพร้อมเพื่อความเข้าใจและมีประสิทธิภาพในการติดต่อสื่อสารซึ่งสิ่งหนึ่งที่ใช้ในการติดต่อสื่อสารเพื่อความเข้าใจกันก็คือภาษาพูดหรือเขียนที่มนุษย์ได้ประดิษฐ์คิดค้นขึ้นมากมาย โดยมีลักษณะแตกต่างกันตามแต่ละภูมิภาคของโลก แต่อย่างไรก็ดีภาษาที่ใช้เป็นมาตรฐานสากลก็คือภาษาอังกฤษนำไปใช้กันเป็นที่แพร่หลายในวงการต่างๆ เช่นการศึกษา การติดต่อธุรกิจ เป็นต้น จะเห็นว่าในสมัยก่อนๆ นั้นจะใช้ตำราเป็นนุคคละทำหน้าที่เป็นสื่อกลางในการสร้างความเข้าใจกันของแต่ละฝ่าย แต่ในปัจจุบันนี้ความรวดเร็วในการสื่อสารทำให้ไม่สามารถฟังล่ำได้ตลอดเวลาอีกทั้งคำพูดใหม่ๆ เกิดขึ้นได้ตลอดเวลาตามความเจริญก้าวหน้าที่เกิดขึ้นด้วยเหตุผลนี้จึงมีการรวบรวมคำศัพท์ทั้งหลายเหล่านั้น เพื่อใช้เป็นแหล่งอ้างอิงในการนำมาใช้ในชีวิตประจำวัน ให้สามารถค้นหาข้อมูลที่ต้องการได้อย่างรวดเร็วเสียค่าใช้จ่ายน้อย มีรูปแบบที่เร้าความสนใจในการเรียนรู้พร้อมทั้งสามารถประเมินผลได้ และสามารถเพิ่มเติมข้อมูลใหม่ๆ ได้อย่างรวดเร็วยิ่งขึ้น

จากปัญหาต่างๆ ที่เกิดขึ้นมานั้นจึงได้มีการเขียนโปรแกรมพจนานุกรมกริยาวลีและสะแลงนี้ได้รับรรจุคำศัพท์ไว้มากกว่า 2,000 คำ พร้อมทั้งได้มีการยกตัวอย่างการนำศัพท์ที่มีอยู่ไปใช้งานจริงๆ ซึ่งรวบรวมค้นคว้ามาจากตำราหลายๆ เล่ม ทำให้ได้คำศัพท์มากที่สุดเท่าที่จะเป็นไปได้นอกจากนี้ยังพร้อมที่จะรับคำศัพท์ใหม่ๆ เพิ่มเติมตามการใช้งานของผู้ใช้งาน โดยโปรแกรมนี้จะทำงานบนเครื่องคอมพิวเตอร์และระบบปฏิบัติการบนคอสตั้งแต่เวอร์ชัน 3.30 ขึ้นไป จอภาพที่ใช้เป็นจอภาพแบบ วีจีเอ และใช้งานบนฮาร์ดดิสก์ 2 เมกะไบต์

บทที่ 2

ทฤษฎีและหลักการ

ในบทนี้ครอบคลุมเนื้อหาวิธีการใช้ภาษาซีเพื่อการออกแบบและพัฒนาโปรแกรมจัดการฐานข้อมูล ช่วยแนะนำขั้นพื้นฐานสำหรับหลักการของการจัดการฐานข้อมูลนอกจากนี้ในบทนี้ยังมีตัวอย่างของการจัดการข้อมูล (DBMS) ที่สมบูรณ์อีกด้วย

ผู้อ่านจะได้รับประโยชน์มากมายจากการอ่านบทนี้ ตั้งแต่ความรู้พื้นฐานทางฐานข้อมูลและการเขียนโปรแกรมภาษาซี อย่างไรก็ตามในบทนี้ก็ไม่ได้สอนการเขียนโปรแกรม แต่สอนการนำอรรถประโยชน์ของภาษาซีมาใช้สร้างระบบซอฟต์แวร์ การเชื่อมโยงโมดูลเข้าด้วยกันตลอดจนเทคนิคในการเขียนโปรแกรม นอกจากนี้ซอร์สโค้ดที่ปรากฏในบทนี้ยังมีประโยชน์ที่จะนำไปใช้ประโยชน์ในการนำไปประยุกต์กับงานทางด้านฐานข้อมูลจริงๆ หรืออาจนำฟังก์ชันบางส่วนไปใช้ประยุกต์กับงานด้านอื่นๆ และจุดมุ่งหมายที่สำคัญในบทนี้ก็คือให้ผู้อ่านสามารถสร้างเครื่องมือสำหรับใช้งานได้ด้วยตนเองตลอดจนเห็นความสำคัญของการสร้างเครื่องมือ

เครื่องมือของซอฟต์แวร์ในบทนี้ ได้เน้นหนักในส่วนที่การสร้างฟังก์ชันด้วยภาษาซีที่จะสามารถนำมาใช้ได้ อีก โดยจะเรียกใช้ฟังก์ชันเหล่านี้จากโปรแกรมประยุกต์ใจความสำคัญของเครื่องมือของซอฟต์แวร์มีอยู่ว่า ระบบซอฟต์แวร์ส่วนใหญ่สร้างขึ้นโดยมีส่วนประกอบร่วมกันบางอย่างส่วนประกอบที่เหมือนกันนี้ สามารถพัฒนาขึ้นมาเป็นฟังก์ชันเพื่อใช้กับระบบใดๆ ที่มีส่วนประกอบดังกล่าว โดยที่ฟังก์ชันเหล่านี้ต้องเขียนขึ้นเป็นโมดูลที่สามารถจะนำมาใช้ได้อีกฟังก์ชันนี้ก็คือเครื่องมือของซอฟต์แวร์นั่นเอง ยังมีชุดของเครื่องมือที่สมบูรณ์มากขึ้นจำนวน โปรแกรมที่จะต้องเขียนโปรแกรมยิ่งน้อยลง

การสร้างเครื่องมือขึ้นมาได้ ต้องรู้วิธีที่จะเขียนฟังก์ชันที่สามารถนำขึ้นมาใช้ได้และวิธีเก็บฟังก์ชันไว้ในไลบรารีนอกจากนี้อีกทางหนึ่ง ที่จะเป็นนักสร้างเครื่องมือทำได้โดยเรียนรู้จากคู่มือและประสบการณ์ของนักเขียนโปรแกรมคนอื่นๆ ซอฟต์แวร์ในบทนี้ ประกอบด้วยกลุ่มของฟังก์ชันที่นักเขียนโปรแกรมได้รวบรวมไว้ในระบบซอฟต์แวร์ ซึ่งถือเป็นเครื่องมือและเป็นส่วนหนึ่งของการประยุกต์ใช้งานทั้งหมด เครื่องมือเหล่านี้มีสิ่งซึ่งระบบต่างๆ ต้องการใช้งานร่วมกัน ดังนั้นการใช้งานเหล่านี้ จะช่วยลดจำนวนโปรแกรมที่ต้องเขียนและเวลาที่ใช้ในการประยุกต์ขึ้นมาโดยมองภาพได้ว่าเครื่องมือเหล่านี้ก็คือ ส่วนที่อยู่ระหว่างโปรแกรมที่เขียนเพื่อใช้กับฮาร์ดแวร์นั้นๆ กับตัวระบบปฏิบัติการ ถ้าเครื่องมือได้รับการพิสูจน์แล้วว่าเชื่อถือได้มากขึ้นตามไปด้วย

สำหรับ โปรแกรมเมอร์บางคนที่ต้องการพัฒนางานใหม่ขึ้น ก็สามารถทำได้โดยเขียนโปรแกรมเพิ่มขึ้นจากงานเก่าเพียงเล็กน้อยคือ อาจจะใช้ความชำนาญประกอบ กับความรู้ดัดแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานเก่าที่มีอยู่ให้ใช้กับงานใหม่ได้ นอกจากนี้เมื่อศึกษาจากเครื่องมือที่มีอยู่ก็สามารถใช้กับเครื่องมือที่มีอยู่เพื่อการประยุกต์ใช้งานในอนาคตได้

จุดประสงค์ในบทนี้คือการเตรียมเครื่องมือของซอฟต์แวร์แบบสมบูรณ์ที่มีประโยชน์ เครื่องมือเหล่านี้ใช้และเข้าใจง่าย ซอฟต์แวร์ที่สร้างขึ้นโดยเครื่องมือนี้สามารถนำไปใช้งานได้จริง และสามารถประยุกต์ใช้มันกับความรู้ที่มีอยู่ในการพัฒนาระบบฐานข้อมูลได้

ในที่นี้ผู้อ่านจำเป็นต้องมีพื้นฐานความรู้เรื่องการโปรแกรมในเครื่องคอมพิวเตอร์และการพัฒนาโปรแกรมภายใต้ระบบปฏิบัติการของคอส นอกจากนี้ยังต้องเข้าใจ โครงสร้างของไฟล์ (file structure) ไครเรกทอรี (directory) และซัพไครเรกทอรี (sub-directory) ของระบบด้วยประกอบกับความรู้อื่นๆ

2.1 ฐานข้อมูลขั้นพื้นฐาน

เนื้อหาของบทนี้จะเกี่ยวข้องโดยตรงกับส่วนของซอฟต์แวร์โดยจะประกอบไปด้วยโปรแกรม (source code) ของระบบปฏิบัติการฐานข้อมูลและสารสนเทศ (Information) ของตัวโปรแกรมนี้ด้วยซึ่งจะกล่าวถึงในส่วนต่อไปสำหรับโปรแกรมที่คั้นจำเป็นต้องมีข้อมูลสารสนเทศในการกระบวนกรต่างๆ ในทางกลับกันสารสนเทศจะมีประโยชน์ได้นั้นก็ต้องพึ่งการดำเนินการกระบวนกรต่างๆ ที่มีประสิทธิภาพและเพื่อที่จะสร้างสารสนเทศที่มีความเหมาะสม สำหรับการดำเนินการในกระบวนกรแต่ละกระบวนกรนั้น ก็จำเป็นต้องจัดองค์ประกอบของมัน ให้อยู่ในลักษณะที่เป็นระเบียบและมีรูปแบบแน่นอน ลักษณะรูปแบบที่กล่าวมานั้นเรียกว่า “ฐานข้อมูล” (data base)

2.1.1 ฐานข้อมูล

ในแต่ละไฟล์ของฐานข้อมูลหนึ่งๆ จะประกอบด้วยองค์ประกอบข้อมูล (data element) ซึ่งได้แก่ หมายเลขประจำตัว วันที่ ตำแหน่งที่อยู่เป็นต้น แล้วแต่ชนิดข้อมูลองค์ประกอบข้อมูลเหล่านี้ จะได้รับการจัดระเบียบ จัดเก็บ และการเรียกใช้เมื่อต้องการจากฐานข้อมูลนั้นๆ ดังนั้นในการที่เรา จะทำการออกแบบฐานข้อมูลหนึ่ง จึงมีความจำเป็นอย่างยิ่งที่จะต้องเข้าใจเกี่ยวกับระบบไฟล์องค์ประกอบข้อมูล ตลอดจนวิธีการใช้งานมันด้วย

สำหรับผู้ที่จะทำการออกแบบฐานข้อมูลนั้น ควรจะทราบความหมายของข้อมูลเสียก่อน ข้อมูลหมายถึง สารสนเทศที่ได้รับการจัดเรียบร้อย เช่น สารสนเทศที่อยู่ในรูปแบบที่คอมพิวเตอร์สามารถนำมาทำการประมวลผลได้ทันทีเป็นต้น สารสนเทศนั้นๆ จะสามารถใช้งานได้ก็ต่อเมื่อมัน ถูกจัดรูปและเปลี่ยนแปลงอยู่ในรูปแบบที่เครื่องคอมพิวเตอร์สามารถอ่านได้นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบที่เล็กที่สุดของข้อมูลภายในคอมพิวเตอร์เรียกว่า "บิต" (bit or binary digit) โดยเป็นองค์ประกอบของเลขฐานสองนั่นคือมีค่าเพียง 0 กับ 1 นั้น ซึ่งในบางครั้งโปรแกรมเมอร์ก็อาจมีความจำเป็นที่จะเข้ามาเกี่ยวข้องกับข้อมูลในลักษณะบิต แต่สำหรับผู้ออกแบบฐานข้อมูลแล้วไม่มีความจำเป็นถึงเพียงนั้น เขาจะมองข้อมูลในระดับที่สูงกว่า โดยบิตแต่ละบิตจะประกอบเข้ากันเป็นไบต์หรืออักขระ (character) ซึ่งจะรวมกันเป็นองค์ประกอบข้อมูลอีกทีหนึ่งในส่วนของแฟ้มข้อมูลนั้น แต่ละแฟ้มสามารถแยกย่อยได้เป็นระเบียบ (record) ซึ่งแต่ละระเบียบก็เกิดจากการรวมตัวกันขององค์ประกอบข้อมูลอีกทีหนึ่งและในท้ายที่สุดเมื่อนำเอาแฟ้มข้อมูลหลายๆ แฟ้มมารวมกันก็จะได้เป็นฐานข้อมูล

2.1.2 องค์ประกอบข้อมูล

ในฐานข้อมูลนั้นจะถูกเพิ่มเติมด้วยหัวข้อ (item) ต่างๆ ของสารสนเทศค่าของหัวข้อต่างๆ เหล่านี้ จำเป็นต้องมีเนื้อหาบางส่วนสำหรับใช้เก็บเนื้อที่สำหรับข้อมูลในแต่ละชั้นเรียกว่า "องค์ประกอบข้อมูล"

ตัวอย่างองค์ประกอบข้อมูลนี้ จะมีหน้าที่โดยตรงในการใช้งานต่างๆ ต้องได้รับการจัดการจากฐานข้อมูล ตัวอย่างขององค์ประกอบข้อมูล เช่น วันที่ เดือน ปี หมายเลขประจำตัว ชื่อ นามสกุล ที่อยู่ หมายเลขโทรศัพท์ เป็นต้น ความสัมพันธ์ระหว่างเนื้อที่ขององค์ประกอบข้อมูลและวัตถุประสงค์ของการใช้งานนั้น ถือว่าเป็นเรื่องสำคัญอย่างหนึ่งองค์ประกอบข้อมูลจะถูกบันทึกลงในฐานข้อมูล เนื่องจากมันมีความเกี่ยวข้องกับหน้าที่ซึ่งอยู่ภายใต้การดูแลของระบบนั้นๆ ยกตัวอย่างเช่น ในระบบงานค่าจ้างค่าแรงจะถือว่าเงินเดือนของพนักงานคนหนึ่งๆ เป็นองค์ประกอบข้อมูลระบบหรืองานในระบบคลังสินค้าจะมีองค์ประกอบข้อมูลเป็นประเภทหัวข้อของสินค้า เป็นต้น เพราะฉะนั้นจึงพบอยู่เสมอว่าในระบบงานที่ต่างชนิดกันจะมีองค์ประกอบข้อมูลที่แตกต่างกันไป

2.1.2.1 พจนานุกรมขององค์ประกอบข้อมูล (Data Element Dictionary)

ในการออกแบบฐานข้อมูลนั้นบ่อยครั้งที่จะเริ่มต้นด้วยการจัดทำรายชื่อ (list) ขององค์ประกอบข้อมูลที่คิดว่าระบบนั้นๆ ต้องมี ฉะนั้นไม่ว่าจะเริ่มต้นด้วยการจัดทำรายชื่อนี้หรือไม่อย่างน้อยที่สุดก่อนที่จะเริ่มลงมือทำงานคุณก็ควรสร้างมันขึ้นมา ด้วยเหตุผลที่ว่ามันจะมีส่วนช่วยให้สามารถเห็นถึงรายละเอียดขององค์ประกอบข้อมูลได้อย่างชัดเจนและสะดวกยิ่งขึ้น ในตารางที่จะกล่าวต่อไปนี้จะ เป็นตัวอย่างหนึ่งของรายชื่อขององค์ประกอบข้อมูลสำหรับระบบงานเล็กๆ เกี่ยวกับบุคลากรระบบหนึ่ง ในตัวอย่างนี้จะแสดงพื้นฐานของการออกแบบฐานข้อมูลนั่นคือ

"พจนานุกรม ขององค์ประกอบข้อมูล"

ชื่อองค์ประกอบข้อมูล	ชนิดข้อมูล	ความยาว
employee_no	numeric	5
employee_name	alphanumeric	25
employee_hired	data	6
salary	currency	10
dept_no	numeric	5
dept_name	alphanumeric	25
proj_no	numeric	5
proj_name	alphanumeric	25

ตารางที่ 2.1 ตารางพจนานุกรมขององค์ประกอบข้อมูล

พจนานุกรมขององค์ประกอบข้อมูลถือได้ว่าเป็นศูนย์กลางของการประยุกต์ใช้งานของระบบจัดการฐานข้อมูลนั้น คุณต้องกำหนดหัวข้อของข้อมูล (data item) ทั้งหมดที่จะปรากฏบนส่วนแสดงผลทางจอภาพบนรายงานต่างๆ ตลอดจนเพิ่มข้อมูลของพจนานุกรมขององค์ประกอบข้อมูลโดยข้อยกเว้นในส่วนที่ไม่ต้องกำหนดก็คือในส่วนรายละเอียดของสารสนเทศที่ใช้บนจอภาพและหัวข้อเรื่อง (title) รวมไปถึงจนถึงส่วนที่แสดงการพร้อมรับข้อมูล ไม่จำเป็นที่จะต้องกำหนดหรือนิยามองค์ประกอบของข้อมูลสำหรับสารสนเทศส่วนนี้แม้ว่าระบบจัดการฐานข้อมูลบางระบบไม่ค่อยแยกความแตกต่างของสารสนเทศออกจากองค์ประกอบข้อมูลที่แท้จริงให้เห็นอย่างเด่นชัด

นิยาม

พจนานุกรมขององค์ประกอบข้อมูลคือตารางขององค์ประกอบข้อมูล ซึ่งอย่างน้อยที่สุดต้องประกอบด้วย

1. ชื่อขององค์ประกอบข้อมูล
2. ชนิดของข้อมูล
3. ความยาวของแต่ละองค์ประกอบข้อมูลที่กำหนดไว้ในชื่อ

2.1.2.2 ชนิดขององค์ประกอบข้อมูล

ในส่วนขององค์ประกอบข้อมูลนั้นจะแสดงเพียงชื่อขององค์ประกอบข้อมูลชนิดและความยาวขององค์ประกอบต่างๆ ดังที่ได้กล่าวไว้แล้วในส่วนนิยามข้างต้นนั้น ข้อมูลเหล่านี้อาจเป็นวันเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เดือน ปี ระบบเงิน ตัวเลขหรืออักขระเป็นต้น แล้วแต่ประเภทของงานนั้นๆ จาก ตัวอย่างที่ยกมาในข้างต้นเป็นเพียงส่วนหนึ่งที่มีใช้ในระบบฐานข้อมูลอื่นๆ แม้ว่าจะเป็นเพียงส่วนเล็กๆ แต่ก็คงพอที่จะช่วยให้มองเห็นภาพรวมขององค์ประกอบข้อมูลที่ใช้กันในวงการธุรกิจ นอกจากนี้แล้วก็ยังมีตัวอย่างประเภทอื่นๆ ขององค์ประกอบข้อมูลอีก เช่น หมายเลขโทรศัพท์ ชื่อตำบล ชื่ออำเภอ ชื่อจังหวัด รหัสไปรษณีย์ หมายเลขการประกันสังคมและอื่นๆ ซึ่งจะพบว่าความต้องการประเภทต่างๆ ขององค์ประกอบข้อมูลนี้จะเปลี่ยนแปลงไปตามระบบแต่ละประเภท ในระบบจัดการฐานข้อมูลของเครื่องคอมพิวเตอร์ขนาดใหญ่ที่นิยมใช้กันมากนั้น มักจะอนุญาตให้มีการกำหนดองค์ประกอบข้อมูลเพียง 2 ประเภท นั่นคือ numeric (คือเป็นลักษณะของข้อมูลที่เป็นตัวเลขล้วนๆ) หรือ alphanumeric (คือรวมตัวอักขระพิเศษ เช่น เครื่องหมายต่างๆ) นั้น โดยไม่ยอมให้มีชนิดขององค์ประกอบข้อมูลแบบอื่นเลย บ่อยครั้งที่พบว่าประเภทขององค์ประกอบที่มีให้ใช้นั้นไม่พอเพียงที่จะใช้อธิบายลักษณะของข้อมูล

หมายเลขโทรศัพท์	วันที่ เดือน ปี
(022)555-1333	02/11/94
PE 5-500	6 Jan,94
Dial 'M' for Mike	The Ides of March

จะเห็นว่าไม่สามารถกำหนดลงไปได้แน่นอนว่าหมายเลขโทรศัพท์หรือวันที่ เดือน ปี จะเป็นองค์ประกอบข้อมูลประเภทไหน นั่นคือ ข้อมูลของหมายเลขโทรศัพท์ที่แสดงไว้ นั้นมีทั้งตัวเลข ตัวอักษรรวมทั้งตัวอักขระที่เป็นเครื่องหมาย ได้แก่ วงเล็บ เครื่องหมายลบและเครื่องหมายคำพูด ส่วนข้อมูลที่เป็นวันที่นั้นยังมีเครื่องหมายหารและจุดภาคเพิ่มมาอีกในส่วนนี้สำหรับผู้ที่ทำการออกแบบฐานข้อมูลแล้วถือว่าเป็นเรื่องที่ต้องการการพิจารณาอย่างรอบคอบเรื่องหนึ่งซึ่งการหนดแบบใดนั้นก็ขึ้นอยู่กับความคิการพิจารณาและประสบการณ์ของผู้ออกแบบฐานข้อมูลของแต่ละคน กล่าวโดยสรุปคือไม่มีกฎเกณฑ์ที่แน่นอนนั่นเอง

2.1.2.3 ชื่อขององค์ประกอบข้อมูล (Data Element Name)

ชื่อขององค์ประกอบข้อมูลภายในองค์ประกอบข้อมูลนั้น จะถูกใช้โดยตัวโปรแกรมของงานฐานข้อมูลและผู้ใช้ในการชี้และแยกประเภทขององค์ประกอบข้อมูล ชื่อเหล่านี้จะได้รับการคัดเลือกเพื่อนำมาใช้ในการแสดงหน้าที่ตลอดจนวัตถุประสงค์ขององค์ประกอบข้อมูลนั้นๆ สำหรับระบบจัดการฐานข้อมูลที่สร้างขึ้นโดยอาศัยชุดคำสั่งภาษาซีในบพนี้ นั้นเราจะใช้ชื่อขององค์ประกอบข้อมูลที่มองเห็น โดยผู้ใช้และ โปรแกรมเมอร์เป็นชื่อเดียวกันและชื่อนี้จะต้องอ่านแล้วเข้าใจจากมนุษย์และเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.4 การแทนค่าของข้อมูล (Data Value Representation)

องค์ประกอบข้อมูลภายในฐานข้อมูลนั้น จะใช้ชนิดของข้อมูลของคอมพิวเตอร์หรือภาษา (programming language) เพื่อทำงานต่างๆ อาทิ การจองเนื้อที่ว่างการจองเวลาเป็นต้น ฉะนั้นในไฟล์หนึ่งๆ จึงอาจพบชนิดของข้อมูลได้หลายๆ ชนิดไม่ว่าจะเป็นเลขจำนวนเต็มฐานสอง (binary integer) เลขทศนิยม (floating point number) สตริง (strings) และ bit flags ในส่วนที่เกี่ยวกับคลังข้อมูล (data storage) นั้นค่าขององค์ประกอบข้อมูลทั้งหมดจะถูกเก็บในรูปแบบของ "null-terminated strings" ด้วยการเก็บข้อมูลในรูปแบบนี้มีข้อดี คือข้อมูลสามารถถูกมองจากยูทิลิตี้ (utility) ของระบบซึ่งอาจแสดงไฟล์ข้อมูลบนจอแสดงผลได้ นอกจากนี้การใช้สตริงของตัวอักษรที่เก็บไว้ด้วยรหัสแอสกี (ASCII character strings) จะอนุญาตให้รูปแบบของข้อมูลและตัวโปรแกรมที่ทำการประมวลผลมันอยู่สามารถเคลื่อนย้าย (transfer) ระหว่างเครื่องคอมพิวเตอร์ได้ แม้ว่าจะอยู่ในรูปแบบฟอร์แมตที่แตกต่างกัน โดยไม่จำเป็นต้องกังวลว่าตัวแปลภาษา (compiler) หรือระบบปฏิบัติการ (operating system) จะจัดการกับรูปแบบหรือฟอร์แมตในระดับบิต (bit pattern) เช่น เครื่องหมายแสดงการสิ้นสุดของแฟ้มข้อมูลอย่างไม่เหมาะสม

2.1.3 แฟ้ม

ฐานข้อมูลหนึ่งๆ นั้นจะประกอบไปด้วยแฟ้มข้อมูลที่มีความสัมพันธ์เกี่ยวเนื่องกัน โดยมีจุดมุ่งหมายที่เหมือนกันอย่างใดอย่างหนึ่งซึ่งที่ได้กล่าวมาแล้ว ในนิยามของฐานข้อมูลข้างต้นมีข้อควรระวังอย่างหนึ่ง คืออย่าสับสนระหว่างคำว่า แฟ้ม (file) กับคำว่าระเบียบ (record) แฟ้มเป็นการรวมตัวของระเบียบ ระเบียบต่างๆ จะมีรูปแบบที่เหมือนกัน เพียงแตกต่างกันตรงที่แต่ละระเบียบจะมีเนื้อหารายละเอียดที่เป็นเอกลักษณ์เฉพาะตัวไม่เหมือนใคร ฉะนั้นระเบียบต่างๆ ภายในแฟ้มต่างๆ ก็มีส่วนประกอบข้อมูลชุดเดียวกันแต่ค่าขององค์ประกอบข้อมูลจะแตกต่างกันออกไปนั่นเอง

อาจกล่าวได้ว่าสามารถหารูปแบบของระเบียบใดระเบียบหนึ่งได้จากรายชื่อ (list) ขององค์ประกอบข้อมูลซึ่งปรากฏอยู่ในไฟล์

ตามโครงสร้าง (organization) ของไฟล์ จะจัดเตรียมเนื้อหาของข้อมูล (โดยแยกตามหน้าที่รับผิดชอบ) ที่เกี่ยวข้องกับจุดประสงค์ของระบบงานซึ่งฐานข้อมูลดูแลอยู่ความสัมพันธ์ภายในไฟล์ต่างๆ ขึ้นอยู่กับความสัมพันธ์ตามหน้าที่ของจุดประสงค์ต่างๆ ของระบบงานในการอธิบายลักษณะของไฟล์นั้นจะใช้วิธีการตั้งชื่อและรายการต่างๆ ขององค์ประกอบข้อมูล โดยการตั้งชื่อไฟล์ควรจะต้องตั้งชื่อให้มีความหมายสัมพันธ์กับจุดประสงค์ของไฟล์นั้นๆ ส่วนชื่อขององค์ประกอบข้อมูลก็ควรตั้งชื่อในลักษณะเกี่ยวกับการตั้งชื่อของไฟล์ด้วย เช่น ในการออกแบบไฟล์ของลูกจ้าง (employees)

ก็ควรตั้งชื่อในทำนอง "EMPLOYEE" หรือ "WORKER" หรืออื่นๆ ที่ให้ความหมายสื่อไปในลักษณะของพนักงาน พนักงาน ลูกจ้าง เป็นต้น ขอให้ดูตัวอย่างที่จะกล่าวถึงต่อไปนี้ซึ่งแสดงถึงวิธีการในการบรรยายลักษณะของไฟล์ลูกจ้าง

EMPLOYEES:

employee_no, employee_name, date_hire, salary

ในการอ้างถึงองค์ประกอบข้อมูลเพื่อหาความยาวขององค์ประกอบข้อมูลนั้นถ้าหากผลรวมของความยาวขององค์ประกอบต่างๆ ภายใน EMPLOYEES เท่ากับ 46 ไบต์ ในภาษาซีการบอกการสิ้นสุดของสตริงทำได้โดยการเพิ่ม "null value byte" ที่ท้ายสุดของสตริง ฉะนั้นจึงจำเป็นต้องเพิ่มอักขระอีกหนึ่งตัวให้กับความยาวของแต่ละองค์ประกอบข้อมูลนั้นคือ ผลรวมของความยาวจะเป็น 47 ไบต์

2.2 การออกแบบฐานข้อมูล

2.2.1 ขั้นตอนในการออกแบบฐานข้อมูล

- 2.2.1.1. พิจารณาความต้องการขั้นพื้นฐานของข้อมูล
- 2.2.1.2. กำหนดหน้าที่และความสามารถของฐานข้อมูลที่เราต้องการ
- 2.2.1.3. สร้างหัวเรื่องของข้อมูล
- 2.2.1.4. จำแนกองค์ประกอบข้อมูลออกจากแฟ้มข้อมูล
- 2.2.1.5. สร้างพจนานุกรมขององค์ประกอบข้อมูล
- 2.2.1.6. รวบรวมองค์ประกอบข้อมูลลงสู่แฟ้มข้อมูล
- 2.2.1.7. พิจารณาลักษณะเฉพาะของแต่ละไฟล์
- 2.2.1.8. พิจารณาความสัมพันธ์ระหว่างไฟล์
- 2.2.1.9. พิจารณา schema สำหรับระบบจัดการฐานข้อมูล (DBMS) ที่เลือกใช้

2.2.2 พิจารณาความต้องการขั้นพื้นฐานของฐานข้อมูล

เมื่อต้องการจะออกแบบฐานข้อมูลสักชิ้นหนึ่งแน่นอนว่าต้องมีแผนงานและเป้าหมายผู้ใช้บางคนอาจต้องการให้ผู้เขียนโปรแกรมทำในบางสิ่งบางอย่างตรงนี้เองที่คำถามต่างๆ ทำให้เกิดปัญหาเข้ามาไม่ว่าความต้องการนั้น จะถูกตอบสนองหรือทำขึ้นได้อย่างไร ผู้เขียนโปรแกรมรู้และเข้าใจปัญหามากน้อยเพียงไร ปัญหานี้ควรจะได้รับการแก้ไขขณะนั้นเลยหรือไม่ คำตอบของปัญหาที่มีอยู่ในขณะนั้นสามารถจัดการกับปัญหาได้จริงหรือไม่ มีใครบางคนในทีมงานที่เคยพบหรือแก้ไขปัญหามิชอบกันนี้ในสิ่งแวดล้อมอื่นๆ หรือไม่เพราะผู้เขียนโปรแกรมต้องการทำการกำหนด

ความต้องการทางด้านหน้าที่และความสามารถในการปฏิบัติงานให้กับฐานข้อมูล ดังนั้นก่อนที่จะลงมือทำงาน เพื่อสนองตอบความต้องการเหล่านี้จึงควรจะทำความเข้าใจกับมันเสียก่อนโดยเริ่มจากการพิจารณาความต้องการขั้นพื้นฐานของข้อมูล

2.2.3 กำหนดความต้องการเกี่ยวกับหน้าที่และความสามารถของฐานข้อมูล

ความต้องการทางด้านหน้าที่ของฐานข้อมูลจะเป็นตัวบอกริชาของข้อมูลที่จะเก็บลงในฐานข้อมูล เพื่อให้สนองตอบต่อความต้องการในส่วนนี้ควรทำเอกสารเกี่ยวกับทุกสิ่งๆที่รู้ในเรื่องของหน้าที่ของระบบงาน โดยเฉพาะข้อมูลทั้งหมดที่ฐานข้อมูลจะต้องรู้ส่วนความต้องการทางด้านความสามารถของฐานข้อมูลนั้น จะเป็นตัวบอกความถี่ ความเร็ว ปริมาณ และขนาด นั่นคือ บอกให้รู้ว่าบ่อยครั้งเพียงใด ความเร็วเท่าไร ปริมาณมากน้อยเพียงไรและขนาดของข้อมูลใหญ่แค่ไหนที่ฐานข้อมูลต้องเป็นตัวจัดการ

ข้อความเกี่ยวกับความต้องการของฐานข้อมูลนั้น ควรจะชัดเจนและไม่คลุมเคลือควรจะแยกเป็นหัวข้อ โดยแต่ละหัวข้อจะต้องมีความสมบูรณ์ในตัวของมันเองและใช้ถ้อยคำที่ผู้เข้าร่วมทั้งโปรแกรมเมอร์สามารถเข้าใจได้ง่ายโดยไม่สับสน

2.2.4 แสดงหัวข้อต่างๆ ของข้อมูล

เมื่อได้เก็บข้อมูลเกี่ยวกับความต้องการของฐานข้อมูลแล้วก็จะเริ่มแปลความต้องการเหล่านี้ไปสู่รูปแบบที่เหมาะสมต่อการทำงานในขั้นตอนนี้ให้หาวิธีบันทึกและจัดรูปแบบของหัวข้อข้อมูลที่ได้

ในการแสดงหัวข้อ (item) ของข้อมูลนั้น ต้องกลับไปสำรวจงานที่ได้มาไว้แล้วอย่างละเอียดถี่ถ้วนอีกครั้งหนึ่ง มองหาหัวข้อของข้อมูลที่จะเป็นไปได้ทั้งหมดจากต้นแบบขั้นพื้นฐานและความต้องการทั้งหมดของงานและบันทึกชื่อและรายละเอียดที่ทราบเกี่ยวกับหัวข้อนั้นเอาไว้วิธีหนึ่งที่ช่วยในการหาหัวข้อของข้อมูลคือ อาจเริ่มต้นด้วยการมองหาคำนามทั้งหมดเพราะเป็นไปได้ที่ในบรรดาคำนามเหล่านั้นอาจเป็นหัวข้อของข้อมูลที่ต้องการขอให้พิจารณาระบบงานบุคลากร (PERSONNEL) ที่กล่าวมาในข้างต้นซึ่งสามารถรวบรวมคำนามได้ดังนี้

employees, departments, salary, date hired,
 projects schedules, hours budget, hours worked,
 department number, department name, employee name,
 project number, project name, start date,
 completion date, assignments, hours expended

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายชื่อที่แสดงมาทั้งหมดนี้สามารถใช้เป็นขั้นแรกของการรวบรวมหัวข้อ (item) ที่ถูกบันทึกและจัดทำรายงานจากระบบฐานข้อมูลที่กำลังออกแบบนี้แน่นอนที่ว่าขณะที่กำลังดำเนินงานอยู่นั้น สามารถเพิ่มเติมหรือคัดออกเป็นบางส่วนจากรายชื่อนี้ได้

2.2.5 แยกองค์ประกอบข้อมูลออกจากเพิ่มข้อมูล

ถึงจุดนี้ลองหันมามองหัวข้อของข้อมูลทั้งหมดที่เก็บรวบรวมเอาไว้เพื่อพิจารณาว่าหัวข้อไหนที่เป็นองค์ประกอบข้อมูล โดยนำเอาหัวข้อทั้งหมดมารวมกันแล้วทำการเรียงลำดับใหม่และบันทึกไว้ในลักษณะที่สังเกตได้ง่าย ซึ่งจะสามารถสังเกตเห็นได้ชัดเจนขึ้นว่าหัวข้อใดเป็นองค์ประกอบข้อมูลและหัวข้อใดไม่ใช่ยกตัวอย่าง เช่น "date" จะเห็นได้ชัดเจนว่าเป็นองค์ประกอบข้อมูลในขณะที่ค่าของ department ไม่ใช่

ยังมีจุดที่ทำให้สับสนได้อีกจุดหนึ่งเช่น องค์ประกอบข้อมูลบางตัวสามารถแบ่งย่อยได้อีก เช่น "date" สามารถแบ่งย่อยเป็น ปี เดือน วัน อาจสงสัยว่าแล้วส่วนประกอบย่อยนั้นเป็นองค์ประกอบข้อมูลหรือไม่คำตอบคือไม่แน่เสมอไป ฉะนั้นในการกำหนดความต้องการนั้น จึงควรมีการบังคับว่าจะแยกย่อยลงไปได้อีกเพียงใด เพื่อไม่ให้เกิดความสับสนขึ้นแต่กรณีที่หลีกเลี่ยงไม่ได้ก็ควรแยกส่วนประกอบนั้นออกเป็นองค์ประกอบข้อมูลตัวใหม่เลย

จุดมุ่งหมายในส่วนนี้ก็คือต้องการที่จะแยกองค์ประกอบข้อมูลออกจากเพิ่มข้อมูล เพื่อให้สามารถทำงานในขั้นตอนต่อไปได้ คำนามจากรายชื่อที่รวบรวมมาทั้งหมดในหัวข้อที่แล้วนั้นจะถูกคัดออกมาเป็นองค์ประกอบข้อมูลดังต่อไปนี้

salary, date_hired, hours budgeted, hours worked,
department number, department name, employee name,
project number, project name, start date,
completion date, hours expended

2.2.6 สร้างพจนานุกรมขององค์ประกอบข้อมูล

ขอย้อนกลับไปดูหัวข้อที่ 2.1 ที่ได้กล่าวไว้ว่าการกำหนดพจนานุกรมขององค์ประกอบข้อมูลเป็นงานขั้นพื้นฐานอย่างหนึ่งของการออกแบบฐานข้อมูล เมื่อสามารถบอกได้ว่า มีอะไรบ้างที่เป็นองค์ประกอบข้อมูล เมื่อนั้นก็สามารถสร้างพจนานุกรมขององค์ประกอบข้อมูลได้

ในขั้นแรกต้องรวบรวมทุกสิ่งที่ทราบหรือที่สามารถบอกรายละเอียดเกี่ยวกับแต่ละองค์ประกอบข้อมูลหรืออย่างน้อยที่สุดก็ควรจะต้องรู้ว่าองค์ประกอบข้อมูลมีขนาดเท่าไรและมีชนิดของข้อมูล (data type) เป็นประเภทใด ถ้าการออกแบบนั้นเป็นการออกแบบที่มีพื้นฐานจากระบบที่มีอยู่

เคมเอกสารต่างๆ ของระบบเคมก็จะมีส่วนช่วยได้มากทีเดียวในการสร้างพจนานุกรมขององค์ประกอบข้อมูลชิ้นใหม่นี้ ถ้าเอกสารที่มีอยู่นั้นไม่สามารถให้ข้อมูลที่ต้องการได้เพียงพอแต่มีซอร์สโค้ด (source code) ของระบบเคม ก็คงต้องหาทางแปลงซอร์สโค้ดนั้นกลับให้มาอยู่ในรูปขององค์ประกอบข้อมูลให้ได้โดยการดูว่าซอฟต์แวร์นั้นทำงานอย่างไรในงานที่ทำอยู่นั้นเป็นระบบที่ทำด้วยมือ ก็จะได้ข้อมูลจากแบบฟอร์มต่างๆ เช่น บัตรตอกเวลาหรืออื่นๆ เพื่อที่จะหาว่าองค์ประกอบข้อมูลต่างๆ นั้นใช้งานอย่างไรบ้าง

จุดที่เน้นในที่นี้ คือต้องกำหนดลงไปอย่างชัดเจนเกี่ยวกับคุณสมบัติทางกายภาพขององค์ประกอบข้อมูลทั้งหมดภายในฐานข้อมูลนั้นคือต้องจัดการเกี่ยวกับกระบวนการของการกำหนดคุณสมบัติต่างๆ ให้มากที่สุดเท่าที่จะทำได้ก่อนที่จะเริ่มทำการออกแบบในขั้นตอนต่อไปจากนั้นจึงค่อยนำมาสร้างพจนานุกรมขององค์ประกอบข้อมูล

มีข้อเสนอแนะว่าควรจะหาใครสักคนหนึ่งในหมู่ผู้ใช้งานช่วยดูพจนานุกรมขององค์ประกอบข้อมูลที่สร้างขึ้น เพื่อให้ยอมรับงานชิ้นนี้หากไม่มีใครเลยที่เห็นด้วยกับงานที่ได้ก็จำเป็นอย่างไรที่จะต้องวิเคราะห์เพิ่มเติมก่อนที่จะลงมือออกแบบในขั้นต่อไป

2.2.7 การกำหนดลักษณะเฉพาะหรือวิธีการในการเรียกใช้ข้อมูลของแต่ละไฟล์

จำเป็นต้องมีการกำหนดวิธีการที่จะเรียกดูข้อมูลของแต่ละไฟล์ที่ต้องการซึ่งจะเห็นว่ามีความเลือกอยู่หลายทาง ในขั้นตอนการออกแบบนั้นจะต้องรวมทางเลือกเหล่านี้ไว้ทั้งหมด

จุดมุ่งหมายของการกำหนดวิธีการในการเรียกใช้ข้อมูลที่ต้องการจากฐานข้อมูลนั้น ก็เพื่อว่าจะได้สามารถตัดสินใจได้ว่าฟิลด์ใดที่ควรจะเป็นคีย์รอง (secondary key) และฟิลด์ใดบ้างที่ควรจะมีความสัมพันธ์กันที่จะกล่าวถึงต่อไปนี้เป็นประเภทต่างๆ ของการเรียกข้อมูลออกจากฐานข้อมูลซึ่งระบบควรจะสามารถทำงานได้

2.2.7.1 คีย์ที่ใช้ในการเรียกข้อมูล (Specific Key Retrieval)

สมมติว่าเมื่อทำงานอยู่กับคอมพิวเตอร์และรันโปรแกรมที่แสดงระเบียบข้อมูลที่ถูกเลือกไว้ก็ควรจะต้องทราบบางอย่างเกี่ยวกับคุณค่าของข้อมูลภายในระเบียบข้อมูลที่ต้องการเรียกดู

คำสั่งของการเรียกข้อมูลเราเรียกว่า "query" คำๆ นี้ถูกใช้เป็นคำมาตรฐานคำหนึ่งในการเรียกข้อมูลเฉพาะของการเรียกข้อมูลจะเป็นตัวกำหนดคีย์ขององค์ประกอบข้อมูลและถ้าระเบียบข้อมูลใดที่มีค่าตรงกับคำสั่งของการเรียกใช้ข้อมูลที่ป้อนเข้าไปก็จะได้ข้อมูลที่ต้องการค้นหาตัวอย่างเช่น

dept_no = 123

ถ้าใช้คำสั่งนี้เรียกดูข้อมูลจากไฟล์ EMPLOYEES ระเบียบที่มีค่าของรหัสแผนกเท่ากับ 123 ก็จะถูกดึงออกมา ลักษณะการใช้คำสั่งนี้จะเป็นมาตรฐานในการค้นหาข้อมูลที่ต้องการในบางกรณี อาจใช้องค์ประกอบข้อมูลอื่นในการเรียกดูข้อมูลก็ได้ เช่น ถ้าต้องการทราบข้อมูลต่างๆ ของคนที่ทราบเพียงชื่อคือ Michael ในกรณีนี้กระบวนการเรียกข้อมูลทั้งหมดของคนที่มีชื่อตรงกันนี้ออกมา โดยใช้ชื่อเป็นคีย์ในการค้นหา เป็นต้น

2.2.7.2 การค้นหาเป็นช่วงของข้อมูล (Key Range Searches)

การเรียกดูข้อมูลสามารถกำหนดขอบข่ายของการค้นหาได้ ตัวอย่างเช่น ถ้าต้องการรายงานเกี่ยวกับข้อมูลของพนักงานทั้งหมด ที่ได้รับเงินเดือนน้อยกว่า 8,000 บาท หรือข้อมูลของพนักงานทั้งหมดที่รับมาทำงานในเดือนนี้ เป็นต้น

ขอบข่ายของการค้นหาข้อมูลจะมีประโยชน์ที่ทำให้ได้รับข้อมูลครั้งละหลายๆ เพื่อให้สามารถคัดเลือกเอาเฉพาะส่วนที่ต้องการ โดยที่ไม่ต้องมาคอยคำสั่งเพื่อหาระเบียบที่ต้องการ

2.2.7.3 Boolean Query Retrieval

Boolean query เป็นลักษณะหนึ่งของการใช้ประโยชน์จากนิพจน์ทางตรรกศาสตร์โดยคำสั่งในการเรียกใช้ข้อมูลแต่ละคำสั่งนั้น จะประกอบไปด้วยเงื่อนไขเพียง 2 ทาง คือถูกหรือผิดนั้น รูปแบบของการใช้นิพจน์ของ Boolean query นั้น มี 3 ประเภท คือ AND, OR และ NOT

ลองพิจารณาเงื่อนไข 2 ชนิดที่ประกอบด้วย Boolean operator ต่อไปนี้

dept_no = 123 AND salary > 20,000

จากคำสั่งข้างต้น มันจะเรียกข้อมูลของพนักงานที่สังกัดแผนกที่มีรหัสแผนกเป็น 123 และพนักงานคนนั้นต้องได้รับเงินเดือนมากกว่า 20,000 บาท

2.2.8 บอกลักษณะความสัมพันธ์ระหว่างไฟล์

ฐานข้อมูลที่ใช้โมเดลเชิงสัมพันธ์นี้โดยปกติไฟล์ข้อมูลจะมีความสัมพันธ์กันเมื่อไฟล์เหล่านี้มีองค์ประกอบข้อมูลร่วมกันและองค์ประกอบข้อมูลนั้นเป็นคีย์ของไฟล์นั้นๆ ในฐานข้อมูล PERSONNEL นี้ไฟล์ EMPLOYEES จะมีความสัมพันธ์กับไฟล์ DEPARTMENTS เพราะระเบียบพนักงาน (employee record) รวมทั้งรหัสแผนก (department number) และชื่อแผนก (department name) จะเป็นคีย์หลักของไฟล์ DEPARTMENTS

2.3 ระบบจัดการฐานข้อมูล

ในหัวข้อที่ 2 และ 3 ได้กล่าวถึงฐานข้อมูล (data base) ลักษณะสถาปัตยกรรมตลอดจนเค้าโครงของฐานข้อมูลมาแล้วในหัวข้อนี้จะอธิบายระบบจัดการข้อมูล (data base management) ซึ่งเป็นซอฟต์แวร์ที่ใช้ในการจัดการลักษณะข้อมูล เมื่อต้องการพัฒนาซอฟต์แวร์ขึ้นสักชิ้นเพื่อมาใช้งาน ซึ่งคงไม่คิดที่จะต้องเริ่มทำงานตั้งแต่ต้น จะได้ไม่ต้องมาเสียเวลาไปกับการพัฒนาคำสั่งที่ทำการอ่านและเขียนระเบียบหรือคำสั่งที่ทำการเก็บรักษาแฟ้มดัชนี (index file) ตลอดจนการเก็บและดึงข้อมูลกลับคืนมากระบวนกรเหล่านี้เป็นความต้องการขั้นพื้นฐานซึ่งแทบทุกระบบต้องมี ฉะนั้นจึงไม่มีความจำเป็นเลยที่จะมาเสียเวลาในส่วนนี้ แต่ปัญหาที่คืออะไรที่จะทำงานเหล่านี้ได้ คำตอบคือ DBMS

2.3.1 DBMS

DBMS (Data Base Management System) หรือระบบจัดการฐานข้อมูลนี้ เป็นซอฟต์แวร์สำเร็จรูปซึ่งมีหน้าที่ ในการจัดการเกี่ยวกับการเก็บรักษาและการเรียกคืนข้อมูลภายในระบบฐานข้อมูล DBMS เป็นตัวกลางระหว่างโปรแกรมใช้งาน (application software) กับระบบฐานข้อมูล (data base system) มันจะทำงานและตอบสนองต่อการร้องขอ (request) ของซอฟต์แวร์ประยุกต์ (application software)



รูปที่ 2.1 ระบบฐานข้อมูล DBMS

เมื่อโปรแกรมใช้งานต้องการข้อมูลระเบียบหนึ่งจากฐานข้อมูล มันก็จะเรียก DBMS และผ่านค่าพารามิเตอร์ที่จะบอก DBMS ว่าจะเรียกข้อมูลระเบียบนั้นได้อย่างไร เช่น ให้หาระเบียนสินค้าในสต็อกหมายเลข 0123 DBMS ก็จะหาระเบียนที่ต้องการจากนั้นจะส่งสำเนา (copy) กลับมาให้โปรแกรมใช้งาน กรณีที่หากว่าโปรแกรมใช้งานมีระเบียบที่ต้องการบันทึกลงในฐานข้อมูลก็จะเรียกใช้ DBMS แล้วส่งสำเนาของระเบียบนั้นไปให้ตัว DBMS ก็จะเก็บระเบียบนั้นไว้ในตำแหน่งที่เหมาะสมซึ่งโปรแกรมใช้งานจะสามารถค้นหาได้พบในเวลาต่อมาส่วนดัชนีของระเบียบและความสัมพันธ์ใดๆ ที่มันมีต่อระเบียบอื่นภายในฐานข้อมูลนั้นจะถูกจัดการและเก็บรักษาไว้ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DBMS เช่นกัน ในส่วนของโปรแกรมใช้งานสิ่งที่มีนจำเป็นต้องการก็มีเพียงได้สร้างระเบียบข้อมูลใหม่ หรือได้เรียกข้อมูลระเบียบใดมาแก้ไข และส่งระเบียบนั้นกลับคืนสู่ฐานข้อมูลนั้น

2.3.2 หลักการพื้นฐานของระบบจัดการฐานข้อมูล

ถ้าต้องการสร้าง DBMS เองก็ควรทำความเข้าใจในพื้นฐานของเทคโนโลยีด้าน DBMS บ้างพอสมควร หลักการเหล่านี้จะเป็นพื้นฐานของระบบที่เราจะสร้างขึ้นในหัวข้อต่อไป

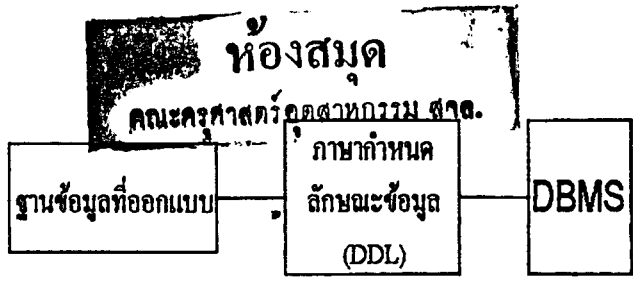
จากรายชื่อที่คงพอจะทราบว่า DBMS เป็นซอฟต์แวร์ที่ถูกใช้ในการจัดการกับฐานข้อมูล ฉะนั้นเพื่อให้ DBMS เป็นสื่อกลางระหว่างโปรแกรมใช้งาน (application program) ที่ต้องการรับระเบียบข้อมูลกับฐานข้อมูลที่รอการประมวลผลกล่าวคือ DBMS คอยผ่านข้อมูลไปมาระหว่างโปรแกรมและฐานข้อมูลนั่นเอง ในการที่ส่งผ่านสารสนเทศ (information) ของ DBMS นี้ จะต้องมีบางสิ่งที่เป็นตัวบอก DBMS ว่าข้อมูลภายในฐานข้อมูลนั้นมีลักษณะเป็นเช่นใดและสิ่งนั้นก็คือภาษากำหนดลักษณะข้อมูลหรือ Data Definition Language (DDL) ภาษากำหนดลักษณะข้อมูลนี้จะจัดเตรียมรายละเอียดของฐานข้อมูลให้อยู่ในรูปแบบที่ DBMS สามารถแปลความได้ นอกจากงานที่ DDL รับผิดชอบนี้ แล้ว DBMS ยังต้องทราบถึงวิธีการในการติดต่อกับโปรแกรมใช้งาน ในส่วนนี้ก็มีภาษาจัดการข้อมูล (data manipulation language) หรือ DML ขึ้นมาก็จะหลีกเลี่ยงไม่ได้ที่จะต้องพิจารณาและทำความเข้าใจกับ DDL และ DML ประกอบกันไป

2.3.2.1 ภาษากำหนดลักษณะข้อมูล

ในหัวข้อที่ 2.1 ซึ่งได้สร้างฐานข้อมูลส่วนบุคคล (personnel database) ขึ้นมาชุดหนึ่งโดยบ่งบอกถึงองค์ประกอบของข้อมูล (data element) เพิ่มข้อมูลตลอดจนความสัมพันธ์ระหว่างเพิ่มข้อมูล ในการออกแบบนั้นจะต้องสร้างตารางเพื่อแสดงลักษณะเฉพาะ (characteristic) ขององค์ประกอบข้อมูลและเพิ่มข้อมูลตารางที่กล่าวนี้จะเป็นตัวบอกถึงรูปแบบของฐานข้อมูลตลอดจนถึงภาษากำหนดลักษณะข้อมูลและเพื่อที่จะทำให้ตารางนี้มีประโยชน์ต่อ DBMS ตารางจะต้องอยู่ในรูปแบบที่คอมพิวเตอร์สามารถแปลงลงสู่ลักษณะทางกายภาพ (physical storage) ของคลังข้อมูลได้และหลังจากที่ได้วางรูปแบบของฐานข้อมูลแล้วก็จะเริ่มเขียน DDL

DDL เป็นภาษาเช่นเดียวกับภาษาอื่นๆ เช่น ภาษาซีและก็เช่นเดียวกับภาษาอื่นๆ ซึ่งแต่ละภาษาก็มีการแยกเป็นหลายประเภทหลายผลิตภัณฑ์ (เช่น เทอร์โบซี ไมโครซอฟซี เป็นต้น) ตัว DDL เองก็มีหลายประเภทขึ้นอยู่กับว่ามันใช้กับ DBMS ประเภทใด ฉะนั้นถ้าหากต้องการออกแบบฐานข้อมูลพร้อมด้วยตารางที่กล่าวไว้ในหัวข้อที่ 2.1 ก็ต้องออกแบบ DDL ให้เหมาะสมกับ DBMS ที่ใช้อยู่ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



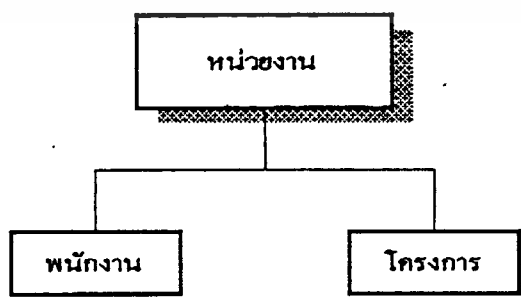
รูปที่ 2.2 โครงสร้างภาษากำหนดลักษณะข้อมูล

ปัจจุบันมีวิธีการมากมายในการเขียน DDL วิธีการที่ใช้กันมากที่สุดคือเขียน DDL เป็นชุดคำสั่งเก็บไว้ในเท็กไฟล์ (text file) แล้วค่อยนำมาสร้างเป็นซอร์สไฟล์ (source file) DBMS บางตัวใช้ตัวแปลภาษาของ DDL ทำการแปลเพิ่มข้อมูลของ DDL ให้เป็นตารางที่ DBMS สามารถเข้าใจได้ วิธีการอื่นก็เช่น ให้ DBMS แปล (interpret) DDL ด้วยตัวเองเป็นต้น ในหัวข้อที่ 2.5 จะกล่าวถึงคุณสมบัติของภาษาซีที่จะทำให้ขั้นตอนเหล่านี้ทำงานได้ หลังจากนั้นจะใช้ชุดคำสั่งแปลภาษาของ DDL ที่แปลเท็กไฟล์ของ DDL ให้เป็น DDL ในรูปแบบภาษาซี

2.3.2.2 รูปแบบของข้อมูล

ขณะที่ทำการออกแบบ DDL จะต้องระวางรูปแบบของข้อมูลที่ตัว DBMS รองรับอยู่ ประเภทของรูปแบบข้อมูล (data model) แบ่งได้เป็น 3 รูปแบบ ได้แก่ รูปแบบตามลำดับชั้น (hierarchical model) รูปแบบเครือข่าย (network model) และรูปแบบความสัมพันธ์ (rerational model)

ก. ฐานข้อมูลแบบลำดับชั้น (Hierarchical Data Base) จะมีความสัมพันธ์ระหว่างเพิ่มข้อมูลเป็นลำดับชั้นหรือตามอาวูโสเพิ่มข้อมูลจะมีตำแหน่งจากบนลงล่าง โดยเพิ่มข้อมูลที่อยู่ในระดับสูงกว่าจะเป็นแม่ของเพิ่มข้อมูลในระดับที่ต่ำกว่าดังรูปต่อไปนี้ ซึ่งได้แสดงลำดับชั้นของพนักงานหน่วยงานหนึ่งไว้เป็นตัวอย่างแล้ว

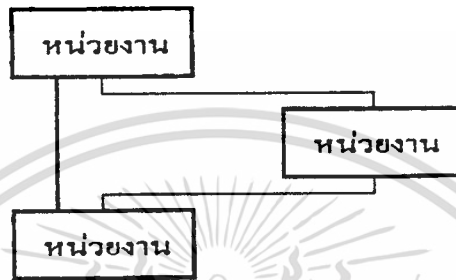


รูปที่ 2.3 ฐานข้อมูลแบบลำดับชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อสังเกตว่าในลำดับชั้นหนึ่งๆ เพิ่มข้อมูลหนึ่งเพิ่มจะมีเพิ่มข้อมูลลูก (child file) ได้มากมาย ในขณะที่เพิ่มข้อมูลลูกจะมีเพิ่มข้อมูลแม่ (parent file) ได้เพียงเพิ่มเดียว

ข. **ฐานข้อมูลแบบเครือข่าย (Network Data Base)** จะมีลักษณะคล้ายกับฐานข้อมูลแบบลำดับชั้นแต่มีข้อแตกต่างตรงที่ฐานข้อมูลแบบเครือข่าย สามารถมีเพิ่มข้อมูลแม่ได้มากกว่าหนึ่งเพิ่มดังรูปต่อไปนี้



รูปที่ 2.4 ฐานข้อมูลแบบเครือข่าย

2.4 ภาษาซีกับภาษากำหนดลักษณะข้อมูล

หัวข้อที่ 2.1 และหัวข้อที่ 2.2 ได้กล่าวถึงเรื่องฐานข้อมูลและการออกแบบไปแล้วหัวข้อที่ 2.3 ก็ได้กล่าวถึงระบบจัดการฐานข้อมูล หัวข้อนี้จะเป็นเรื่องของระบบจัดการฐานข้อมูลสำหรับโปรแกรมเมอร์ภาษาซีบนคอมพิวเตอร์ใช้คุณสมบัติของภาษาซี ในการสร้างภาษา 2 ภาษา คือภาษากำหนดลักษณะข้อมูล (data definition language) และภาษาจัดการข้อมูล (data manipulation language) ภาษาทั้งสองได้รวมเอา 3 ส่วนประกอบของสิ่งที่เกี่ยวข้องกับฐานข้อมูลไว้ด้วยกัน

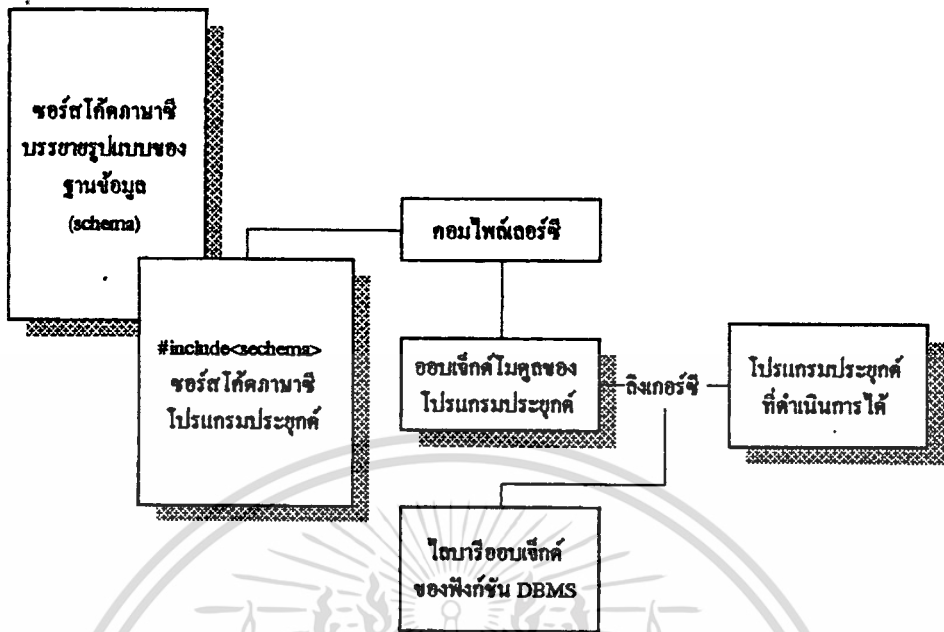
2.4.1 แบบแผนฐานข้อมูล (schema)

2.4.2 โปรแกรมประยุกต์ (application program)

2.4.3 ระบบจัดการฐานข้อมูล (DBMS)

ขั้นตอนเริ่มจากอธิบายแบบแผนฐานข้อมูลไว้ในโมดูลหนึ่งอาจเก็บไว้เป็นเท็กซ์ไฟล์ เพื่อที่โปรแกรมประยุกต์ก็จะรวมมันเอาไว้เป็นส่วนหนึ่งของโปรแกรมจากนั้นคอมไพล์ไปเป็นออบเจกต์โมดูล ส่วนระบบจัดการฐานข้อมูลแยกคอมไพล์ไว้อีกโมดูลหนึ่งแล้วเชื่อมโยงทั้งสอง โมดูลเข้าด้วยกันระบบจัดการฐานข้อมูลจะอ้างอิงถึงแบบแผนฐานข้อมูลใดๆ อย่างกว้างๆ เมื่อเชื่อมโยงโปรแกรมประยุกต์กับระบบจัดการฐานข้อมูลเข้าด้วยกันแล้วแบบแผนข้อมูลที่ถูกอ้างอิงถึงจะถูกแทนที่ด้วยแบบแผนฐานข้อมูลที่เราเชื่อมโยงเข้าไปด้วยวิธีการแบบนี้ แบบแผนฐานข้อมูลและระบบจัดการฐานข้อมูลก็จะเป็นส่วนหนึ่งของโปรแกรมประยุกต์ในรูปแบบที่สามารถประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 รูปแบบแผนฐานข้อมูล

การใช้วิธีดังกล่าว โปรแกรมผลลัพธ์ที่ออกมาจะเป็นโปรแกรมที่ประมวลผลได้ในตัวเองมีส่วนที่จำเป็นสำหรับการจัดการฐานข้อมูลในตัวเอง ได้แก่ แบบแผนฐานข้อมูลซึ่งเป็นเซตของอาร์เรย์, ระบบจัดการฐานข้อมูลเป็นไลบรารีฟังก์ชันภาษาซี (ส่วนหลักในการจัดการฐานข้อมูลและโปรแกรมประยุกต์ใดๆ) และโปรแกรมประยุกต์ ซึ่งแต่ละโปรแกรมก็มีจุดมุ่งหมายในการใช้งานเฉพาะอย่างไป เมื่อนำทั้งสามส่วนนี้มารวมกันก็จะเป็นโปรแกรมที่ประมวลผลได้หนึ่งโปรแกรม เทคนิคที่กล่าวนี้ ต้องใช้ภาษาในโปรแกรมที่มีคุณสมบัติเหมาะสมแล้วจะพบว่าภาษามีคุณสมบัติดังกล่าว

ภาษาโปรแกรมที่ใช้ต้องยอมให้มีการกำหนดค่าให้ชื่อใดๆ มีค่าเป็นตัวเลขการจะอธิบายแบบแผนฐานข้อมูลในรูปแบบของข้อมูลได้ต้องสามารถกำหนดชื่อไฟล์และชื่อองค์ประกอบข้อมูลได้เมื่อจำนวนของชื่อเพิ่มขึ้นหรือลดลง ซอฟต์แวร์ก็ควรที่จะจัดการได้โดยไม่ต้องยุ่งยากนักชื่อที่ตั้งเอาไว้ใช้ในการอ้างถึงไฟล์ ก็สามารถใช้ในการเข้าถึงตำแหน่งในตารางขององค์ประกอบข้อมูลได้ด้วย เช่นเดียวกับชื่อขององค์ประกอบข้อมูลก็ใช้ในการเข้าถึงตำแหน่งในตารางของของชนิดและความยาวของมันได้ด้วย จากหัวข้อที่ 2.2 และหัวข้อที่ 2.4 ได้แสดงฐานข้อมูลที่อยู่ในรายการของไฟล์และแสดงไฟล์ในรูปแบบรายการขององค์ประกอบข้อมูล รายการนี้จะง่ายขึ้นถ้าอยู่ในรูปของตัวเลขจำนวนเต็ม (integer) แต่ถ้าใช้ในรูปแบบของสตริงจะทำให้เกิดการสับสนเปลืองที่หน่วยความจำ

ในการเข้าถึงตารางของแบบแผนฐานข้อมูลและกินเนื้อที่หน่วยความจำสำหรับสตริง นอกจากนี้ยังต้องใช้เวลามากในการค้นหาสตริงที่ต้องการอีก ในระบบจัดการฐานข้อมูลส่วนใหญ่จะแปลงชื่อให้อยู่ในรูปของตัวเลขจำนวนเต็มในการเข้าถึงตาราง ดังนั้นในที่นี้จะใช้เลขจำนวนเต็มแทนชื่อเพื่อความสะดวกและรวดเร็ว

ภาษาซี มีมาโครพรีโพรเซสเซอร์ #define สามารถกำหนดชื่อไฟล์ให้เป็นเลขจำนวนเต็มได้เช่น

```
#define PROJECTS 0
```

```
#define EMPLOYEES 1
```

```
#define DEPARTMENTS 2
```

และกำหนดชื่อองค์ประกอบข้อมูล ให้เป็นเลขจำนวนเต็มได้เช่น

```
#define EMPLOYEE_NO 1
```

```
#define EMPLOYEE_NAME 2
```

```
#define DATE_HIRED 3
```

โดยการใส่มาโครพรีโพรเซสเซอร์ #define ของภาษาซี กำหนดชื่อของไฟล์และองค์ประกอบข้อมูลเป็นเลขจำนวนเต็มสามารถที่จะกำหนด ชื่อไฟล์ให้อยู่ในอาร์เรย์ของเลขจำนวนเต็ม ซึ่งมีประโยชน์ต่อการใช้งานของทั้ง โปรแกรมเมอร์และเครื่อง คือภาษาค้นกำเนิดจะมีชื่อไฟล์และชื่อองค์ประกอบข้อมูลที่มีความหมายตามที่โปรแกรมเมอร์ต้องการ ขณะที่ซอฟต์แวร์ก็ใช้ชื่อเหล่านี้ ในรูปของเลขจำนวนเต็มระหว่างทำกระบวนการต่างๆ วิธีนี้ทำให้สามารถส่งผ่านชื่อไฟล์หรือองค์ประกอบข้อมูลให้ฟังก์ชันที่ต้องการชื่อไฟล์และชื่อองค์ประกอบข้อมูลในรูปของจำนวนเต็มด้วยไลบรารีฟังก์ชันที่จัดการเก็บและคืนเรคอร์ดของฐานข้อมูลก็สามารถใช้ค่านี้เป็นดัชนีต่าง ในการเข้าถึงตารางของแบบแผนฐานข้อมูลได้

ภาษาซี เป็นภาษาที่เชื่อมโยงโมดูลหลายๆ โมดูลที่คอมไพล์อย่างอิสระต่อกันเข้าด้วยกัน เป็น 1 โมดูลได้ ระบบจัดการฐานข้อมูลควรถูกคอมไพล์ครั้งเดียว เพื่อใช้เชื่อมโยงกับโปรแกรมประยุกต์อื่นๆ ต่อไป ดังนั้นภาษาซีสนับสนุนความต้องการข้อนี้

ภาษาโปรแกรม ที่ใช้ต้องมีนัยสำคัญของชื่อตัวแปรที่เพียงพอต่อความยาวของชื่อไฟล์และชื่อองค์ประกอบข้อมูล นัยสำคัญของชื่อตัวแปรควรเพียงพอต่อการที่จะใช้งานในการทำไตเติลหรือ query แต่เดิมภาษาซียอมให้ชื่อตัวแปรมีนัยสำคัญแค่ 8 ตำแหน่ง ไม่เพียงพอต่องานที่จะใช้แต่ปัจจุบันได้ถึง 31 ตำแหน่ง

คอมไพเลอร์ภาษาซี แต่ก่อนไม่ยอมให้สมาชิกต่างโครงสร้างกันมีชื่อเดียวกัน ยกเว้นแต่เมื่อเทียบกันแล้วจะมีตำแหน่งตรงกันและต้องมีชนิดเดียวกันด้วย แต่ปัจจุบันตัวภาษาได้ยกเลิกข้อ

จำกัดข้อนี้และคอมไพล์ส่วนใหญ่ก็ปรับปรุงให้เป็นไปตามข้อกำหนดใหม่ด้วย ต่อไปจะเห็นว่าจำเป็นต้องนำคุณสมบัติของภาษาซีข้อนี้มาใช้

ภาษาที่ใช้ต้องมีตัวแปรแบบ external มาใช้ด้วยและไม่ควรเป็นภาษาที่จำกัดชนิดของตัวแปรและขนาดของอาร์เรย์ตายตัว เช่น ถ้าฟังก์ชันที่มีพารามิเตอร์ที่ประกาศไว้เป็นชนิดหนึ่งการเรียกใช้ฟังก์ชัน ก็ควรยืดหยุ่นยอมให้อาร์กิวเมนต์มีชนิดอื่นได้และจากขนาดของอาร์เรย์ที่ประกาศเอาไว้ ก็ควรยอมให้ใช้ดัชนีต่างนอกเขตที่ประกาศไว้

ฟังก์ชันในภาษาซี สามารถส่งผ่านแอดเดรสของโครงสร้างของข้อมูลไปยังอีกฟังก์ชันหนึ่งได้ฟังก์ชันที่ถูกเรียกก็จะมองเป็นแอดเดรสบางสิ่งบางอย่างที่ไม่รู้ชนิดและขอบเขต ซึ่งฟังก์ชันที่ถูกเรียกก็สามารถทำงานได้ถูกต้องจะนำคุณสมบัติข้อนี้มาใช้ประโยชน์ต่อไป นอกจากนี้อาร์เรย์ในภาษาซีที่ประกาศไว้ต้องมีขนาดกำหนดด้วย เพราะโมดูลอื่นที่จะใช้อาร์เรย์นี้จะไม่รู้ขนาดเลย ตัวอย่างเช่น

```

/* source file A*/
int i [ ] = {1,2,3,4,0};
main( )
{
    foo( );
}
/* source file B */
extern int i [ ];
foo( )
{
    int *j = i ;
    while (*j)
        printf( "\n%d", *j++);
}

```

อาร์เรย์ของเลขจำนวนเต็ม i ในไฟล์ A ประกาศเอาไว้ในแบบมี 5 องค์แรกกอบในไฟล์ B จะรู้ว่าเป็นอาร์เรย์ของเลขจำนวนเต็ม โค้ดที่เรียกใช้อาร์เรย์ของเลขจำนวนเต็มนี้ ก็มีวิธีที่จะรู้ความยาวของอาร์เรย์ได้จากตัวอย่างดังกล่าวจะเห็นว่าทำให้ค่าอาร์เรย์ตัวสุดท้ายเป็นศูนย์ ก็เพื่อให้รู้ว่าอาร์เรย์นี้จบลงที่ใด ถ้ามีการเปลี่ยนจำนวนองค์ประกอบของอาร์เรย์ในไฟล์ A ก็ไม่ต้องคอมไพล์

ไฟล์ B ใหม่หรือทำอะไรทั้งสิ้นเพียงแค่คอมไพล์ไฟล์ A แล้วเชื่อมโยงกับออบเจ็กต์โมดูลของ B นั้น

เมื่อคุณใช้คุณสมบัติของภาษาซีอีกแบบโดยการประกาศตัวแปรเป็นอาร์เรย์เฉพาะที่ (local) ในฟังก์ชัน main ฟังก์ชัน main จะส่งแอดเดรสของอาร์เรย์ไปให้ฟังก์ชัน foo ซึ่งจะรับค่าที่ส่งมาเป็นแบบตัวชี้ไปยังเลขจำนวนเต็มแทนที่จะเป็นอาร์เรย์ของเลขจำนวนเต็มเช่นเดียวกับตัวอย่างก่อน ฟังก์ชัน foo จะไม่รู้ขนาดของอาร์เรย์ รู้แต่ว่าเลขจำนวนเต็มตัวสุดท้ายที่จะใช้เป็นศูนย์

```
main( )
{
    static int I[ ]={1,2,3,4,0};
    foo( i );
}
foo( j )
int *j;
{
    while( *j )
        printf( "\n%d", *j++);
}
```

ข้อดี ของภาษาที่ได้กล่าวมามีความสำคัญต่อการเขียนโปรแกรมนำมาใช้ในการสร้างภาษา กำหนดลักษณะข้อมูลได้ เนื่องจากว่าสามารถกำหนดชื่อไฟล์เป็นเลขจำนวนเต็ม และระบุเป็น รายการขององค์ประกอบข้อมูลที่อยู่ในรูปของเลขจำนวนเต็ม

ฟังก์ชันของระบบจัดการฐานข้อมูลที่จัดการกับระเบียบ เมื่อได้รับชื่อไฟล์มา (อยู่ในรูปเลข จำนวนเต็ม) จะใช้ค่าของชื่อไฟล์เป็นดัชนีต่าง เพื่อเข้าถึงอาร์เรย์ขององค์ประกอบข้อมูล โดยที่ไม่รู้ ถึงความยาวของอาร์เรย์ ซึ่งก็หมายความว่าเราสามารถที่จะคอมไพล์ฟังก์ชันระบบจัดการฐานข้อมูล เพียงครั้งเดียวจากนั้นก็เชื่อมโยงเข้ากับโปรแกรมสำเร็จประยุกต์ต่างๆ โปรแกรมและหลายๆ แบบ แผนของฐานข้อมูลต่างๆ ได้โดยไม่ต้องเปลี่ยนแปลงโค้ดของระบบจัดการฐานข้อมูลใหม่

คุณสมบัติสุดท้ายของภาษาซีที่ต้องนำมาใช้ในที่นี้ก็คือ ความสามารถในการปฏิบัติการกับ ตัวชี้มากมายและความหลากหลายในการใช้ตัวชี้ เช่น ตัวชี้ของอาร์เรย์, ตัวชี้ของโครงสร้าง, ตัวชี้ ของตัวชี้, ตัวชี้ไปยังอาร์เรย์ของตัวชี้ ฯลฯ และความสัมพันธ์กันอย่างใกล้ชิดของตัวชี้กับอาร์เรย์

การปฏิบัติการกับตัวชี้ในภาษาซีเป็นจุดสำคัญในการสร้างภาษากำหนดลักษณะข้อมูล คุณสมบัติเหล่านี้เป็นส่วนยากที่สุดที่จะต้องใช้เงินจำนวนมาก

2.5 ข้อมูลภาษาซี

ตามที่ได้อธิบายไปแล้วว่าในหัวข้อที่ 2.4 อรรถประโยชน์ของภาษาซีสามารถถูกนำไปใช้อธิบายฐานข้อมูลแบบจำลองเชิงสัมพันธ์ได้ เทคนิคนี้และฟังก์ชันภาษาซีที่สนับสนุนจะนำมาใช้ในข้อมูลภาษาซีเป็นระบบจัดการฐานข้อมูลที่มีชื่อเต็มๆ ว่า Cheap Data Base Management System ข้อมูลภาษาซี ไม่ใช่ชื่อสินค้า แต่เป็นชื่อย่อที่ใช้ในเล่มนี้เพื่อบ่งเทคนิคที่ใช้

เพื่อเป็นประโยชน์ในการอธิบายต่อไปจะขอยกตัวอย่างระบบหนึ่งขึ้นมาซึ่งเป็นระบบการคิดเงินของบริษัทที่ปรึกษา ระบบนี้สามารถให้บริษัทที่ปรึกษาคิดค่าจ้างกับค่าใช้จ่ายสำหรับ โครงการงานของลูกค้าได้และสามารถเตรียมใบแจ้งหนี้ได้

ระบบบิลลิ่งของบริษัทที่ปรึกษา (Consultant's Billing System หรือ CBS) มีฐานข้อมูลซึ่งบันทึกเวลาและค่าใช้จ่ายที่เสียไปกับโครงการของลูกค้าและเตรียมใบแจ้งหนี้สำหรับเวลาการงานของลูกค้าและค่าใช้จ่ายทั้งหมดรวมทั้งคำนวณค่าจ้างจากอัตราค่าจ้างต่อชั่วโมงแก่ลูกค้า

ข้อความที่กล่าวมาทั้งหมดนี้เป็นการอธิบาย problem specification สำหรับระบบซึ่งเป็นขั้นตอนแรกในการออกแบบฐานข้อมูลตามที่อธิบายไว้ในหัวข้อที่ 2.2 เราจะสมมติว่าได้ผ่านขั้นตอนการออกแบบไปแล้วทั้ง 9 ขั้นตอน ผลที่ได้จากการออกแบบคือ ฐานข้อมูลมี 4 ไฟล์ ได้แก่ ไฟล์ที่ปรึกษา (consultants file), ไฟล์ลูกค้า (clients file), ไฟล์โครงการ (project file) และไฟล์มอบหมาย (assignments file) ส่วนผลลัพธ์จากการออกแบบส่วนอื่นๆ จะบอกตอนแปลงเป็นรูปแบบแผนฐานข้อมูลของภาษาซี

2.5.1 ภาษากำหนดลักษณะข้อมูลภาษาซี

ข้อมูลภาษาซี ใช้ DDL เหมือนกับวิธีที่ได้อธิบายไว้ในหัวข้อที่แล้วในตอนแรกจะเขียน DDL โดยใช้ภาษาซีจริงๆ ซึ่งจะถูกรวมไฟล์ไปพร้อมๆ กับโปรแกรมประยุกต์ จากนั้นต่อไปจะเปลี่ยนเป็นสร้างคอมไพเลอร์ขึ้นเพื่อแปลงรูปแบบของภาษาที่สร้างขึ้นใหม่ให้เป็น DDL ที่เขียนด้วยภาษาซีคือให้ผลลัพธ์เหมือนวิธีแรก การใช้วิธีที่สองจัดการจะแก้ไขได้ง่ายกว่าวิธีแรกโดยเฉพาะฐานข้อมูลที่ซับซ้อน ต่อไปเป็นการอธิบายการเขียนด้วยวิธีแรกก่อน

2.5.1.1 พจนานุกรมองค์ประกอบข้อมูลภาษาซี

ขั้นตอนหนึ่งในการออกแบบฐานข้อมูล CBS คือ การสร้างพจนานุกรมขององค์ประกอบ

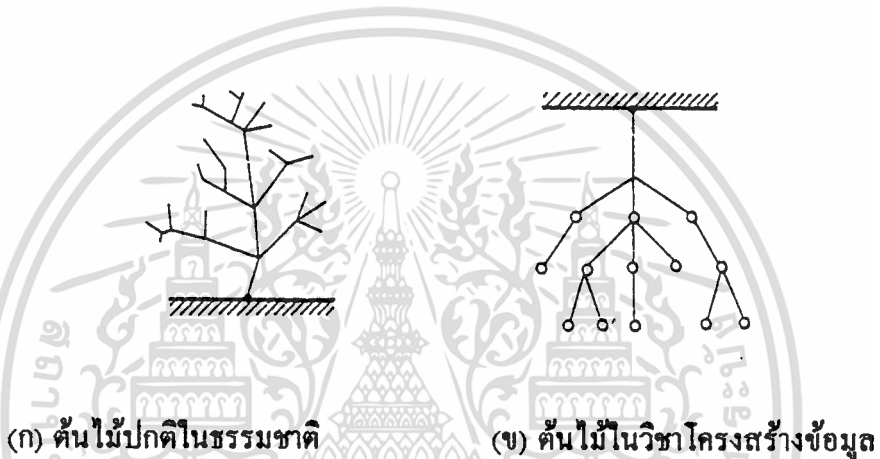
ชื่อขององค์ประกอบข้อมูล	ชนิดของข้อมูล	ความยาว
alien_no	ตัวเลข	5
alien_name	ตัวเลขและตัวอักษร	25
address	ตัวเลขและตัวอักษร	25
city	ตัวเลขและตัวอักษร	25
state	ตัวเลขและตัวอักษร	2
zip	ตัวเลข	5
phone	ตัวเลข	10
amt_due	ตัวเลขและตัวอักษร	8
project_no	ตัวเลข	5
project_name	ตัวเลขและตัวอักษร	25
amt_expended	สัญลักษณ์เงินตรา	9
consultant_no	ตัวเลข	5
consultant_name	สัญลักษณ์เงินตรา	25
rate	สัญลักษณ์เงินตรา	5
payment	สัญลักษณ์เงินตรา	9
expense	สัญลักษณ์เงินตรา	9
hours	ตัวเลข	2
date_paid	วันที่	6

ตารางที่ 2.2 การสร้างพจนานุกรมขององค์ประกอบข้อมูล

ข้อมูล (data element dictionary) จากการออกแบบก็จะได้พจนานุกรมขององค์ประกอบข้อมูลตาม ตารางที่ 2.2 จะอธิบายองค์ประกอบข้อมูลเหล่านี้ ในรูปแบบซึ่งโปรแกรมสำเร็จประยุกต์ระบบจัดการฐานข้อมูลและโปรแกรมรรถประโยชน์ (utility program) สามารถจะนำไปใช้ได้ ลักษณะของพจนานุกรมจะถูกแสดงตามวยากรสัมพันธ์ (Syntax) ของภาษาซี

2.6 โครงสร้างข้อมูลแบบต้นไม้

โครงสร้างต้นไม้ (tree) เป็นโครงสร้างชนิดไม่เชิงเส้นที่สำคัญที่สุดของโครงสร้างข้อมูล โครงสร้างต้นไม้มีความสัมพันธ์อย่างใกล้ชิดกับธรรมชาติของข่าวสารและวิธีการแปลงข่าวสารมาก โครงสร้างต้นไม้มีลักษณะที่สมชื่อของตนเองเพราะมีลักษณะคล้ายกิ่งก้านของต้นไม้ ต้นไม้ตามธรรมชาติจะงอกจากล่างไปบน ส่วนโครงสร้างข้อมูลที่มีลักษณะต้นไม้จะวาดหรือให้เจริญจากบนลงมาล่างจุดที่มีการแตกกิ่งก้านสาขาออกไปจะเรียกว่าโหนด (node) โดยข่าวสารจะเก็บอยู่ที่โหนดกิ่งที่ต่อระหว่างโหนดจะแสดงความสัมพันธ์ระหว่างโหนดเรียกว่าลิงค์ (link) อนึ่งตามรูปจุดปลายของกิ่งจะเรียกว่าโหนดด้วย



รูปที่ 2.6 ลักษณะของโครงสร้างต้นไม้

2.6.1 นิยามของโครงสร้างต้นไม้

สามารถนิยามโครงสร้างต้นไม้อย่างรีเคอร์ซีฟว่าเป็นกลุ่มของโหนด T ดังนี้

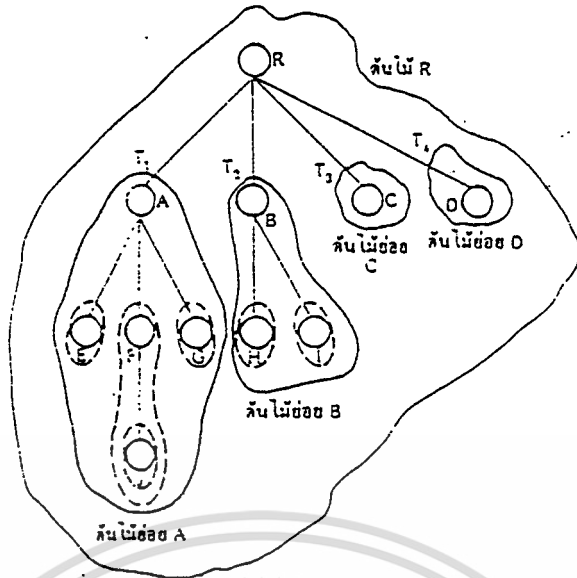
- ก. มีโหนดพิเศษโหนดหนึ่งเรียกว่า รากหรือรูต (root node), R
 - ข. โหนดอื่นๆ ที่ไม่ใช่รูตสามารถถูกแบ่งย่อยออกเป็น n กลุ่ม โดยที่แต่ละกลุ่มไม่มีโหนดร่วมกันเลย สมมติให้ชื่อแต่ละกลุ่มเป็น T_1, T_2, \dots, T_n ($n > 0$) ซึ่งแต่ละกลุ่มก็เป็นต้นไม้เหมือนกัน แต่จะเรียกว่าเป็น ต้นไม้ย่อย (subtree) ของโหนด R ดังรูปที่ 2.7
- ซึ่งจะเห็นได้ว่า

R เป็นรูตโหนดของต้นไม้ย่อย A, B, C, D

A เป็นรูตโหนดของต้นไม้ย่อย E, F, G

F เป็นรูตโหนดของต้นไม้ย่อย J

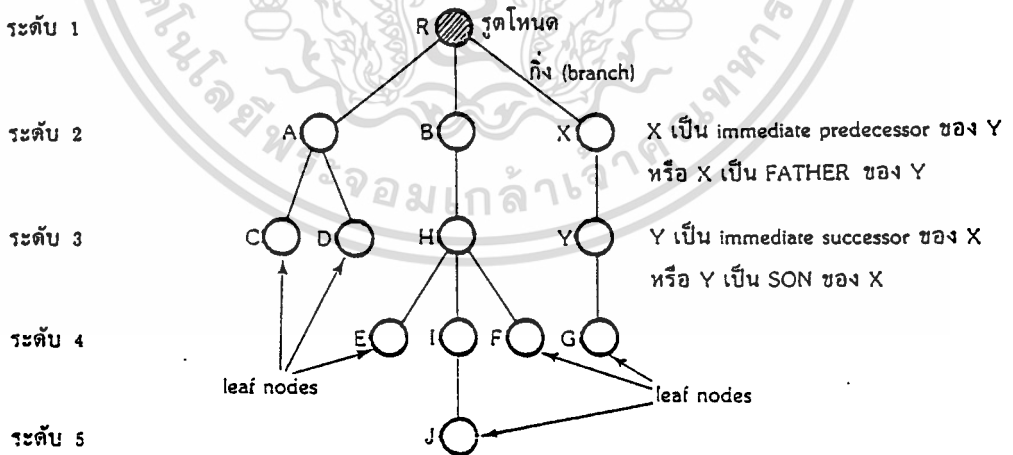
B เป็นรูตโหนดของต้นไม้ย่อย H และ I เป็นต้น



รูปที่ 2.7 แสดงลักษณะของต้นไม้ตามนิยามแบบรีเคอร์ซีฟ

2.6.1.1 การเรียกส่วนต่างๆ ของต้นไม้

ลักษณะของโครงสร้างต้นไม้มีลักษณะเหมือนการลำดับบรรพบุรุษ ดังนั้น ชื่อที่ใช้จึงได้มาจากลำดับบรรพบุรุษ เช่น FATHER, SON, GRANDSON ฯลฯ ที่ใช้บ่อยจะเป็นความสัมพันธ์ของ พ่อ-ลูก (หรือ FATHER-SON)



รูปที่ 2.8 การเรียกชื่อส่วนต่างๆ ของโครงสร้างต้นไม้

2.6.1.2. ระดับของโหนด (Level)

ระดับของโหนดหนึ่งๆ แสดงถึงหน่วยระยะทางตามแนวกิ่งของโหนดนั้น ว่าอยู่ห่างจากรoot โหนดเท่าไร ถ้ากำหนดว่ารุดโหนดของต้นไม้ที่อยู่ที่ระดับ 1 และกิ่งทุกกิ่งมีความยาวเท่ากัน คือยาว 1 หน่วย เลขระดับของโหนดใดๆ คือจำนวนกิ่งที่น้อยที่สุดจากรุดโหนดบวกหนึ่งเช่น F มีเลขระดับเป็น 4 เป็นต้น

2.6.1.3. ดีกรีของโหนด (Level Degree)

ดีกรีของโหนด คือจำนวนต้นไม้ย่อยของโหนดนั้นหรือจำนวน branch ที่ออกจากโหนดในแนวลงมา ตามรูปที่ 2.8 โหนด x มีดีกรี 1 โหนด A มีดีกรี 2 ส่วนโหนด H มีดีกรี 3 โหนด B มีดีกรี 1 และโหนด E มีดีกรี 0 เป็นต้น

2.6.1.4. โหนดที่เป็นใบ (Leaf Node หรือ Terminal Node)

โหนดที่เป็นใบหมายถึงโหนดที่มีดีกรีเป็น 0 เช่น โหนด C,D,E,I,F และ G ส่วนโหนดที่มีดีกรีไม่เท่ากับ 0 เรียกว่าโหนดภายใน หรือ interior or branch node.

2.6.1.5. Predecessor และ Successor (หรือ Ancestor และ Descendant)

Predecessor หมายถึงผู้มาก่อน

Successor หมายถึงผู้มาทีหลัง

ตามรูปที่ 2.8 R,B,H จะเป็น predecessor ของโหนด E,I,F,J โหนด E และ F ไม่มี successor ส่วนโหนด J เป็น successor ของโหนด I และสรุปได้ว่าโหนด I จะเป็น predecessor ของโหนด J (หรือ J เป็น successor ของโหนด I) ถ้าโหนด I และโหนด J มีกิ่ง (หนึ่งกิ่งหรือมากกว่า) เชื่อมถึงกันและโหนด I มีเลขหมายระดับน้อยกว่าโหนด J

2.6.1.6. Immediate Successor หรือ SON ของโหนด I

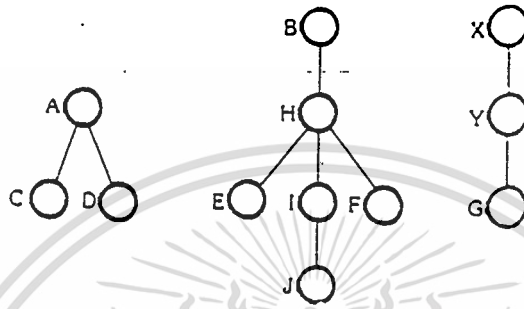
immediate successor คือโหนดทั้งหลายของต้นไม้ย่อย I ที่มีค่าระดับต่ำกว่าโหนด I อยู่หนึ่ง เช่น SON ของโหนด H คือโหนด E,I และ F

2.6.1.7. Immediate Predecessor หรือ FATHER ของโหนด I

immediate predecessor คือ โหนดที่มีค่าระดับสูงกว่าโหนด I อยู่หนึ่ง เช่น FATHER ของโหนด J คือโหนด I, FATHER ของโหนด I คือโหนด H เป็นต้น

2.6.1.8 ป่า (Forest)

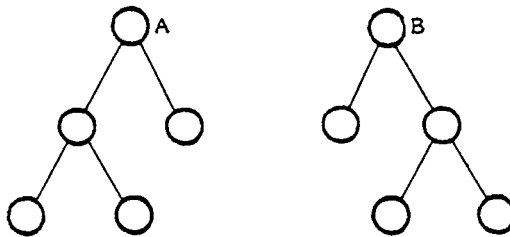
ป่า หมายถึงกลุ่มของต้นไม้ที่เกิดจากการเอาจุดโหนดของต้นไม้ต้นหนึ่งทิ้งไปตามรูปที่ 2.8 ถ้าลบจุดโหนด R ออกจะได้ต้นไม้ 3 ต้นประกอบขึ้นเป็นป่า ดังรูปที่ 2.9



รูปที่ 2.9 ป่าที่ประกอบด้วยต้นไม้ A,B และ X

2.6.2 ต้นไม้ที่มีแบบแผน (Ordered tree)

เราเรียกต้นไม้ A ว่าเป็นต้นไม้ที่มีแบบแผน ถ้าโหนดต่างๆ ของต้นไม้มีความสัมพันธ์ที่แน่นอนประการหนึ่ง เช่น ก่อน, ไปทางขวา, ไปทางซ้าย เป็นต้น ปกติต้นไม้จะถือว่าเป็นแบบที่ไม่มีแบบแผนนั้น คือไม่ได้กำหนดความสัมพันธ์ (เชิงตำแหน่ง) ระหว่างโหนด ฉะนั้นต้นไม้รูปที่ 2.10 ซึ่งจึงถือได้ว่าเหมือนกันเพราะมีจำนวนโหนดและจำนวนต้นไม้ย่อยเท่ากันนอกจากนั้นยังสามารถ "หมุน" ต้นไม้ทั้งสองต้นนี้ให้อยู่ในแบบเดียวกันได้อีกด้วย แต่ถ้าเรากำหนดว่าต้นไม้ A และ B มีแบบแผนถือว่าต้นไม้ A และต้นไม้ B เป็นคนละต้น เพราะต้นไม้ A มีโบททางซ้าย 2 โบ ส่วนต้นไม้ B มีโบททางขวา 2 โบ



รูปที่ 2.10 ต้นไม้ A และ B จะแตกต่างกันถ้ากำหนดเป็นต้นไม้ที่มีแบบแผน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.3. การแทนโครงสร้างต้นไม้ในคอมพิวเตอร์

ต้นไม้แบบที่กล่าวมาแล้วนี้สามารถเก็บในคอมพิวเตอร์ โดยอาศัยโครงสร้างของแต่ละโหนดดังตารางที่ 2.3

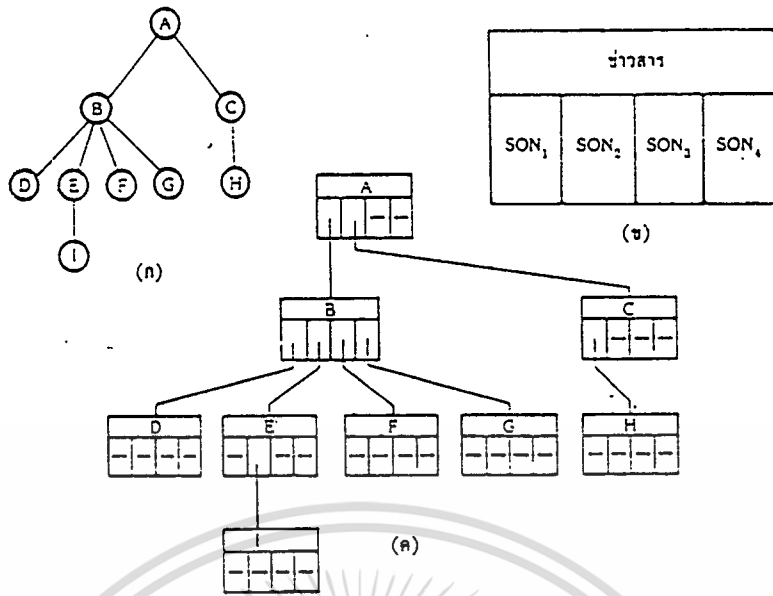
เนื่องจากแต่ละโหนดของต้นไม้จะมีที่ SON ก็ได้ ดังนั้นจึงให้ส่วนลิงค์ของโหนดมี k ส่วน ถ้าโหนดส่วนหนึ่งในต้นไม้มี 3 SONs ก็จะได้ $k = 3$ ซึ่งแสดงว่าโครงสร้างของโหนดนั้นนอกจากจะมีส่วนข่าวสารแล้ว ยังมีส่วนลิงค์ 3 ส่วน คือ SON₁, SON₂, และ SON₃ แต่ถ้าอีก

ข่าวสาร					
son ₁	son ₂	son ₃			son _k

ตารางที่ 2.3 โครงสร้างโหนดของต้นไม้ทั่วไป

โหนดหนึ่งในต้นไม้มี 5 SON, โครงสร้างของโหนดนั้นจะมีส่วนลิงค์ 5 ส่วน คือ SON₁ ถึง SON₅, จะเห็นได้ว่าการแทนโหนดลักษณะนี้ขนาดของพื้นที่ความจำ ที่ใช้สร้างแต่ละโหนด จะมีขนาดไม่คงที่ขึ้นอยู่กับจำนวน SON ของโหนดนั้น การจัดการกับโครงสร้างที่มีขนาดไม่เท่ากันตลอดนี้ จะมีความยุ่งยากมากในแง่การจัดการและเลิกจัดการพื้นที่ความจำให้แก่โหนด วิธีหนึ่งที่จะผ่านข้อยุ่งยากนี้ไปคือ ให้ขนาดของโหนดที่คงที่ตลอด ทั้งนี้ให้กำหนดว่า ค่า k คือค่าคิรีสูงสุดของโหนดในต้นไม้ตามรูปที่ 2.11 ต้นไม้ A มีคิรีสูงสุดเท่ากับ 4 (จำนวน SON ของโหนด B) ดังนั้นโหนดที่ใช้จะมีโครงสร้างดังรูปที่ 2.11 (ข)

โครงสร้างต้นไม้ดังรูปที่ 2.11 อาจจำลองด้วยอาร์เรย์ 5 อาร์เรย์ชื่อ INFO (9), SON₁(1:9), SON₂(1:9), SON₃(1:9), SON₄ (1:9) แต่ละอาร์เรย์จะมี 9 ช่อง เนื่องจากต้นไม้มี 9 โหนด รูปที่ 2.12 เป็นลักษณะค่าในแต่ละช่องของอาร์เรย์ ค่าตัวเลขที่เติมไว้ได้มาจากการมองคู่ต้นไม้รูปที่ 2.11 (ก) และเลขลำดับของข่าวสารภายในอาร์เรย์ ทั้งนี้ต้องเก็บข่าวสารของโหนดไว้ในอาร์เรย์ชื่อ INFO ดังนั้น INFO (1) จะเก็บค่า A SON₁ (1) จะต้องชี้ไปยัง B ซึ่งคือ 2 ส่วน SON₂ (1) ต้องชี้ไปยัง C ซึ่งอยู่ในตำแหน่ง 3 ใน INFO ดังนั้นค่าของ SON₂ (1) จึงเป็น 3 ส่วนค่าของ SON₃ (1) และ SON₄(1)จะเป็นศูนย์ (หรือตัวพิเศษอะไรก็ได้ ที่แสดงว่าเป็นส่วนลิงค์ที่เป็นศูนย์หรือว่างเปล่า) ในที่นี้เขียนแทนด้วย 0



รูปที่ 2.11 การแทนต้นไม้ รูป (ก) โดยโครงสร้างโหนดรูป (ข) ซึ่งจะได้อโครงสร้าง (ค)

	1	2	3	4	5	6	7	8	9
INFO	A	B	C	D	E	F	G	H	I
SON ₁	2	4	8	—	9	—	—	—	—
SON ₂	3	5	—	—	—	—	—	—	—
SON ₃	—	6	—	—	—	—	—	—	—
SON ₄	—	7	—	—	—	—	—	—	—

รูปที่ 2.12 การแทนโครงสร้างต้นไม้ตามรูปที่ 2.11 (ก) ด้วยอาร์เรย์ 5 อาร์เรย์ โดยใช้โครงสร้างโหนดตามรูปที่ 5.7(ข) โดยที่ 1 โหนดใช้หนึ่งคอลัมน์ทั้ง 5 อาร์เรย์

จากรูปที่ 2.11 หรือ 2.12 จะเห็นได้ว่า ช่องส่วนที่ใช้เป็นพอยน์เตอร์จะว่างเปล่าและไม่ได้ใช้ประโยชน์อยู่มาก ทั้งนี้เนื่องจากคิกริชของโหนดต่างกัันนั่นเอง ถ้าทุกโหนดในต้นไม้ไม่มีคิกริชเท่ากัน การแทนในคอมพิวเตอร์จะไม่มีช่องว่างในส่วนพอยน์เตอร์เลย ถ้าคิกริชระหว่างโหนดต่างกัันมาก ถ้าส่วนพอยน์เตอร์ก็จะสูญเปล่มาก เพราะให้ขนาดของโหนดที่ใช้เท่ากันตลอดคั้งที่กล่าวไว้แล้ว

ปกติถ้าต้นไม้ไม่มีคิกริช k (k คือ คิกริชสูงสุดของโหนดในต้นไม้ นั้น) และ มีโหนด n โหนด การแทนโดยโครงสร้างที่กล่าวมาแล้วจะต้องใช้

ส่วนข่าวสาร n ช่อง , ส่วนพอยน์เตอร์ kn ช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนพอยน์เตอร์ kn ช่อง

เนื่องจากต้นไม้มี n โหนด ดังนั้นจำนวนช่องพอยน์เตอร์ที่ใช้จริงเพียง $n-1$ ช่อง(ไม่นับรูตโหนด เพราะไม่มีโหนดใดชี้ไปยังรูตโหนด) ดังนั้น

$$\begin{aligned} \text{จำนวนพอยน์เตอร์ที่ไม่ได้ใช้ประโยชน์} &= kn - (n-1) \cdot 0 \\ &= n(k-1) + 1 \end{aligned}$$

ตามรูปที่ 2.11 หรือ 2.12 เมื่อ $n = 9$ และ $k = 4$ จะเห็นได้ว่า จำนวนช่องที่ไม่ได้ใช้ประโยชน์เท่ากับ $9(4-1) + 1 = 28$ ช่อง

ในหัวข้อต่อไปเราจะศึกษาถึงต้นไม้ไบนารี และวิธีการแทนต้นไม้ในคอมพิวเตอร์ อย่างมีประสิทธิภาพ สำหรับต้นไม้ธรรมดาอาจแปลงเป็นต้นไม้ไบนารีได้

2.6.4 ต้นไม้แบบไบนารี (Binary Tree)

2.6.4.1 นิยาม

ต้นไม้ไบนารีเป็น *rooted binary tree* ที่ว่างเปล่า หรือประกอบด้วยรูตโหนดและต้นไม้ไบนารี 2 กลุ่มที่ไม่มีโหนดร่วมกันแต่ละกลุ่มจะมีชื่อเรียก (โดยตำแหน่งที่อยู่หรือที่เขียน) ว่าต้นไม้ย่อยทางซ้าย (*Left subtree*) และต้นไม้ทางขวา (*right subtree*) ตามลำดับ คำว่า ว่างเปล่า ในนิยามหมายความว่า ต้นไม้ไบนารีต้นนั้นมีแต่รูตโหนดเพียงโหนดเดียวนั้น

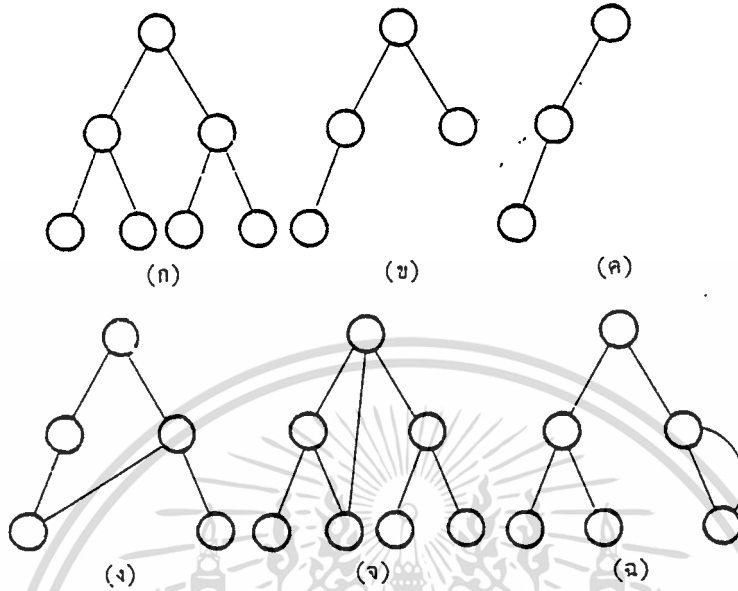
โครงสร้างต้นไม้ ที่ใช้ในการจัดการข่าวสารโดยคอมพิวเตอร์นั้น ส่วนใหญ่จะเป็นแบบโครงสร้างต้นไม้ไบนารี ลักษณะพิเศษจำเพาะของต้นไม้ไบนารีคือแต่ละโหนดมีคิกร้อยอย่างมากที่สุดเท่ากับ 2 นอกจากนี้ต้นไม้ไบนารียังเป็นต้นไม้ที่แบบแผนความสัมพันธ์เชิงตำแหน่งระหว่างโหนด นั่นคือที่โหนดใดๆ ต้นไม้ย่อยของโหนดนั้นจะเรียกชื่อ เฉพาะว่าเป็น ต้นไม้ย่อยทางขวา หรือต้นไม้ย่อยทางซ้าย ถ้าดูแค่โหนดจากโหนดหนึ่งจะเป็นความสัมพันธ์ FATHER และมีอีก 2 โหนดต่อลงไปเป็นโหนดทางซ้ายหรือ *left son* หรือ *LSON* และมีโหนดทางขวา หรือ *right son* หรือ *RSON* ดังรูป

2.7 เพิ่มข้อมูล (data file)

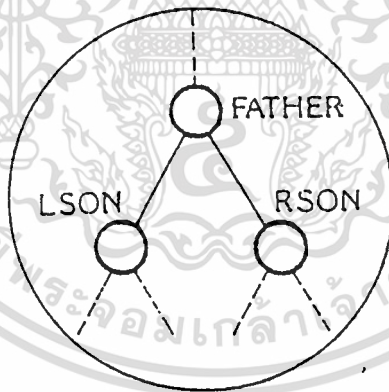
2.7.1 ความหมายของเพิ่มข้อมูลและลักษณะการประมวลผลเพิ่มข้อมูล

เพิ่มข้อมูล หมายถึงการรวบรวมข้อมูลที่สัมพันธ์กันหรือเกี่ยวข้องกันมาไว้ที่เดียวกันเพื่อให้สะดวกในการเรียกใช้และค้นคว้าหาข้อมูล ดังนั้นในภาษาซึ่งจะมองข้อมูลที่เก็บอยู่ในเพิ่มข้อมูล มีลักษณะที่เป็นข้อมูลที่อยู่ต่อเนื่องกัน ตั้งแต่ต้นเพิ่มข้อมูลจนจบเพิ่มข้อมูล โดยไม่ผู้แบ่งเป็นระเบียบ (record) เหมือนกับภาษาระดับสูงต่างๆ ไปในการประมวลผลเพิ่มข้อมูลในแต่ละครั้งผู้เขียนโปรแกรมจะเป็นผู้กำหนดขอบเขตในการนำข้อมูลไปประมวลผลเอง

ลักษณะการประมวลผลเพิ่มข้อมูลโดยทั่วไปจะแบ่งเป็นสองแบบใหญ่ๆ คือ



รูปที่ 2.13 (ก), (ข), (ค) ต้นไม้ไบนารีและ (ง), (จ), (ฉ) ไม่ใช่ต้นไม้ไบนารี



รูปที่ 2.14 โครงสร้างต้นไม้ไบนารี LSON

- ก. การบันทึกข้อมูลลงในเพิ่มข้อมูล
 - ข. การอ่านข้อมูลที่เก็บอยู่ในเพิ่มข้อมูลออกมาทำงาน
- การอ่านทั้งสองแบบนี้ จะมีขั้นตอนการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การบันทึกข้อมูล	การอ่านข้อมูล
1. เปิดเพิ่มข้อมูล (open file)	1. เปิดเพิ่มข้อมูล (open file)
2. การบันทึกข้อมูลลงบนแฟ้ม (write file)	2. การอ่านข้อมูลจากแฟ้มข้อมูล (read file)
3. ปิดเพิ่มข้อมูล (close file)	3. ปิดเพิ่มข้อมูล (close file)

ตารางที่ 2.4 ลักษณะการประมวลเพิ่มข้อมูล

2.7.1.1 การบันทึกข้อมูล

ผู้เขียนโปรแกรม (Programmer) สามารถบันทึกข้อมูลลงไปในแฟ้มข้อมูลได้ต่อเมื่อมีการเปิดเพิ่มข้อมูลเรียบร้อยแล้ว ซึ่งจะทำให้มีตัวชี้ตำแหน่งข้อมูลในแฟ้ม (filepointer) ซึ่งอยู่ที่ตำแหน่งแรกของแฟ้มข้อมูล โดยตัวชี้ตำแหน่งแฟ้มข้อมูลนี้จะใช้บอกถึงตำแหน่งที่ทำการประมวลผลบนแฟ้มข้อมูลให้และในขณะเดียวกันก็จะมีการเตรียมบัฟเฟอร์ (buffer) ซึ่งเป็นที่เก็บข้อมูลชั่วคราวที่สร้างขึ้นภายในหน่วยความจำให้ ด้วยข้อมูลที่จะทำการประมวลผลจะถูกเก็บไว้ในบัฟเฟอร์ก่อน เช่นการบันทึกข้อมูลลงบนแฟ้มข้อมูล ข้อมูลจะนำไปเก็บไว้ในบัฟเฟอร์ก่อนโดยผู้เขียนโปรแกรมมีความรู้สึกว่าได้บันทึกข้อมูลนั้นลงบนแฟ้มข้อมูลเรียบร้อยแล้ว การบันทึกข้อมูลแต่ละครั้งจะทำให้ตำแหน่งของตัวชี้ตำแหน่งข้อมูลในแฟ้มข้อมูลเปลี่ยนแปลงไปตามจำนวนข้อมูลที่บันทึก เมื่อมีการบันทึกข้อมูลจนเต็มบัฟเฟอร์เครื่องจะนำข้อมูลในบัฟเฟอร์นั้นเข้าไปบันทึกในแฟ้มข้อมูลให้เองโดยอัตโนมัติ ในการประมวลผลแฟ้มข้อมูลแต่ละครั้งผู้เขียนโปรแกรมอาจจะเปิดเพิ่มข้อมูลได้มากกว่า 1 แฟ้มข้อมูล ซึ่งเครื่องก็จะเตรียมบัฟเฟอร์ให้เท่ากับจำนวนแฟ้มข้อมูลที่เปิดนั้น โดยมีตัวชี้ตำแหน่งข้อมูลในแฟ้มแสดงตำแหน่งในแฟ้มข้อมูลคนละตัว ดังรูปแสดงไว้ข้างบน ในกรณีที่มีการบันทึกข้อมูลไม่เต็มบัฟเฟอร์แล้วมีการปิดเพิ่มข้อมูลเครื่องก็จะนำข้อมูลที่เหลืออยู่ในบัฟเฟอร์ทั้งหมดบันทึกลงไปในแฟ้มข้อมูลให้ก่อนที่จะปิดเพิ่มข้อมูลนั้น

2.7.1.2 การอ่านข้อมูล

เมื่อมีการใช้ฟังก์ชันอ่านข้อมูลจากแฟ้มข้อมูล เครื่องจะนำข้อมูลมาเก็บไว้ในบัฟเฟอร์ให้ก่อน โดยที่ผู้เขียนโปรแกรมจะมีความรู้สึกว่าได้อ่านข้อมูลจากแฟ้มข้อมูลโดยตรงในการอ่านข้อมูลแต่ละครั้งนั้นจะได้ข้อมูลในตำแหน่งที่ตัวชี้ตำแหน่งข้อมูลในแฟ้มชื่ออยู่และตำแหน่งของตัวชี้ตำแหน่งข้อมูลในแฟ้มก็จะเปลี่ยนไปตามขนาดของข้อมูลที่อ่านออกมาทำงานจนถึงสิ้นสุดแฟ้มข้อมูล (end of file)

2.7.2 ฟังก์ชันที่ใช้ในการประมวลผลเพิ่มข้อมูล

ฟังก์ชันที่ใช้ในการประมวลผลเพิ่มข้อมูล แบ่งได้เป็นสองระดับใหญ่ๆ คือ

2.7.2.1 ฟังก์ชันมาตรฐานในการรับข้อมูลและแสดงผลข้อมูล (standrd I/O)

เป็นฟังก์ชันที่เหมาะสมกับการเขียนโปรแกรมที่มีการประมวลผลผ่านบัฟเฟอร์

2.7.2.2 ฟังก์ชันระดับระบบที่ใช้ในการรับข้อมูลและแสดงผลข้อมูล (System level I/O)

เป็นฟังก์ชันที่เหมาะสมกับการเขียนโปรแกรมโดยตรงโดยไม่ต้องใช้บัฟเฟอร์ฟังก์ชันต่างๆ เหล่านี้จะเข้าถึง (access) โดยตรงกับโปรแกรมระบบดำเนินการ (Operating System) ทำให้มีความเร็วในการประมวลผลสูงและประหยัดขนาดของหน่วยความจำ

2.7.3 ชนิดของเพิ่มข้อมูล

เนื่องจากเทอร์โบซี (turbo-c) เป็น compiler ที่ใช้บน MS-DOS รหัสพิเศษบางอย่างเช่น \n (new line) มีความหมายถึงการขึ้นบรรทัดใหม่ในภาษาซี แต่เมื่อนำมาบันทึกลงบนเพิ่มข้อมูลที่จะนำไปใช้บน MS-DOS ความหมายจะเปลี่ยนไป ดังนั้นเพื่อให้ใช้เพิ่มข้อมูลที่สร้างขึ้นด้วยภาษาซีสามารถใช้กับซอฟต์แวร์อื่นๆ บน MS-DOS ได้ จึงแบ่งชนิดของเพิ่มข้อมูลออกเป็นสองชนิดคือ โดยผู้เขียนโปรแกรมจะเป็นผู้กำหนดชนิดของเพิ่มข้อมูลที่ใช้ได้เอง

2.7.3.1 text file

2.7.3.2 Binary file

โดยผู้เขียนโปรแกรมจะเป็นผู้กำหนดชนิดของเพิ่มข้อมูลที่ใช้ได้เอง

2.7.3.1 text file

ในการกำหนดเพิ่มข้อมูลเป็นชนิด text file ถ้ามีการบันทึกข้อมูลลงไปบนเพิ่มข้อมูลแล้วพบรหัส \n (new line) จะถูกเปลี่ยนเป็น carriage return และ line feed ให้แล้วบันทึกลงในเพิ่มข้อมูลนั้นไว้ในกรณีที่เป็นกรอ่านข้อมูลจากเพิ่มข้อมูลถ้าพบรหัส carriage return และ line feed ก็ จะเปลี่ยนกลับเป็น new line ให้

2.7.3.2 Binary File

ในการกำหนดเพิ่มข้อมูลเป็นชนิด binary file ถ้าพบรหัส \n (new line) จะไม่มีการเปลี่ยนเป็น carriage return และ line feed ให้

2.7.4 การเปิดเพิ่มข้อมูล (Open File)

การเปิดเพิ่มข้อมูลเป็นการประกาศว่าจะมีการใช้เพิ่มข้อมูลพร้อมกับกำหนดตัวชี้ตำแหน่งข้อมูลในแฟ้มให้กับแฟ้มข้อมูลที่ต้องการใช้นั้น การเปิดเพิ่มข้อมูลทำได้โดยใช้ฟังก์ชัน `fopen ()` ซึ่งอยู่ในแฟ้มข้อมูล `stdio.h` และตัวชี้ตำแหน่งข้อมูลในแฟ้มจะถูกกำหนดให้เป็นตัวแปรชนิด `FILE` โดยประกาศได้ดังนี้

รูปแบบ

```
FILE *fp;
```

`fp` เป็นตัวแปรชนิด `FILE` ที่ทำหน้าที่เป็นตัวชี้ตำแหน่งข้อมูลในแฟ้ม สำหรับฟังก์ชันที่ใช้ในการเปิดเพิ่มข้อมูลจะมีรูปแบบดังนี้

รูปแบบ

```
FILE *fp
```

```
fp= fopen (filename,mode)
```

`filename` เป็นชื่อข้อมูลของแฟ้มข้อมูลที่ต้องการเปิด

`mode` เป็นลักษณะของแฟ้มข้อมูลที่จะเปิดว่าต้องการเปิดขึ้นมาว่าต้องการอะไร

mode		ความหมาย
text file	binary file	
r หรือ rt	rb	เปิดเพิ่มข้อมูลสำหรับการอ่านข้อมูล
w หรือ wt	wb	สร้างข้อมูลสำหรับการบันทึกข้อมูล
a หรือ at	ab	ใช้สำหรับบันทึกข้อมูลต่อจากแฟ้มข้อมูลเดิม
r+ หรือ r+t	r+b	เปิดเพิ่มข้อมูลสำหรับอ่าน/บันทึกข้อมูล
w+ หรือ w+t	w+b	สร้างแฟ้มข้อมูลสำหรับอ่าน/บันทึกข้อมูล
a+ หรือ a+t	a+b	ใช้ต่อหรือสร้างข้อมูลเพิ่มเข้าไปในแฟ้มข้อมูลเดิม

ตารางที่ 2.5 ลักษณะของแฟ้มข้อมูล

2.8 สถาปัตยกรรมของ IBM PC และแบบแผนการจัดการหน่วยความจำภาษาซี

ตัวประมวลผลในตระกูล 8086 ที่มีใช้ในเครื่องคอมพิวเตอร์สามารถที่จะอ้างหน่วยความจำได้ถึง 1 เมกกะไบต์ สำหรับเบอร์ 80286 และ 80386 หรือสูงกว่านี้สามารถที่จะอ้างหน่วยความจำได้สูงกว่านี้ เมื่อทำงานในโหมดป้องกัน (protected mode) ตัวประมวลผลเหล่านี้เมื่ออยู่ภายใต้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สภาพแวดล้อม MS-DOS จะปฏิบัติงานอยู่ในโหมดจริง (real mode) จึงถูกจำกัดอยู่ที่ 16 bits ซึ่งสามารถที่จะอ้างแอดเดรสได้โดยตรงเพียง 64 กิโลไบต์ จึงมีวิธีคัดแปลงบางประการ

เพื่อที่จะอ้างหน่วยความจำให้ได้ถึง 1 เมกกะไบต์ นอกจากยังมีรีจิสเตอร์ที่ใช้งานทั่วไปของตัวประมวลผลเหล่านี้แล้วยังมีเซกเมนต์รีจิสเตอร์อีก 4 ตัว คือ code(CS), data(DS), stack (ST) และ extra (ES) เมื่อมีคำสั่งใดๆ อ้างถึงหน่วยความจำมันจะระบุแอดเดรส 16-bit โดยตรงหรืออ้างผ่าน register value และ offset ค่าที่ได้ 16-bit นี้จะถูกรวมเข้ากับรีจิสเตอร์ตัวใดตัวหนึ่งจากทั้ง 4 ตัวนี้

$$\text{full address} = \text{effective_address} + 16 \times \text{segment_value}$$

เซกเมนต์รีจิสเตอร์ทั้ง 4 ตัวจะยอมให้โปรแกรมอ้างพื้นที่หน่วยความจำ 64 กิโลไบต์ได้ต่างกัน 4 พื้นที่ ค่าของเซกเมนต์รีจิสเตอร์สามารถเปลี่ยนได้ด้วยโปรแกรมแต่มันจะไปมีผลเสียในเรื่องขนาดของโปรแกรมและความเร็วในการ execute ซึ่งมีกลยุทธ์วิธีการอยู่หลายวิธีที่เรียกว่า memory model เพื่อที่จะจัดการกับรีจิสเตอร์เหล่านี้เมื่ออยู่ในขั้นตอนการเขียนโปรแกรม ดังต่อไปนี้

2.8.1 Tiny Model

เซกเมนต์รีจิสเตอร์ทั้ง 4 ตัวถูกชี้ไปที่พื้นที่ 64 กิโลไบต์อันเดียวกันเมื่ออยู่ในขบวนการ execute ของโปรแกรม รูปแบบเป็นแบบที่ไม่ซับซ้อนและไม่มี overhead ที่ใช้สำหรับเปลี่ยนเซกเมนต์รีจิสเตอร์ แต่อย่างไรก็ตามก็จะถูกจำกัดการใช้หน่วยความจำอยู่เพียง 64 กิโลไบต์

2.8.2 Small model

รีจิสเตอร์ที่ทำหน้าที่ชี้ที่อยู่ของโปรแกรมหรือ code segment Register จะทำหน้าที่ชี้บอกตำแหน่งของชุดคำสั่งของโปรแกรมในส่วนเซกเมนต์รีจิสเตอร์ตัวอื่นๆ จะชี้ไปยังพื้นที่ต่างหากอีกที่หนึ่งซึ่งมีไว้เพื่อเก็บข้อมูลขณะที่ยังอยู่ในขบวนการ execute นั้น เซกเมนต์รีจิสเตอร์จะไม่มี การเปลี่ยนแปลง โดยมีพื้นที่ของชุดคำสั่งและพื้นที่ของข้อมูลต่างก็สามารถปรับได้ถึง 64 กิโลไบต์ และไม่มี overhead สำหรับเปลี่ยนเซกเมนต์รีจิสเตอร์

2.8.3 Medium Model

ในรูปแบบนี้จะสามารถเปลี่ยน code segment register ได้เพียงแต่เรียกใช้ฟังก์ชัน call ส่วนรีจิสเตอร์ตัวอื่น จะถูกกำหนดให้ชี้ไปที่พื้นที่ที่เป็นข้อมูลและจะไปเปลี่ยนไม่ได้ด้วยวิธีการนี้ ทำให้สามารถอ้างพื้นที่ของชุดคำสั่ง (code) ได้ถึง 1 เมกกะไบต์ โดยมีพื้นที่ส่วนหนึ่ง 64 กิโลไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับเป็นพื้นที่ข้อมูล อย่างไรก็ตามก็จะมีผลเสียให้ขนาดของโปรแกรมใหญ่ขึ้นบ้างและความเร็วในการ execute ลดลงบ้างพอสมควร

2.8.4 Large Model

code segment register จะถูกเปลี่ยนแปลงในระหว่างการเรียกใช้ฟังก์ชัน call นอกจากนี้ data segment register และ extra segment register จะถูกเปลี่ยนแปลงตามความจำเป็นที่จะต้องอ้างหน่วยความจำของข้อมูลให้ได้ถึง 1 เมกกะไบต์ เวลาที่เสียไปในการ execute จะเพิ่มขึ้น 25 เปอร์เซ็นต์ หรือมากกว่านี้ และจะส่งผลให้โปรแกรมต้องการหน่วยความจำ RAM เพิ่มขึ้นอีกมาก

2.8.5 Compact Model

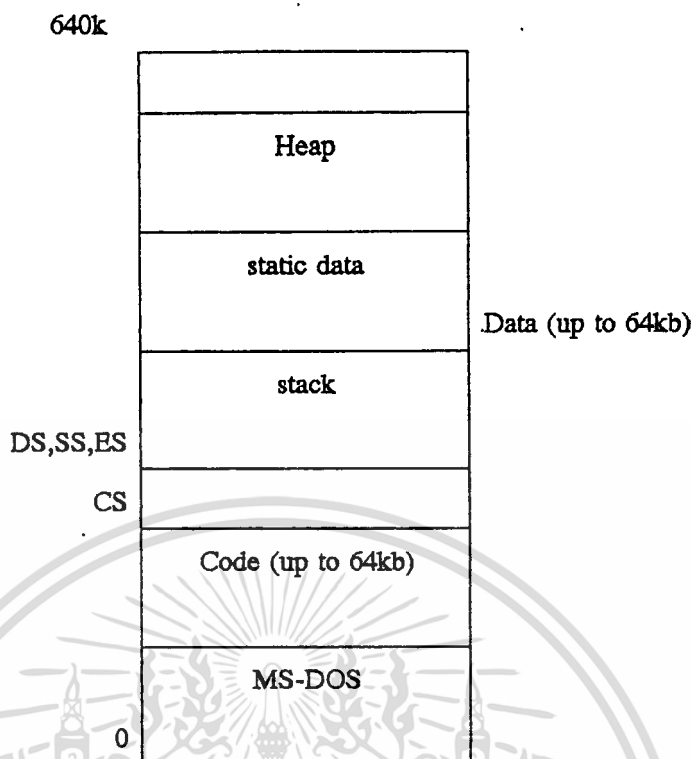
แบบนี้จะตรงกันข้ามกับแบบ Medium คือ code segment register จะเปลี่ยนแปลงไม่ได้ด้วยคำสั่ง call จึงจำกัดขนาดชุดคำสั่งอยู่ที่ 64 กิโลไบต์ ส่วน data segment register และ extra segment register สามารถเปลี่ยนแปลงได้เพื่อให้ซีแอด्रेसให้ได้ถึง 1 เมกกะไบต์ ของส่วนข้อมูลและจะส่งผลให้ขนาดและความเร็วของโปรแกรมใกล้เคียงกับ โมเดลแบบ Large

2.8.6 Huge Model

โมเดลนี้จะเหมือนกับโมเดล Large อย่างไรก็ตามคืออะเรย์ (array) จะถูกขอมให้ใหญ่กว่า 64 กิโลไบต์

2.9. โครงสร้างหน่วยความจำของโปรแกรมภาษาซี

ตามรูปที่ 2.15 แสดงโครงสร้างหน่วยความจำสำหรับโปรแกรมภาษาซีที่คอมไพล์ด้วยไมโครซอฟท์ซีหรือใช้ Quick C ในโมเดลแบบ small ชุดคำสั่งของโปรแกรมจะถูกวางในหน่วยความจำ RAM เหนือขึ้นไปจากบริเวณที่ถูกครอบครองโดย DOS ฟังก์ชันทั้งหมดจะทำงานโดยโปรแกรมให้อยู่กันเป็นกลุ่มพื้นที่ข้อมูลจะวางอยู่หลายส่วนพื้นที่ข้อมูลแบบคงที่ (static data area) จะประกอบด้วยตัวแปรที่จะต้องประกาศด้วยคีย์เวิร์ด static หรือ ประกาศอยู่นอกฟังก์ชันพื้นที่ของฮีฟ (heap) จะเป็นพื้นที่ที่ถูกจับจองแบบไดนามิก ซึ่งสามารถเปลี่ยนแปลงขนาดได้ (malloc และ calloc) ฮีฟจะโตขึ้นในลักษณะคั่งรูปตามความพอใจที่ต้องการพื้นที่หน่วยความจำเพิ่มขึ้น พื้นที่ stack จะประกอบด้วย ส่วนรีเทิร์นแอดเดรส, ส่วนเก็บค่ารีจิสเตอร์, อาร์กิวเมนต์ของฟังก์ชันและ

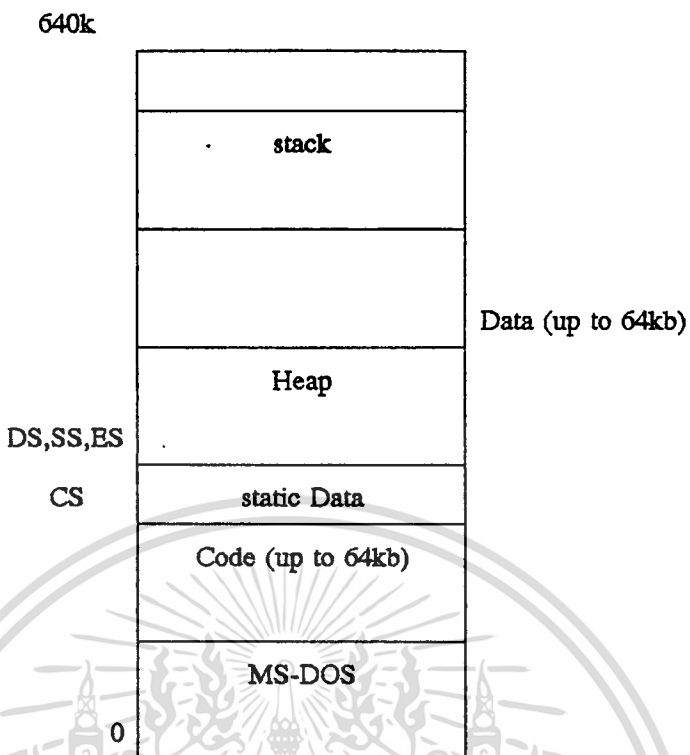


รูปที่ 2.15 โครงสร้างหน่วยความจำของโมเดลโครซอฟท์ซี โมเดล small

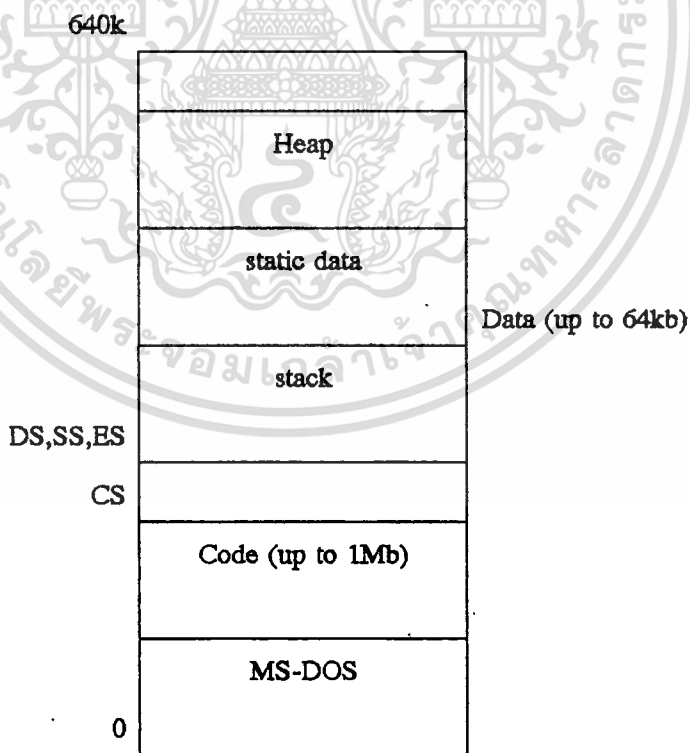
ตัวแปรแบบ automatic การโตของ stack จะมีลักษณะพุ่งลงคังรูป เมื่อโปรแกรมเข้าสู่ฟังก์ชันและจะหดตัวขึ้นเมื่อโปรแกรมออกจากฟังก์ชัน สำหรับพื้นที่ว่างๆที่อยู่ใต้พื้นที่แรงแงของสแตคนั้น มีไว้สำรองการเติบโตของสแตคนั่นเอง

2.10 โครงสร้างหน่วยความจำ และ ขนาด stack

พื้นที่ของชุดคำสั่งและพื้นที่ของข้อมูลแบบ static มีขนาดที่คงที่และหาค่าได้จากเมื่อโปรแกรมถูกสร้างด้วย linker แต่ในขณะที่โปรแกรมกำลังทำงานนั้นพื้นที่ heap และ stack สามารถเปลี่ยนแปลงได้ ในเครื่องคอมพิวเตอร์บางแบบที่มีฮาร์ดแวร์ จัดการด้านหน่วยความจำ



รูปที่ 2.16 โครงสร้างหน่วยความจำของ เทอร์โบซี - โมเดล small



รูปที่ 2.17 โครงสร้างหน่วยความจำแบบ Large

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถถูกนำมาใช้ปรับแต่งขนาดของมันในระหว่างที่มีการ execute ได้อย่างไรก็ดีในสภาพแวดล้อมของ MS-DOS บนเครื่องคอมพิวเตอร์แล้วนั้น เมื่อโปรแกรมเริ่มการ execute จะกำหนดให้ตำแหน่งของ stack และ heap คงที่ในรูปที่ 2.15 ถ้าโปรแกรมบังเอิญใช้จำนวน stack มากเกินกว่าขอบเขตของ stack ที่ให้ไว้จะส่งผลให้โปรแกรมอาจเขียนทับพื้นที่ข้อมูลแห่งอื่นหรืออาจจะเขียนทับตัวมันเอง เพราะฮาร์ดแวร์ของคอมพิวเตอร์ไม่ได้ใส่ความสามารถในการตรวจสอบการ overflow ของ stack แบบอัตโนมัติ แต่ในส่วนของคอมพิวเตอร์ซีนั้นสามารถที่จะเพิ่มเติมการตรวจสอบ stack ไว้ในโปรแกรมเพื่อคอยติดตาม stack ใดๆ ที่สร้างปัญหาการ overflow ด้วยไมโครซอฟท์ซีและ Quick C นั้นสามารถระบุขนาด stack เพื่อที่จะใช้ให้พอเพียงกับโปรแกรมได้

เทอร์โบซี ใช้โครงสร้างหน่วยความจำที่ต่างกันโมเดล small ดังรูปที่ 2.16 stack ในแบบนี้จะถูกวางไว้บนสุดของพื้นที่ข้อมูล 64 กิโลไบต์จะไม่เกิดการ overflow ของ stack เลยจนกว่า stack และ heap จะโตมาชนกัน ดังนั้นเมื่อใช้เทอร์โบซีในโมเดล small ก็ไม่จำเป็นต้องไปกังวลเกี่ยวกับขนาดของ stack

รูปที่ 2.17 แสดงโครงสร้างหน่วยความจำที่ใช้กับทั้ง ไมโครซอฟท์ซีและเทอร์โบซี ในโมเดลแบบ Large ซึ่งแบบนี้ต้องระบุขนาดของ stack ตามความเหมาะสม

2.11 CRC

CRC เป็นย่อมาจากคำว่า Cyclical Redundancy Check ซึ่งเป็นสูตรการคำนวณหาจากโพลิโนเมียล สูตรการคำนวณหา CRC ได้ถูกคิดค้นขึ้นโดยองค์กร CCITT โดยเริ่มแรกนั้น CITT นำ CRC มาใช้กับโปรโตคอลสำหรับสื่อสารข้อมูล เช่น HDLC หรือ ZMODEM ในปัจจุบันการใช้ CRC เพื่อตรวจสอบความถูกต้องของข้อมูลได้รับความนิยมสูงขึ้นเนื่องจากเป็นวิธีที่ดีและรวดเร็ว

การคำนวณ CRC ใช้วิธีการที่เรียกว่า "polynomial division" หรือ การหารโพลิโนเมียล โดยการนำแต่ละบิตในข้อมูลซึ่งไม่จำกัดความยาวของข้อมูล มาใช้เป็นค่าประสิทธิ์หรือ coefficient ของโพลิโนเมียล ซึ่งจะเรียกโพลิโนเมียลนี้ว่า message polynomial ในการหาค่า CRC จะต้องใช้โพลิโนเมียลชุดหนึ่งที่เรียกว่า generater polynomial มาหาร message polynomial ซึ่งจะได้ผลลัพธ์และเศษของการหาร การหารโพลิโนเมียลที่ว่านี้ จะใช้การหารที่ เรียกว่า modula-2 ซึ่งจะแทนการลบด้วย XOR และไม่มีการยืมเหมือนการลบ เศษของการนี้จะให้เป็น CRC

generater polynomial ที่ใช้ในการหาค่า CRC นั้นจะใช้โพลิโนเมียลใดๆ ก็ได้ ตามแต่ต้องการ แต่โดยทั่วไปมักจะใช้ค่าที่ CCITT กำหนดขึ้นมา ถ้าเป็นการคำนวณหา CRC ขนาด 16 บิต จะนิยมใช้โพลิโนเมียล

2.11.1 คุณภาพของ CRC-32

โปรแกรมตัวอย่างโปรแกรมหนึ่งชื่อ CRC-MAN โดย CRCMAN จะเป็นโปรแกรมที่คำนวณค่า CRC-32 หรือค่า CRC ขนาด 32 บิต ของไฟล์เก็บไว้ในไฟล์หนึ่ง ซึ่งสามารถนำกลับมาตรวจสอบได้ภายหลังว่า ไฟล์แต่ละไฟล์มีการเปลี่ยนแปลงอย่างไรบ้าง สาเหตุที่ใช้ CRC-32 ใช้จำนวนบิตถึง 32 บิต ซึ่งจะมีค่า CRC ที่เป็นไปได้ทั้งหมด 232 หรือเท่ากับ 4,294,967,296 ซึ่งโอกาสที่ไฟล์เปลี่ยนแปลงแก้ไขแล้วค่า CRC จะมีค่าเท่าเดิมนั้นแทบจะเป็นไปไม่ได้เลย นอกเหนือจากข้อดีดังกล่าวนี้ CRC-32 ยังมีข้อดีอื่นๆ อีก

2.11.1.1 บิตทุกบิตในข้อมูลจะเป็นส่วนทำให้เกิดค่า CRC ซึ่งก็หมายความว่า การเปลี่ยนแปลงในบิตใดๆ ของข้อมูลจะทำให้ค่า CRC เปลี่ยนแปลงไป ลองเปรียบเทียบกับค่า Checksum ในกรณีของข้อมูลที่เป็น 0FE10h กับ 10FEh จะเห็นว่าค่า Checksum จะเท่ากันทั้งสองกรณี แต่สำหรับ CRC แล้ว ข้อมูลสองชุดนี้จะให้ค่า CRC ต่างกัน เนื่องจากแต่ละบิตของข้อมูลต่างกันสลับไปมา ข้อมูลที่จะให้ค่า CRC เท่ากัน ก็คือข้อมูลแต่ละบิตมีค่าเท่ากันหรือข้อมูลเดียวกัน

2.11.1.2 การเปลี่ยนแปลงเพียงเล็กน้อยของข้อมูลก็มีผลให้ค่า CRC เปลี่ยนแปลงไปข้อมูลนี้เป็นผลโดยตรงจากข้อแรก

2.11.1.3 ฮิสโตแกรม (histogram) ของค่า CRC ต่อข้อมูลใดจะมีลักษณะแบบราบ



นั่นคือโอกาสที่จะเกิดค่า CRC ทุกข้อที่กล่าวมาคงจะเห็นว่า 00000000h ถึง 0FFFFFFFFh นั้นมีค่าเท่ากันหมด

จากคุณสมบัติของ CRC ทุกข้อที่กล่าวมาคงเห็นว่าโอกาสที่มีการแก้ไขข้อมูลในที่นี้สนใจก็คือข้อมูลที่เก็บในไฟล์โดยที่ค่า CRC ไม่เปลี่ยนแปลงนั้นแทบจะเป็นไปไม่ได้เลย ดังนั้นจึงไม่มั่นใจได้ว่า การใช้ CRC-32 ในโปรแกรมทำให้ตรวจพบการเปลี่ยนแปลงแก้ไขไฟล์ได้แน่นอนไม่มีพลาดเลยครั้งเดียว

ข้อดีอีกอย่างหนึ่งในการใช้ CRC ในการตรวจสอบการเปลี่ยนแปลงของไฟล์คือ CRC ไม่สามารถคำนวณกลับได้ ซึ่งเป็นผลดีอย่างมากในกรณีที่ผู้ใช้ตรวจสอบการเปลี่ยนแปลงจากการคิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไวรัส ยกตัวอย่างเช่น ไฟล์ COMMAND.COM ซึ่งสมมุติว่ามีค่า CRC-32 เท่ากับ 02F8690Ch เมื่อไวรัสพยายามแพร่ตัวเองให้ไปติดกับไฟล์นี้ และพยายามที่จะเพิ่มข้อมูลอาจเป็นหนึ่งหรือสองไบต์ เข้าไปเพื่อทำให้ค่า CRC มีค่าเท่าเดิมโดยอาศัยค่าคำนวณจากค่า CRC เดิมนั้นจะไม่สามารถทำได้เลย ซึ่งผิดกับกรณีของค่า Checksum ในกรณีของ Checksum นั้นสามารถคำนวณกลับได้ โดยเพิ่มไบต์ที่มีค่าเท่ากับผลต่างของ Checksum ก่อนและหลังแก้ไขไฟล์ ถ้าในกรณีของ CRC นั้น จะไม่สามารถแปลงกลับได้เลย แต่โชคร้ายตรงที่ว่าถ้าไวรัสใช้วิธีสุ่มข้อมูลขึ้นมาแล้วคำนวณค่า CRC จนกว่าจะได้ค่าเดิม ก็ไม่สามารถช่วยได้ ยังมีทางแก้ไขคือ เพิ่มการเก็บขนาดของไฟล์เข้าไปด้วยในโปรแกรม

2.11.2 CRCMAN

โปรแกรมตัวอย่างที่เขียนขึ้นมานี้มีชื่อว่า CRCMAN ซึ่งเขียนมาให้ทำงานสองลักษณะคือ สร้างไฟล์ที่ทันทาหน้าที่เก็บ CRC-32 ของไฟล์อื่นและตรวจสอบว่า CRC-32 ที่เก็บในไฟล์กับค่าที่คำนวณได้จริง หรือการตรวจสอบการเปลี่ยนแปลงแก้ไขไฟล์ คำสั่งที่ใช้เพื่อให้ CRCMAN ทำงานในแบบแรก คือ

```
CRCMAN-b directory crc-filename
```

ซึ่งจะทำให้ CRCMAN สร้างไฟล์ crc-filename และเก็บค่า CRC-32 ของไฟล์ทุกๆ ไฟล์ในไดเรกทอรีรวมทั้งไฟล์ในซับไดเรกทอรีไว้ใน crc-filename ไฟล์ crc-filename ที่ถูกสร้างขึ้นโดย CRCMAN นั้นจะเป็นเท็กไฟล์ธรรมดา โดยจะเก็บค่า CRC-32 พร้อมทั้งชื่อไฟล์นั้นยกตัวอย่างเช่น

```
CRCMAN-b C:\CRCFILETEST.CRC
```

ซึ่งจะมีผลให้ CRCMAN สร้างไฟล์ TEST.CRC ขึ้นที่ C:\CRCFILE เพื่อเก็บข้อมูล RC-32 และชื่อไฟล์ใน directory หรือ directory ปัจจุบัน โดยมีลักษณะดังตัวอย่างที่ 1

```
d5766938 C:\DOS\MONOUMB.386
```

```
10687a67 C:\DOS\VFINTD.386
```

```
199f61d4 C:\DOS\DBLSPACE.BIN
```

```
35b7232a C:\DOS\CHOICE.COM
```

```
9c23b6a0 C:\DOS\COMMAND.COM
```

```
e3550aa7 C:\DOS\DOSKEY.COM
```

```
30376cb1 C:\DOS\EDIT.COM
```

```
4486fa70 C:\DOS\FORMAT.COM
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5d6f7e7a C:\DOS\HELP.COM
 28693de3 C:\DOS\LOADFIX.COM
 fe70f51 C:\DOS\MODE.COM
 fb9cc476 C:\DOS\MORE.COM
 7370c941 C:\DOS\SYS.COM
 da8091ea C:\DOS\UNFORMAT.COM

ตัวอย่างที่ 1

สำหรับการทำงานในแบบที่สอง คือทำหน้าที่ตรวจสอบค่า CRC-32 ของไฟล์นั้น RCMAN ก็จะอ่านข้อมูลที่เก็บในไฟล์ที่สร้างจากการทำงานในแบบแรกมาแล้วทำการคำนวณค่า CRC-32 ของไฟล์ที่มีชื่อเก็บในไฟล์ข้อมูลนั้น รวมทั้ง CRC-32 เดิมเพื่อเปรียบเทียบอีกครั้ง โดยมีลักษณะการทำงานดังตัวอย่างที่สอง ซึ่งถ้ามีไฟล์ที่มีการเปลี่ยนแปลงก็จะเตือน

2.1.1.3 อธิบายโปรแกรม

โปรแกรม RCMAN สามารถคอมไพล์ได้จากคอมไพเลอร์ภาษาซี เกือบทุกคอมไพเลอร์แต่ต้องบนระบบปฏิบัติการคอนันั้น การทำงานของโปรแกรมก็เริ่มจากฟังก์ชัน main ซึ่งเริ่มแรกจะเป็นการ initial ตารางที่ใช้สำหรับการคำนวณ CRC-32 ซึ่งก็คือฟังก์ชัน Build CRCTable โดยการ initial ค่าเก็บในตาราง CRCTable [] หลังจากนั้นก็จะเป็นการตรวจสอบว่า ผู้ใช้งานเรียก RCMAN ให้ทำงานในแบบไหน ในสองแบบที่กล่าวมา โดยตรวจสอบจากจำนวนพารามิเตอร์ที่ผ่านมาทางคอมมานด์ไลน์ โดยถ้า argc เท่ากับ 2 ก็จะไปเรียกฟังก์ชัน

File	Expected	Got
C:\DOS\MONOUMB.386	d5766938	d5766938
C:\DOS\VFINTD.386	10687a67	10687a67
C:\DOS\DBLSPACE.BIN	199f61d4	199f61d4
C:\DOS\CHOICE.COM	35b7232a	35b7232a
C:\DOS\COMMAND.COM	9c23b6a0	9c23b6a0
C:\DOS\DOSKEY.COM	e3550aa7	e3550aa7
C:\DOS\EDIT.COM	30376cb1	30376cb1
C:\DOS\FORMAT.COM	4486fa70	4486fa70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C:\DOS\HELP.COM	5d6f7e7a	5d6f7e7a
C:\DOS\LOADFIX.COM	28693de3	28693de3

Press any key to continue.....

ตัวอย่างที่ 2

CheckFile() ซึ่งเป็นการทำงานในโหมดที่มีการตรวจสอบไฟล์ ถ้า arge เท่ากับ 4 แสดงว่าเป็นการทำงานในโหมดที่มีการสร้างไฟล์เก็บ CRC ซึ่งก็จะไปเรียกฟังก์ชัน BuildCRCFile () แต่ถ้า agc เป็นค่าอื่นๆ นอกเหนือจาก 2 และ 4 ก็จะพิมพ์ข้อความแสดงการใช้งาน CRCMAN ออกมาแล้วจบโปรแกรม

2.11.4 การสร้างไฟล์เก็บ CRC

ในการทำงานทั้งสองโหมดของ CRCMAN โหมดที่เขียนโปรแกรมยากที่สุดคือ โหมดที่มีการสร้างไฟล์เก็บ CRC เพราะต้องมีการคำนวณ CRC จากไฟล์ และเก็บลงในไฟล์เก็บ CRC นอกจากนี้ยังต้องสแกนหาไฟล์ในซับไดเรกทอรีย่อยอีกด้วย

ในการสร้างไฟล์เก็บ CRC ฟังก์ชัน BuildCRCFile () จะทำหน้าที่เปิดไฟล์เก็บ CRC ที่ังไว้ แล้วไปเรียกฟังก์ชัน ProcessAllFiles ซึ่งมีอัลกอริทึมดังนี้

```

ProcessAllFile (Path)
    dir = OpenDirectory (Path)
    while FileLeftInDirectory (dir)
        filename = GetNextFile (dir)
        if filename is a directory then
            ProcessAllFiles (Filename)
        else
            crc = CalculateCRC (filename)
            write filename and crc
        endif
    end of while
end of ProcessAllFiles

```

ในฟังก์ชัน PorcessAllFiles () ในลิสตติง ถ้าวัดองไต้ซอร์สดูจะเห็นฟังก์ชันที่ใช้การคำนวณ CRC ของไฟล์ คือฟังก์ชัน CalculateCRCFile () ฟังก์ชันนี้ใช้คำนวณ CRC แบบ 32 bit หรือ CRC-32 โดยวิธีเปิดตาราง และเก็บค่า CRC-32 และชื่อไฟล์ลงในไฟล์เก็บ CRC

2.11.5 การคำนวณ CRC-32 ในโปรแกรม

การคำนวณค่า CRC-32 ของไฟล์นั้นเป็นเรื่องง่ายๆ ซึ่งในฟังก์ชัน CalculteCRC () จะใช้การอ่านไฟล์ขึ้นมาทีละบล็อก ขนาดของบล็อกเท่ากับ 16K ซึ่งสามารถแก้ค่าใหม่ได้จากบรรทัดที่ใช้ preprocessor คือ

```
#define BUFFSIZE 16384U
```

แล้วจึงส่งให้ฟังก์ชัน CalculateBufferCRC () จะทำงานโดยเริ่มจากการ initial ค่า CRC เป็น 0 FFFFFFFFh เสียก่อนหลังจากคำนวณค่า CRC เสร็จ ก็ XOR ค่า CRC กลับด้วย 0FFFFFFFh อีกครั้ง การทำเช่นนี้เป็นการ coding แบบหนึ่งเพื่อเพิ่ม SNR (Signal-to-noise Ratio) ซึ่งมีใช้ในโปรโตคอลสื่อสารทั่วไป เช่น ZMODEM แม้กระทั่งโปรแกรมคั่งๆ อย่าง PKZIP และ ARJ

2.11.6 การตรวจสอบไฟล์

การทำงานอีกโหมคหนึ่งของ CRCMAN คือโหมคตรวจสอบไฟล์ซึ่งใช้ฟังก์ชัน CheckFILE () ซึ่งไม่ต้องสแกนหาไฟล์ในซิปใดเรียกทอริเนื่องจากเก็บชื่อไฟล์รวมทั้ง path ของไฟล์ไว้ในไฟล์เก็บ CRC เรียบร้อยแล้ว เมื่อตอนสร้างฟังก์ชันนี้ จึงทำหน้าที่ง่ายๆ เพียงเปิดไฟล์และคำนวณ CRC-32 ของไฟล์เพื่อเปรียบเทียบกับค่าที่เก็บเอาไว้

2.11.7 การใช้โปรแกรม

วิธีใช้โปรแกรม CRCMAN โดยการใช้พารามิเตอร์ -b เพื่อสร้างไฟล์เก็บ CRC และไม่ต้องใส่พารามิเตอร์ -b เพื่อตรวจสอบไฟล์ การใช้โปรแกรม CRCMAN ก็คือ ตรวจสอบการติดไวรัสของโปรแกรมไม่สนใจว่าไวรัสที่ไฟล์ คือไวรัสชนิดใดวิธีการทำ batch file ตรวจจับ

1. อยู่ใน root drive เช่น c:\ ที่มี file CRCMAN.EXE
2. CRCMAN -b c:\sub1\chkvirus.crc (ไปเช็คที่ sub1 และได้ไฟล์ chkvirus.crc ที่ root)
3. นำไฟล์ chkvirus.crc ที่ root ไปเก็บที่ c:\sub1
4. สามารถตรวจ virus ได้เมื่ออยู่ใน sub1 โดยพิมพ์ CRCMAN chkvirus.crc (ต้องมีไฟล์ CRCMAN.EXE อยู่ใน c:\sub1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12 สอนภาษาซีให้รู้จักภาษาไทย

ในเรื่องที่จะได้ทำความเข้าใจต่อไปนี้จะเป็นการใช้คำสั่ง 2 คำสั่งคือ คำสั่งในการกำหนดรูปแบบของเส้นตรงที่ใช้ควบคู่กับคำสั่ง `line` ด้วยคำสั่งนี้จะได้รับรูปแบบของเส้นเป็นไปตามความต้องการของเราได้ และ `rectangle` ซึ่งเป็นการวาดรูปสี่เหลี่ยม โดยคำสั่งแรกมีรูปแบบดังนี้

```
setlinestyle { int linestyle, unsigned upattern, int thickness };
```

เมื่อ `linestyle` คือ ลักษณะของเส้นมีค่าดังนี้

0 = SolidLn เส้นตรงที่บดตลอด

1 = DottedLn

2 = CenterLn เส้นประ 3 แบบ

3 = DashedLn

4 = UserBintLn (สำหรับให้ผู้ใช้เขียนโปรแกรมกำหนดลักษณะเส้นตรงได้เอง)

`upattern` รหัสกำหนดลักษณะของเส้นตรงมีผลเฉพาะการกำหนด `Linestyle` หมายเลข 4 นั้น (ถ้ากำหนด `Linestyle` เป็น 0-3 จะไม่มีผลแต่ประการใด) โดยกำหนดค่าได้ตั้งแต่ 0-65535 (ฐานสิบ) หรือ 0000-FFFF (ฐานสิบหก) คือเป็นตัวแปรชนิด `unsigned int` ขนาด 2 ไบต์ แต่จะกำหนดด้วยค่าขนาด 1 ไบต์ก็ได้

`thickness` ขนาดของเส้นซึ่งมี 2 ค่า คือ

1 = Norm Width ได้เส้นตรงขนาดปกติ

2 = Tick Width ได้เส้นตรงที่มีความหนาขึ้น

การใช้งาน

เช่น `setlinestyle (1,255,1)`; กรณีนี้ค่า 255 ไม่มีผลต่อลักษณะของเส้นตรงจึงจะกำหนดเป็นค่าอื่นก็ได้

`setlinestyle (4,255,1)`; กรณีนี้ค่า 255 มีผลต่อลักษณะของเส้นตรงจะได้เส้นมีลักษณะลงจุด 8 จุด แล้วเว้นไม่ลงจุด 8 จุดสลับกันไป

`setlinestyle(4,65534,1)`; ได้เส้นตรงแบบลงจุด 15 จุดเว้น 1 จุด ถ้าใช้คำสั่ง `setlinestyle (4,9+(9<<6)+(9<<11),1)`; จะได้เส้นตรงมีลักษณะดังนี้

0000000000001001 = 9

0000001001000000 = 9<<6

0100100000000000 = 9<<11

0100101001001001 = ผลที่ได้เมื่อรวมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$9 << 6$ คือ การเลื่อนบิตไปทางซ้าย 6 บิต นั่นคือบิตที่ 3 เปลี่ยนเป็นบิตที่ $3+6=9$ และบิตที่ 0 จะเป็นบิตที่ $0+6=6$ เป็นต้น ถ้าเลื่อนขวาก็คล้ายๆ กัน เช่น $225 >> 3$ จะได้ผลดังนี้

$11111111 = 255$

$00011111 = 255 >> 3 = 30$

คือ บิตที่ 7 จะเปลี่ยนเป็นบิตที่ $7-3=4$ หรือบิตที่ 2 จะเปลี่ยนเป็นบิตที่ $2-3=-1$ (ข้อมูลจึงหายไป) คำสั่งที่สองมีรูปแบบดังนี้

```
rectangle { int x1,int y1,int x2,int y2 } ;
```

เมื่อ $x1,y1$ คือ มุมบนซ้ายของรูปสี่เหลี่ยม

$x2,y2$ คือ มุมล่างขวาของรูปสี่เหลี่ยม

การใช้งานเช่น `rectangle(40,60,400,200);` เป็นต้น

2.12.1 สร้างคำสั่งใหม่

เพื่อให้สะดวกในการให้คำอธิบายในโปรแกรม จึงขอใช้เวลาในตอนนี้ใช้คำสั่ง `setlinestyle` สร้างอักขระภาษาไทย

คำสั่งใหม่ที่สร้างขึ้นมาใช้งานเพิ่มเติมมีดังต่อไปนี้

1. คำสั่งสำหรับลบจอภาพในขอบเขตที่ต้องการ มีรูปแบบดังนี้

```
clsgr(int x1,int y1,int x2,int y2)
```

เมื่อ

$x1,y1$ คือ มุมบนซ้ายของกรอบสี่เหลี่ยมลบจอภาพที่ต้องการลบ

$x2,y2$ คือ มุมล่างขวาของกรอบสี่เหลี่ยมลบจอภาพที่ต้องการลบ

2. คำสั่งสำหรับตรวจการกดแป้นซึ่งจะตรวจทั้งแป้นพิมพ์ปกติและแป้นฟังก์ชันทั้งหลาย คำสั่งนี้คือ `readchr ()`

โดยกำหนดให้ตัวแปร `ascii` แทนรหัสแอสกีของแป้นพิมพ์ปกติ และ `funckey` แทนแป้นฟังก์ชันการทำงานจะเป็นการตรวจสอบที่อินเตอร์รัพต์หมายเลข 16 (ฐานสิบ) โดยตรงซึ่งต้อง `#include <dos.h>` ด้วยคำสั่งที่ใช้คือ `geninterrupt (0x16)`

พร้อมชุดกับได้คัดแปลงชุดคำสั่งในตอน 1 คือ `opengraph ()`, `setcolor ()` และสร้างชุดคำสั่งการอ่านรหัสของตัวอักษรของชุดคำสั่ง `readfont ()` ขึ้นมาใช้งาน

2.18 ทฤษฎีภาษาอังกฤษ

2.18.1 วลี

วลี หมายถึงกลุ่มคำที่มีความหมาย แต่ไม่ได้ใจความสมบูรณ์เพราะขาดคำกริยาหรือภาคแสดง ถ้าเมื่อไรมีคำกริยากลุ่มคำเหล่านี้จะเป็นประโยคที่สมบูรณ์ ตัวอย่างเช่น

A beautiful girl

In the room

Is writing a letter

ทั้งสามกลุ่มคำนี้เป็น วลี ซึ่งเมื่อรวมกันเข้าเป็นประโยคคงต่อไปนี้จะได้ใจความสมบูรณ์

A beautiful girl is writing a letter in the room ผู้หญิงสวยนั่งเขียนจดหมายอยู่ในห้อง

2.18.2 วลีและอนุประโยคทำหน้าที่คล้ายคำวิเศษณ์

มีกลุ่มคำ หรือ วลี และอนุประโยคทำหน้าที่คล้ายคำวิเศษณ์ดังต่อไปนี้

2.18.2.1 วลีทำหน้าที่คล้ายคำวิเศษณ์ มีกลุ่มคำ หรือวลีทำหน้าที่คล้ายคำวิเศษณ์ คือ

ก. กลุ่มคำวิเศษณ์แสดงสถานที่ กลุ่มคำที่มีคำบุรพบท คำนาม อาจทำหน้าที่คล้ายคำวิเศษณ์แสดงสถานที่ เช่น

at home ที่บ้าน

by the window ใกล้หน้าต่าง

เช่นประโยคต่อไปนี้

He entered into the house. เขาเข้าไปในบ้าน

She is at home. เธออยู่ที่บ้าน

He sat by the window. เขานั่งใกล้หน้าต่าง

ข. กลุ่มคำวิเศษณ์แสดงกริยาทำทาง กลุ่มคำวิเศษณ์ที่มีบุรพบท เช่น by (โดย) และ with (ด้วย) และคำนาม ตามหลังจะทำหน้าที่คล้ายคำวิเศษณ์ แสดงกริยาทำทาง หรือวิธีการ เช่น

by plane โดยเครื่องบิน

by train โดยรถไฟ

นอกจากนี้ยังมีบุรพบทและคำนามอื่นๆ เช่น

in a hurry อย่างรีบร้อน อย่างรวดเร็ว

in a low voice อย่างค่อยๆค่อยๆ ด้วยเสียงต่ำ

เช่นประโยค

She traveled by plane. เธอเดินทางโดยเครื่องบิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

You can go there by bus. คุณไปที่นั่นได้ด้วยรถประจำทาง

ค. กลุ่มคำวิเศษณ์แสดงเวลาและแสดงบ่อย ๆ กลุ่มคำวิเศษณ์ที่มีบุรพบท เช่น in, on, at, during และคำนามเกี่ยวกับเวลา หรือกลุ่มคำแสดงความบ่อย จะทำหน้าที่คล้ายคำกริยาวิเศษณ์แสดงเวลา เช่นคำต่อไปนี้

in the morning ตอนเช้า ในตอนเช้า

at night ตอนกลางคืน ในตอนกลางคืน

เช่นประโยคต่อไปนี้

We see him every day. เราเห็นเขาทุกวัน

She comes once a year. เธอมาปีละหน

2.13.3 คำบุรพบท

คำบุรพบทเป็นอีกคำหนึ่งที่มีความสำคัญในภาษาอังกฤษ เพราะเป็นคำที่ไปเชื่อมคำอื่นในประโยคภาษาชัดเจน และสละสลวยขึ้น คำบุรพบทมีทั้งที่เป็นคำเดี่ยว หรือรวมเป็นกลุ่มกับคำอื่นๆ ดังนั้นในตอนนี้จะกล่าวถึง ความหมาย หน้าที่ ชนิด ประเภท และการใช้บุรพบทกับคำบางคำ ดังต่อไปนี้

2.13.3.1 ความหมายของคำบุรพบท

คำบุรพบทหมายถึงคำเชื่อมนามและสรรพนามให้เข้ากับประโยค ตัวอย่างที่เห็นบ่อยๆ ของคำบุรพบทได้แก่

on บน under ใต้

in ใน out of นอก

from จาก to ถึง

2.13.3.2 หน้าที่ของบุรพบท

มีหน้าที่ควบคุมคำนามและสรรพนามที่อยู่ตามหลัง เพื่อเชื่อมคำนามและสรรพนามเหล่านี้ให้เข้ากับประโยค และนามและสรรพนามที่ตามหลังคำบุรพบทจะเป็นกรรมของบุรพบทเสมอ ฉะนั้น อาจกล่าวได้ว่า บุรพบทมีหน้าที่เชื่อมกรรมให้เข้ากับประโยค ซึ่งกรรมนี้คือ คำนาม และคำแทนนาม คือ สรรพนาม และนามกริยา เช่นประโยคต่อไปนี้

He spoke to me. เขาได้พูดกับผม

I got a letter from her. ผมได้รับจดหมายจากเธอหนึ่งฉบับ

2.13.3.3 ชนิดของคำบุรพบท

คำบุรพบทอาจแบ่งได้เป็นสองชนิด คือคำบุรพบทที่เป็นคำเดียว หรือกลุ่มคำ (วลี) ดังต่อไปนี้

ก. คำบุรพบทที่เป็นคำเดียว ได้แก่คำบุรพบทที่เป็นคำเดียวโดดๆ ซึ่งทำหน้าที่เชื่อมกรรมให้เข้ากับประโยค เช่น

for	สำหรับ	at	ที่
in	ใน	on	บน
under	ใต้	by	โดย
through	ผ่าน	across	ข้าม

เช่นประโยคต่อไปนี้

This flower is for you. ดอกไม้นี้สำหรับคุณ

She is at the party. เธออยู่ในงานเลี้ยง

ข. คำบุรพบทที่เป็นกลุ่มคำ ได้แก่คำบุรพบทที่มีหลายคำอยู่เป็นหน่วยเดียวกัน ซึ่งทำหน้าที่เชื่อมกรรมให้เข้ากับประโยค เช่น

in front of	ข้างหน้า
in back of	ข้างหลัง
on top of	บน
according to	ตาม

เช่นประโยคต่อไปนี้

He parks his car in front of the building. เขาจอดรถไว้ข้างหน้าดี

2.13.3.4 ประเภทของคำบุรพบท

คำบุรพบทอาจแบ่งได้เป็น 4 ประเภท คือ

ก. บุรพบทแสดงสถานที่หรือตำแหน่งที่อยู่ ได้แก่คำบุรพบทที่แสดงให้ทราบตำแหน่งแหล่งที่ของสิ่งต่างๆ เช่น

on	บน	in	ใน
under	ใต้	below	ล่าง
near	ใกล้	by	ข้าง
before	ก่อน	after	หลัง

เช่นประโยคต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The doctor is now at the hospital. หมออยู่ที่โรงพยาบาลขณะนี้

ข. บุรพบทแสดงทิศทาง ได้แก่คำบุรพบทที่แสดงให้ทราบทิศทาง หรือวิธีการของสิ่ง หรือ การกระทำต่างๆ เช่น

in	ใน	into	ข้างใน
out of	ออกจาก	toward	ไปยัง ไปสู่ มาสู่
by way of	โดยทิศทาง	away from	หนีไปจาก

เช่นประโยคต่อไปนี้

They walked out of the hotel. เขาเดินออกมาจากโรงแรม

You can go by way of Chachoengsao to Pataya. คุณไปทางฉะเชิงเทราถึงพัทยา

ค. บุรพบทแสดงเวลา ได้แก่คำบุรพบทเกี่ยวกับเวลา ของเหตุการณ์ต่างๆ ที่เกิดขึ้น เช่น

in	ใน	at	ที่ ใน
around	ประมาณ	by	ประมาณ

เช่นประโยคต่อไปนี้

You should not visit him at the end of month. คุณไม่ควรไปพบเขาตอนปลายเดือน

ง. บุรพบทแสดงกริยาอาการ ได้แก่คำบุรพบทที่แสดงวิธีการ หรืออาการ ซึ่งไปขยายคำกริยา ของประธานในประโยค เช่น

by	โดย	on	โดย บน
in	ใน โดย ด้วย	with	กับ
like	เหมือน	as	เป็น
by	เป็น	for	เพื่อ

เช่นประโยคต่อไปนี้

We go by bus. เราไปโดยรถประจำทาง

We write with pens. เราเขียนด้วยปากกา

บทที่ 3

การออกแบบและการสร้าง

โปรแกรมนี้มีเป้าหมายที่จะนำเสนอผลงานให้น่าสนใจกว่าโปรแกรมที่มีลักษณะเช่นเดียวกัน จึงต้องเน้นทั้งทางด้านคุณภาพ ความสวยงามตามสมควร และความถูกต้องโดยในการออกแบบจะแบ่งออกเป็น 3 หมวด ใหญ่ ๆ คือ

3.1 ด้านเนื้อหา ภาษาอังกฤษได้ค้นคว้ามาจากตำราหลายเล่มและได้ทำการรวมข้อมูลจนถึงได้ทำการประยุกต์ หลักจิตวิทยา หลักการสอน การเรียนรู้ มาประยุกต์ใช้ให้เอื้อประโยชน์สูงสุดโดยยึดหลักการให้เร้าความสนใจต่อผู้พบเห็นให้มากที่สุดเท่าที่จะทำได้ นอกจากนี้จะต้องให้ประสิทธิภาพการเรียนสูงสุดสามารถเก็บบันทึกสถานะไว้เพื่อใช้เรียนในคราวต่อไป นอกจากนี้ยังจะต้องเพิ่มส่วนการทดสอบโดยประเมินผลงานจากการเล่นเกมส์

3.2 ด้านเทคนิควิธีการเขียนโปรแกรม โดยพยายามที่จะประหยัดหน่วยความจำให้มากที่สุดแต่ให้ได้เนื้องานมากที่สุดเช่นกัน สำหรับการวางแผนในการทำงานจะยึดหลักการดังนี้

3.2.1 ศึกษากลุ่มเป้าหมายที่จะใช้โปรแกรมนี้ซึ่งคาดว่าจะเป็นผู้ที่ทำงานในสำนักงานต่างๆ ซึ่งปัจจุบันนี้ได้ใช้ภาษาอังกฤษเป็นภาษากลางในการติดต่อ ซึ่งได้ข้อสรุปว่ากลุ่มที่จะใช้งานมากที่สุดคือกลุ่มวิศวกร

3.2.2 ออกแบบระบบจากการมองภาพรวมจนกระทั่งนำไปสู่ส่วนย่อย ๆ ซึ่งเรียกว่าการวางแผนระบบแบบจากบนลงล่าง(Top down design)

3.2.3 ดำเนินการเขียนโปรแกรมโดยใช้ภาษาซีเป็นภาษาโปรแกรม ซึ่งจะให้ความเร็วในการใช้งาน รวมไปถึงความอ่อนตัวในการปรับปรุงระบบ

3.2.4 ทดสอบความถูกต้อง โดยจะต้องพยายามสมมุติสถานะการจำลองในลักษณะต่างที่อาจจะมีโอกาสเกิดขึ้นและปกป้องความผิดพลาดทั้งหลายที่จะมีเข้าไป

3.2.5 นำออกทดลองใช้กับผู้ใช้คอมพิวเตอร์ในกลุ่มและระดับต่างๆ เพื่อดูปัญหาต่างๆ ที่เกิดขึ้น

3.3 ด้านงาน ฐานข้อมูลได้นำหลักการฐานข้อมูลเชิงสัมพันธ์มาประยุกต์ใช้ โดยมีอัลกอริทึมในการสืบค้นข้อมูลแบบไบนารีทรี ซึ่งจะให้ความเร็วสูงแม้ว่าอาจจะต้องใช้หน่วยความจำเพิ่มมากขึ้นกว่าการใช้อัลกอริทึมแบบลิงค์ลิสต์ การจัดการด้านฐานข้อมูลจะเริ่มด้วยการออกแบบพจนานุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล ซึ่งจะเป็นการกำหนดขนาดความยาวของฟิลด์ข้อมูลต่างๆ ที่ใช้ในโปรแกรม สำหรับชนิดข้อมูลในโปรแกรมนี้นแบ่งออกเป็น 3 แบบใหญ่ๆ

แบบ A ใช้กับข้อมูลที่เป็นระบบภาษาไทย

แบบ E เป็นแบบที่ใช้ข้อมูลภาษาอังกฤษ

แบบ Z ใช้กับข้อมูลประเภทตัวเลขที่เต็มศูนย์ข้างหน้า

ในการแบ่งชนิดเช่นนี้เพื่อป้องกันความผิดพลาดจากการพิมพ์ข้อมูลเข้าไปเก็บไว้ในระบบฐานข้อมูล ซึ่งง่ายต่อการตรวจสอบความผิดพลาดแม้จะมีผลเสียอยู่บ้างในด้านความอ่อนตัวในการใช้งานดังตัวอย่างข้างล่าง

```
; ----- ph_pvd.voc Phasal verb dictionary
```

```
; ----- data element dictionary
```

```
#schema PH_PVD
```

```
#dictionary
```

```
MAIN_WORDS,      E, 13, "-----"
PHASE_LIST,      B, 60, "-----"
MAIN2_WORDS,     E, 13, "-----"
PHAS2_LIST,      B, 60, "-----"
MAIN3_WORDS,     E, 13, "-----"
PHAS3_LIST,      B, 60, "-----"
PHASE_NO,        Z,  5, "-----"
PHASE_NAME,      E, 40, "-----"
PHASE_MEANS,     A, 45, "-----"
EXAM_ENG1,       E, 50, "-----"
EXAM_ENG2,       E, 20, "-----"
EXAM_THAI,       A, 70, "-----"
PHAS2_NO,        Z,  5, "-----"
PHAS2_NAME,      E, 40, "-----"
PHAS2_MEANS,     A, 45, "-----"
EXAM2_ENG1,      E, 50, "-----"
EXAM2_ENG2,      E, 20, "-----"
EXAM2_THAI,      A, 70, "-----"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PHAS3_NO, Z, 5, "_____"
 PHAS3_NAME, E, 40, "_____"
 PHAS3_MEANS, A, 45, "_____"
 EXAM3_ENG1, B, 50, "_____"
 EXAM3_ENG2, E, 20, "_____"
 EXAM3_THAI, A, 70, "_____"
 STUDENTS_NAME, A, 40, "_____"
 MAINWORDS_NO, Z, 5, "_____"
 #end dictionary

จากนั้นจะมีการออกแบบระบบไฟล์ของฐานข้อมูล ดังตัวอย่างข้างล่าง

; ---- file specifications

#file DICWORDS

MAIN_WORDS

PHASE_LIST

#end file

#file DICWORD2

MAIN2_WORDS

PHAS2_LIST

#end file

#file DICWORD3

MAIN3_WORDS

PHAS3_LIST

#end file

#file VOCPHASE

PHASE_NO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PHASE_NAME
 PHASE_MEANS
 EXAM_ENG1
 EXAM_ENG2
 EXAM_THAI
 #end file

#file VOCPHAS2
 PHAS2_NO
 PHAS2_NAME
 PHAS2_MEANS
 EXAM2_ENG1
 EXAM2_ENG2
 EXAM2_THAI
 #end file

#file VOCPHAS3
 PHAS3_NO
 PHAS3_NAME
 PHAS3_MEANS
 EXAM3_ENG1
 EXAM3_ENG2
 EXAM3_THAI
 #end file

#file STUDENTS
 STUDENTS_NAME
 MAINWORDS_NO
 #end file



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; ----- index specifications
#key DICWORDS MAIN_WORDS
#key DICWORD2 MAIN2_WORDS
#key DICWORD3 MAIN3_WORDS
#key VOCPHASE PHASE_NO
#key VOCPHAS2 PHAS2_NO
#key VOCPHAS3 PHAS3_NO
#key STUDENTS STUDENTS_NAME
#end schema

```

```

/* ----- PH_PVD ----- */

```

```

#define MAIN_WORDS 1
#define PHASE_LIST 2
#define MAIN2_WORDS 3
#define PHAS2_LIST 4
#define MAIN3_WORDS 5
#define PHAS3_LIST 6
#define PHASE_NO 7
#define PHASE_NAME 8
#define PHASE_MEANS 9
#define EXAM_ENG1 10
#define EXAM_ENG2 11
#define EXAM_THAI 12
#define PHAS2_NO 13
#define PHAS2_NAME 14
#define PHAS2_MEANS 15
#define EXAM2_ENG1 16
#define EXAM2_ENG2 17
#define EXAM2_THAI 18
#define PHAS3_NO 19

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define PHAS3_NAME 20
#define PHAS3_MEANS 21
#define EXAM3_ENG1 22
#define EXAM3_ENG2 23
#define EXAM3_THAI 24
#define STUDENTS_NAME 25
#define MAINWORDS_NO 26
#define DICWORDS 0
#define DICWORD2 1
#define DICWORD3 2
#define VOCPHASE 3
#define VOCPHAS2 4
#define VOCPHAS3 5
#define STUDENTS 6

struct dicwords
char main_words [14];
char phase_list [61];
;
struct dicword2
char main2_words [14];
char phas2_list [61];
;

struct dicword3
char main3_words [14];
char phas3_list [61];
;

struct vocphase
char phase_no [6];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char phase_name [41];
char phase_means [46];
char exam_eng1 [51];
char exam_eng2 [21];
char exam_thai [71];
;

```

```

struct vocphas2

```

```

char phas2_no [6];
char phas2_name [41];
char phas2_means [46];
char exam2_eng1 [51];
char exam2_eng2 [21];
char exam2_thai [71];
;

```

```

struct vocphas3

```

```

char phas3_no [6];
char phas3_name [41];
char phas3_means [46];
char exam3_eng1 [51];
char exam3_eng2 [21];
char exam3_thai [71];
;

```

```

struct students

```

```

char students_name [41];

```

```

char mainwords_no [6];
;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* ----- PH_PVD ----- */

int ellen [ ] = 3,60,13,60,13,60,5,40,45,50,20,70,5,40,45,50,20,70,5,40,45,50,20,70,40,5 ;
int f_dicwords [ ] =
MAIN_WORDS,
PHASE_LIST,
0
;

int f_dicword2 [ ] =
MAIN2_WORDS,
PHAS2_LIST,
0
;

int f_dicword3 [ ] =
MAIN3_WORDS,
PHAS3_LIST,
0
;

int f_vocphase [ ] =
PHASE_NO,
PHASE_NAME,
PHASE_MEANS,
EXAM_ENG1,
EXAM_ENG2,
EXAM_THAI,
0
;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int f_vocphas2 [ ] =
PHAS2_NO,
PHAS2_NAME,
PHAS2_MEANS,
EXAM2_ENG1,
EXAM2_ENG2,
EXAM2_THAI,
0
;

```

```

int f_vocphas3 [ ] =
PHAS3_NO,
PHAS3_NAME,
PHAS3_MEANS,
EXAM3_ENG1,
EXAM3_ENG2,
EXAM3_THAI,
0
;

```

```

int f_students [ ] =
STUDENTS_NAME,
MAINWORDS_NO,
0
;

```

```

int *file_ele [ ] =
f_dicwords,
f_dicword2,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
f_dicword3,
f_vocphase,
f_vocphas2,
f_vocphas3,
f_students,
0
;
```

```
int x1_dicwords [ ] =
MAIN_WORDS,
0
;
```

```
int *x_dicwords [ ] =
x1_dicwords,
0
;
```

```
int x1_dicword2 [ ] =
MAIN2_WORDS,
0
;
```

```
int *x_dicword2 [ ] =
x1_dicword2,
0
;
```

```
int x1_dicword3 [ ] =
MAIN3_WORDS,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0

;

```
int *x_dicword3 [ ] =
```

```
x1_dicword3,
```

0

;

```
int x1_vocphase [ ] =
```

```
PHASE_NO,
```

0

;

```
int *x_vocphase [ ] =
```

```
x1_vocphase,
```

0

;

```
int x1_vocphas2 [ ] =
```

```
PHAS2_NO,
```

0

;

```
int *x_vocphas2 [ ] =
```

```
x1_vocphas2,
```

0

;

```
int x1_vocphas3 [ ] =
```

```
PHAS3_NO,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0

;

int *x_vocphas3 [] =

x1_vocphas3,

0

;

int x1_students [] =

STUDENTS_NAME,

0

;

int *x_students [] =

x1_students,

0

;

int **index_ele [] =

x_dicwords,

x_dicword2,

x_dicword3,

x_vocphase,

x_vocphas2,

x_vocphas3,

x_students,

0

;

cd

del chk_vr.crc

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

copy c:.exe
copy c:_vrs2.pwa
cd
copy c:_vrs1.exe
ph_vrs1.exe
pause
cd
copy c:_vr.crc
pause
cd
dicphas2
cd
copy c:_vrs2.exe c:
cd
exe2bin ph_vrs2.exe ph_vrs2.com
copy ph_vrs2.com c:
cd
copy ph_vrs2.com *.pwa
cd
copy c:.exe
dbinit
copy c:.exe
ph_schem < ph_pvd.voc > ph_pvd.i1 -1
ph_schem < ph_pvd.voc > ph_pvd.i2 -2
ph_schem < ph_pvd.voc > ph_pvd.i3 -3
pause
copy ph_pvd.voc c:
copy ph_pvd.i* c:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การใช้งานและการทดลองโปรแกรม

4.1 ขั้นตอนการทำโปรแกรมโดยใช้เทอร์โบซี

เมื่อเขียนโปรแกรมในภาษาซีเสร็จเรียบร้อยแล้วเทอร์โบซีจะมีขั้นตอนการทำงาน เพื่อให้ได้ผลลัพธ์ออกมา 2 ขั้นตอน คือ

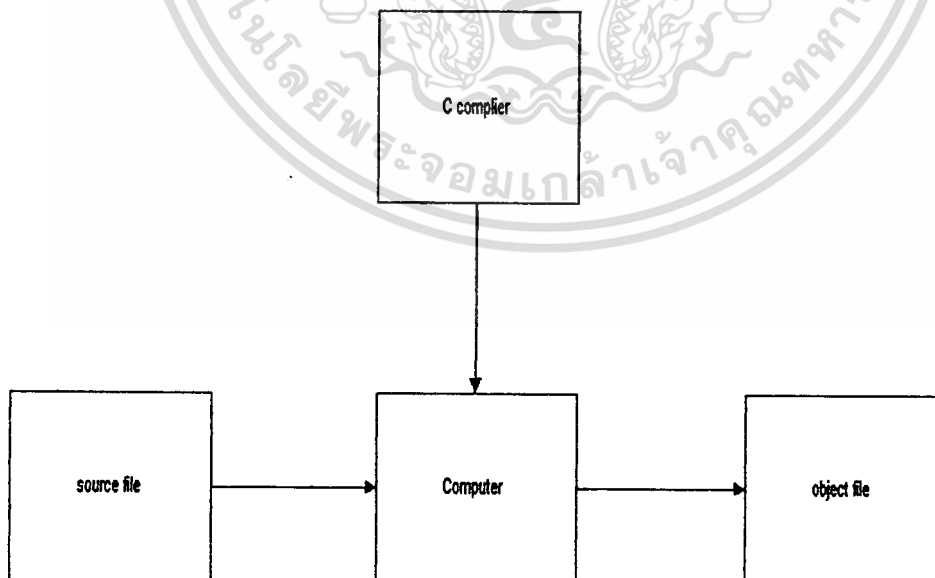
ขั้นตอนที่ 1 การแปลโปรแกรม (Compile)

เป็นการนำโปรแกรมที่เขียนขึ้น (source file) มาทำการแปลเป็นโปรแกรมภาษาเครื่อง (object file) ดังรูปที่ 4.1

source file เป็นโปรแกรมที่เขียนขึ้นซึ่งมีกฎเกณฑ์การตั้งชื่อเหมือนกับการตั้งชื่อแฟ้มข้อมูลใน DOS แต่จะมีชนิดของแฟ้มข้อมูลเป็น .C เท่านั้น เช่น TOP.C เมื่อโปรแกรมนี้ผ่านการแปลโปรแกรมแล้วจะได้เป็นโปรแกรมภาษาเครื่อง (object file) ที่มีชนิดของแฟ้มข้อมูลเป็น .OBJ เช่น TOP.OBJ

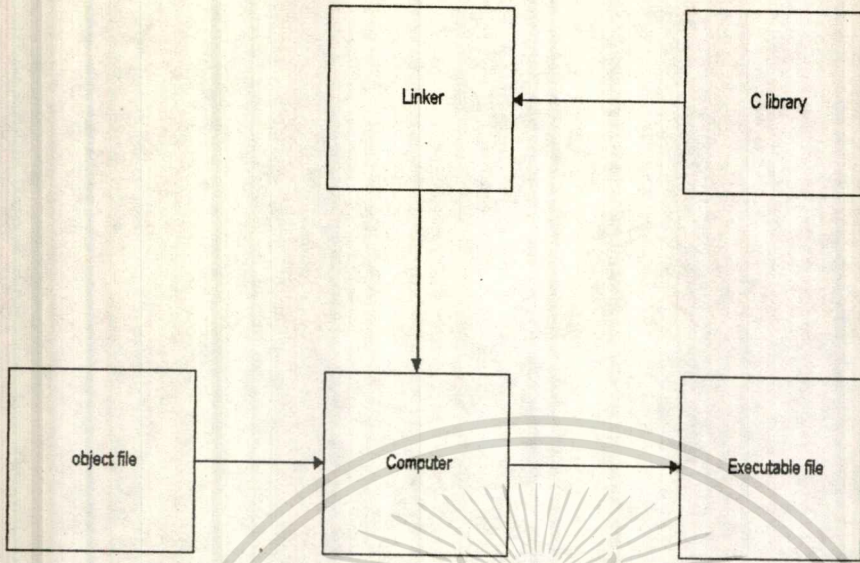
ขั้นตอนที่ 2 การเชื่อมโยง (Link)

เป็นการนำโปรแกรมภาษาเครื่อง (object file) ทำการเชื่อมโยงกับฟังก์ชันที่ต้องการจากไลบรารี เพื่อสร้างให้เป็น Executable file ดังรูป 4.2



รูปที่ 4.1 การแปลโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



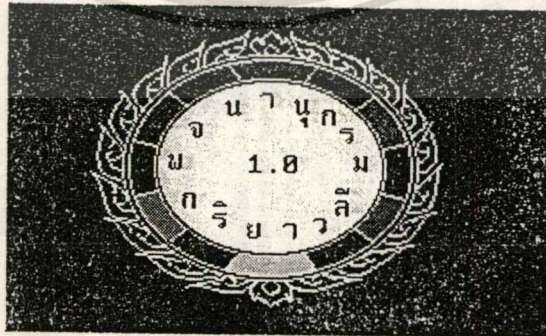
รูปที่ 4.2 การเชื่อมโยง

Executable file จะเป็นแฟ้มข้อมูลที่สามารถสั่งให้ปฏิบัติงานได้โดยตรงในขณะที่อยู่ใน DOS แฟ้มข้อมูลนี้จะเป็นชนิด .EXE

การใช้งานโปรแกรม พจนานุกรม กิริยาวลีและสะแลง

เมื่อติดตั้งโปรแกรมเรียบร้อยแล้ว สามารถใช้งาน โปรแกรมพจนานุกรมกิริยาวลีและสะแลง ในการค้นหา เรียนรู้ เพิ่มศัพท์ และทดสอบได้ โดยการพิมพ์คำสั่ง

C:/> phases



รูปที่ 4.3 โลโก้ของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กด Enter จะปรากฏโลโก้ของโปรแกรมพจนานุกรมกริยาวลีและแสงจอภาพปรากฏดังรูปที่ 4.3

ให้กดคีย์Enter (ในกรณีไม่เซตเมาส์) หรือคลิกเมาส์จะปรากฏเมนูหลักของโปรแกรมประกอบด้วย

- 1.Help
- 2.เงิน
- 3.ศัพท์
- 4.ซ่อมแซม
- 5.DOS
- 6.Exit

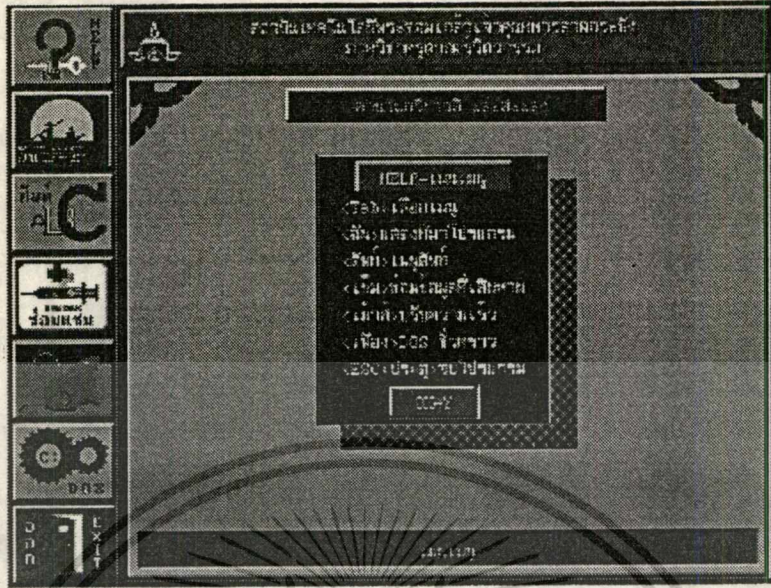
จอภาพปรากฏดังรูปที่ 4.4



รูปที่ 4.4 เมนูหลักของโปรแกรม

4.1 Help

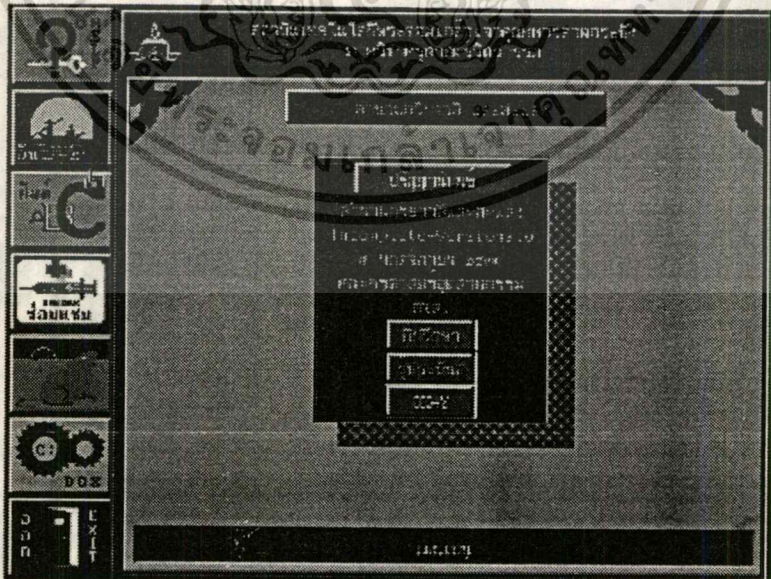
เป็นส่วนที่ทำหน้าที่ช่วยอธิบายความหมายของรูปเมนูหลักและหน้าที่การทำงานของปุ่มคีย์บางปุ่มบนคีย์บอร์ด ถ้าต้องการที่จะดู Help ก็ให้นำรูปเมาส์ไปคลิกที่ Help จอภาพออกมาปรากฏดังรูปที่ 4.5



รูปที่ 4.5 ส่วนช่วยเหลือของโปรแกรม

4.2 ฉันทน์

เป็นส่วนที่ทำหน้าที่อธิบายว่าปฏิญญาพันธสัญญาคืออะไร มีเมนูชื่อผู้ประพันธ์และอาจารย์ ผู้ให้คำปรึกษาคลิกเมาส์ที่ ฉันทน์ จอภาพปรากฏดังรูปที่ 4.6

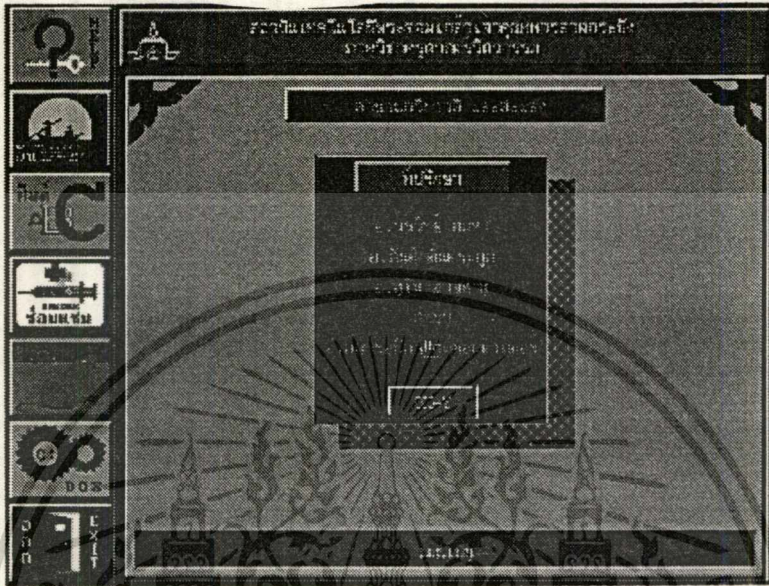


รูปที่ 4.6 เมนูในส่วนของฉัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

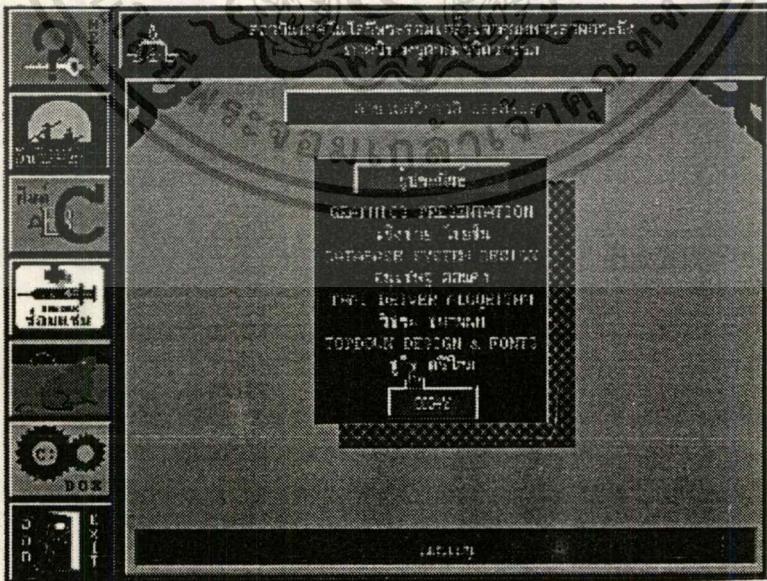
การใช้งานส่วนนี้

4.2.1 เมื่อผู้ใช้เลือกเมนูในส่วนของที่ปรึกษา จอภาพปรากฏดังรูปที่ 4.7



รูปที่ 4.7 แสดงชื่อของผู้ให้คำปรึกษาในการจัดทำโปรแกรม

4.2.2 เมื่อผู้ใช้เลือกเมนูในส่วนของผู้ประพันธ์ จอภาพปรากฏดังรูปที่ 4.8

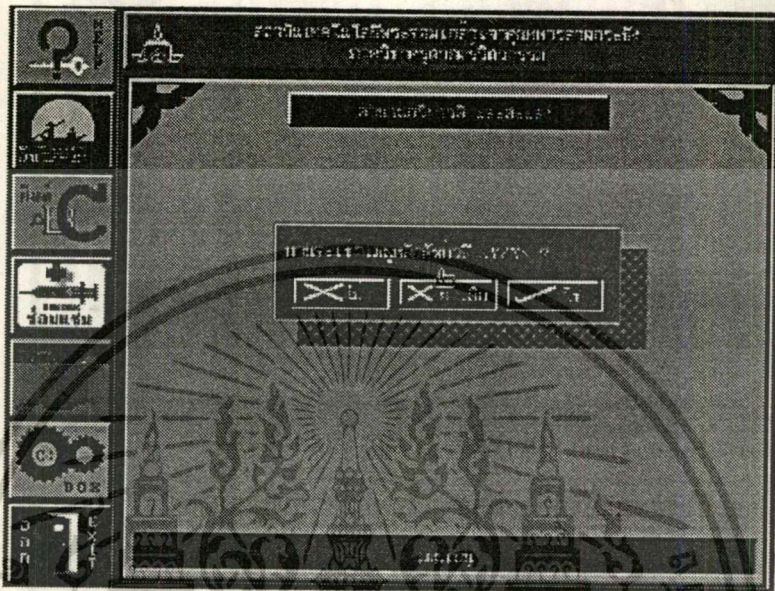


รูปที่ 4.8 แสดงชื่อและหน้าที่ของผู้จัดทำโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

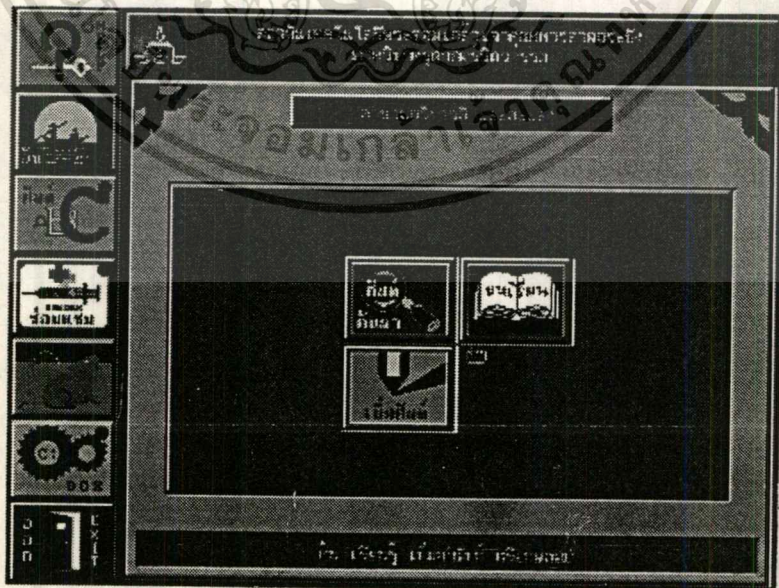
4.3 ศัพท์

เป็นส่วนที่ทำหน้าที่จัดการเกี่ยวกับการใช้งานที่เป็นส่วนที่เป็นการใช้งานหลักต่างๆ ของโปรแกรมพจนานุกรมกริยาวลีและสะแลง คลิกเมาท์ศัพท์ จอภาพปรากฏดังรูปที่ 4.9



รูปที่ 4.9 เริ่มต้นเข้าสู่เมนูของศัพท์

ถ้าต้องการใช้งานเกี่ยวกับเรื่องต่างๆ คลิกเมาท์ที่"ใช่" จอภาพปรากฏที่หน้าจอดังรูปที่ 4.10



รูปที่ 4.10 เมนูของศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในเมนูย่อยของเมนูศัพท์นี้จะประกอบด้วยเมนูต่อไปนี้

1. เพิ่มศัพท์
2. ค้นหาศัพท์
3. เรียนรู้
4. ทดสอบ

การใช้งานส่วนของศัพท์

4.3.1 การค้นหาจะเป็นส่วนที่ทำหน้าที่ค้นหาศัพท์ตามที่ใช้ต้องการเรียกใช้ คลิกเมาส์ที่ค้นหาปรากฏเมนูการใช้งานเพิ่มขึ้นมาใหม่จอภาพปรากฏดังรูปที่ 4.11 เมนูต่างๆ ดังรูปที่ 4.11 มีหน้าที่ดังต่อไปนี้

4.3.1.1 Help ทำหน้าที่อธิบายถึงหน้าที่ของปุ่มบางปุ่มบนคีย์บอร์ดที่ใช้ในการเลือกเมนูแทนเมาส์หรือทำงานอื่นๆ บางอย่าง

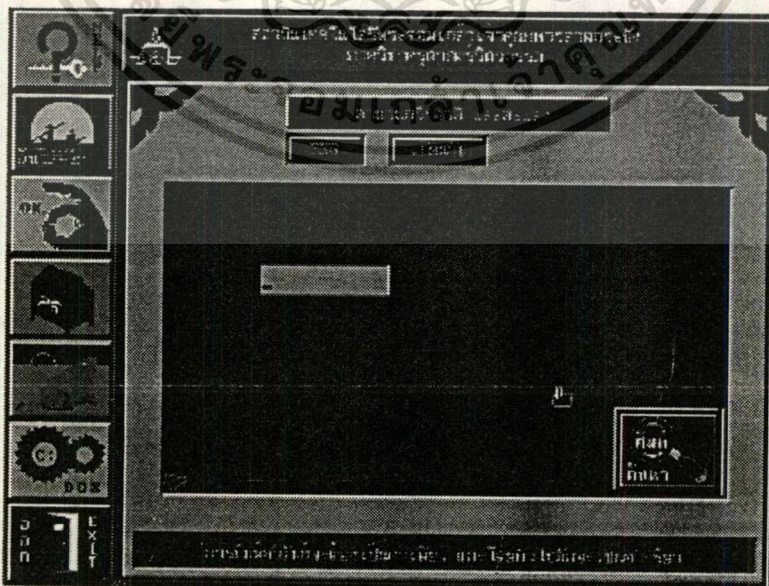
4.3.1.2 OK ตกลงจะใช้ค้นหาคำศัพท์หรือไม่

4.3.1.3 รูปถังขยะ ลบคำศัพท์ที่พิมพ์ผิดหรือไม่ต้องการทิ้ง(ทั้งหมด)

4.3.1.4 รูปหนู(mouse) ปรับความเร็วของเมาส์

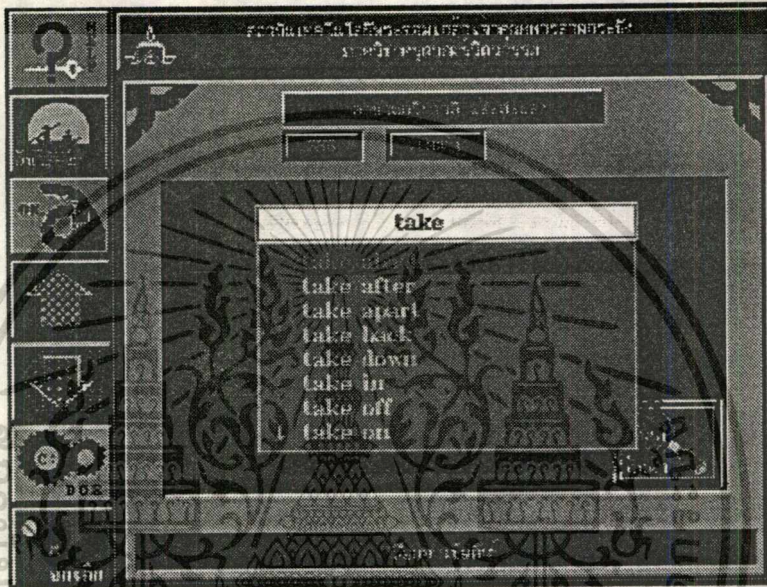
4.3.1.5 DOS ออกสู่ระบบคอสซัควคราว

4.3.1.6 Exit ออกจากโปรแกรมค้นหานี้กลับไปยังเมนูหลัก

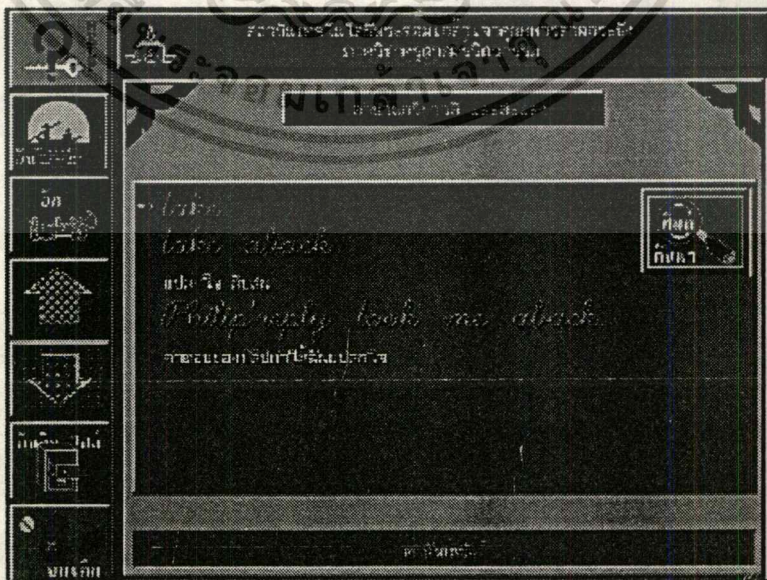


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 4.11 เมนูในส่วนของการค้นหาศัพท์ทำให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่เป็นตัวอักษรอังกฤษด้านล่างใช้ในกรณีที่ใช้ไม่ต้องการพิมพ์คำศัพท์ก็เลือกตัวอักษรเหล่านั้นได้โดยการนำเมาส์ไปคลิกที่ตัวอักษร ตัวอักษรตัวนั้นก็จะไปปรากฏที่สี่เหลี่ยมช่องว่างช่องเติมศัพท์ เมื่อผู้ใช้ต้องการค้นหาคำศัพท์ใดก็ได้เติมคำนั้น ซึ่งคำศัพท์ที่เติมนั้นจะต้องเป็นคำศัพท์หลักเท่านั้นคือ เป็นคำเดียวเมื่อเติมเสร็จแล้วคลิกเมาส์ที่ OK จอภาพปรากฏดังรูปที่ 4.12



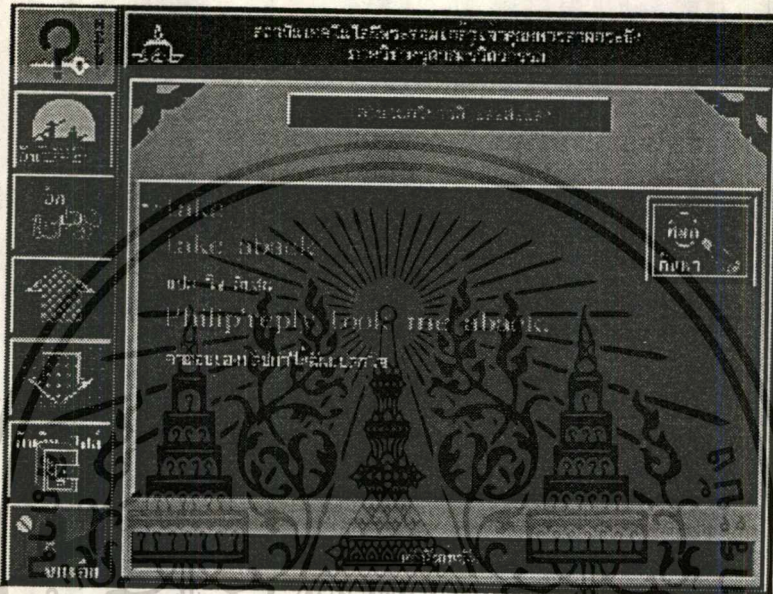
รูปที่ 4.12 ค้นหาศัพท์ในกรณีที่มีศัพท์อยู่ในโปรแกรม



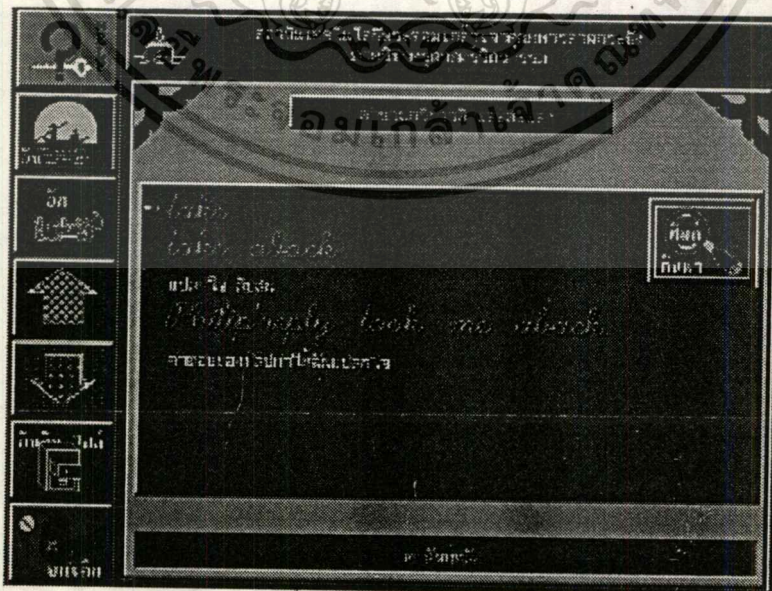
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ รูปที่ 4.13 อธิบายความหมายของศัพท์ ำดูให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวลูกศรชี้ขึ้นบนและลงล่าง ทำหน้าที่เลือกค้นหาคำศัพท์ขึ้นไปทางด้านบนและลงล่าง
เมื่อพบศัพท์ที่ต้องการตอบ OK จอภาพปรากฏดังรูปที่ 4.13

อักษรเป็นเมนูที่เลือกรูปแบบตัวอักษรได้ 4 รูปแบบ โดยการคลิกเมาส์ที่อักษรรูปแบบที่ 1
จอภาพปรากฏดังรูปที่ 4.14



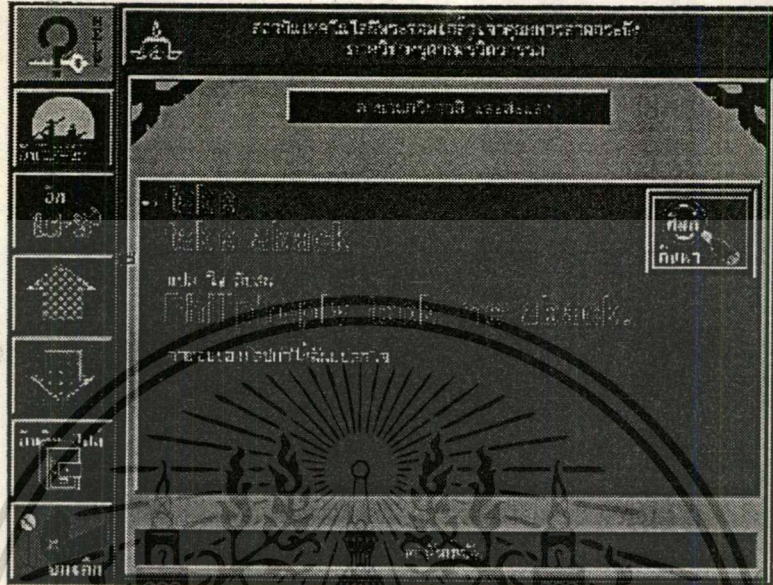
รูปที่ 4.14 ตัวอักษรแบบมาตรฐาน



รูปที่ 4.15 ตัวอักษรแบบเอียง

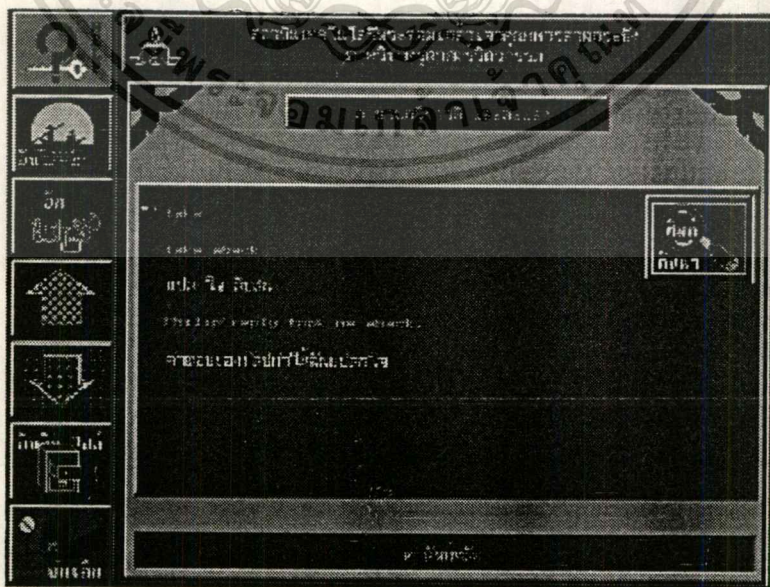
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบที่ 2 จอภาพปรากฏดังรูปที่ 4.15 และ รูปแบบที่ 3 จอภาพปรากฏดังรูปที่ 4.16



รูปที่ 4.16 ตัวอักษรแบบสองชั้น

รูปแบบที่ 4 เป็นรูปแบบของอักษรตัวเล็กมีลักษณะเหมือนกับรูปแบบอักษรตัวใหญ่ จอภาพปรากฏดังรูปที่ 4.17



รูปที่ 4.17 ตัวอักษรแบบตัวเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานในท้องถิ่นเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

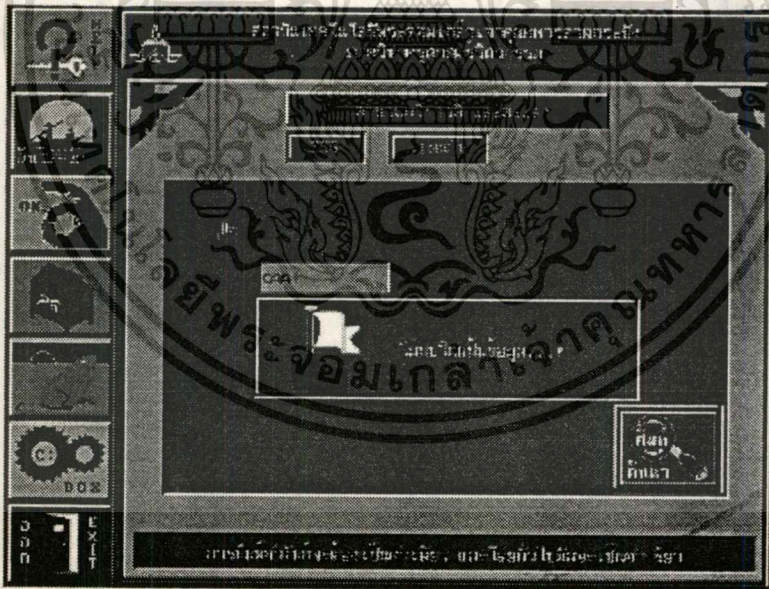
ตัวลูกศรชี้ขึ้นบน ทำหน้าที่เลื่อนตำแหน่งขึ้นไปทางด้านบนบอกว่าคำในบรรทัด นั้นทำหน้าที่เป็นอะไร

ตัวลูกศรชี้ลงล่าง ทำหน้าที่เลื่อนตำแหน่งลงฝทางด้านล่างบอกว่าคำในบรรทัดนั้นทำหน้าที่เป็นอะไร

มีอยู่ 5 ประเภทด้วยกัน

1. คำศัพท์หลัก
2. ส่วนนกริยาวิลีหรือสะแลง
3. ความหมายทั่วไป
4. ตัวอย่างการใช้งาน
5. คำแปล

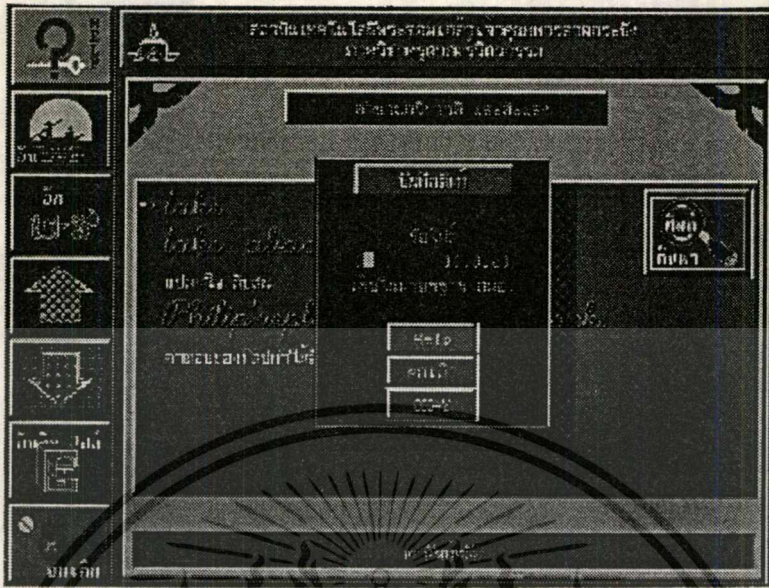
ในกรณีคำศัพท์ที่ต้องการค้นหา นั้นไม่มีในโปรแกรม เมื่อคลิกเมาส์ที่ค้นหาภาพปรากฏ
 ดังรูปที่ 4.18



รูปที่ 4.18 ค้นหาศัพท์ในกรณีศัพท์ในโปรแกรมไม่มี

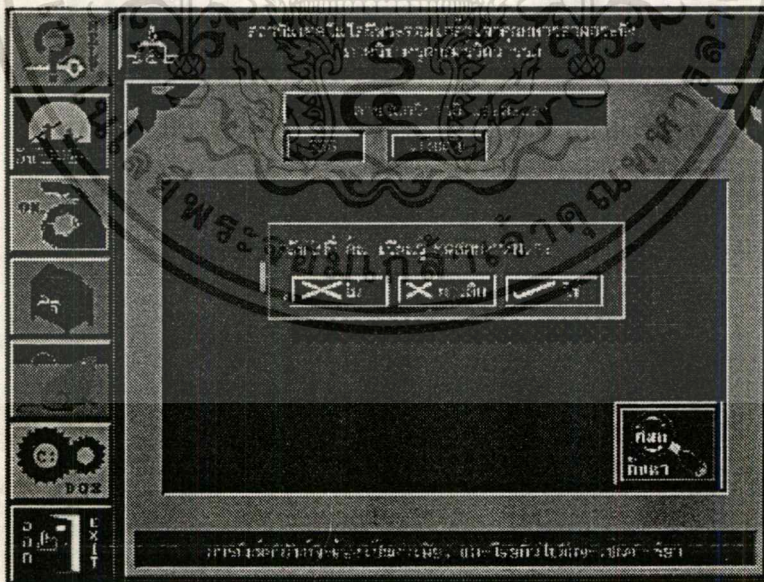
เมื่อจอภาพปรากฏดังรูปที่ 4.18 กด ESC โปรแกรมกลับสู่เมนูค้นหาและที่ดูบันทึกไฟล์ ทำหน้าที่เก็บข้อมูลที่ต้องการ จอภาพปรากฏดังรูปที่ 4.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 บันทึกไฟล์คำสั่ง

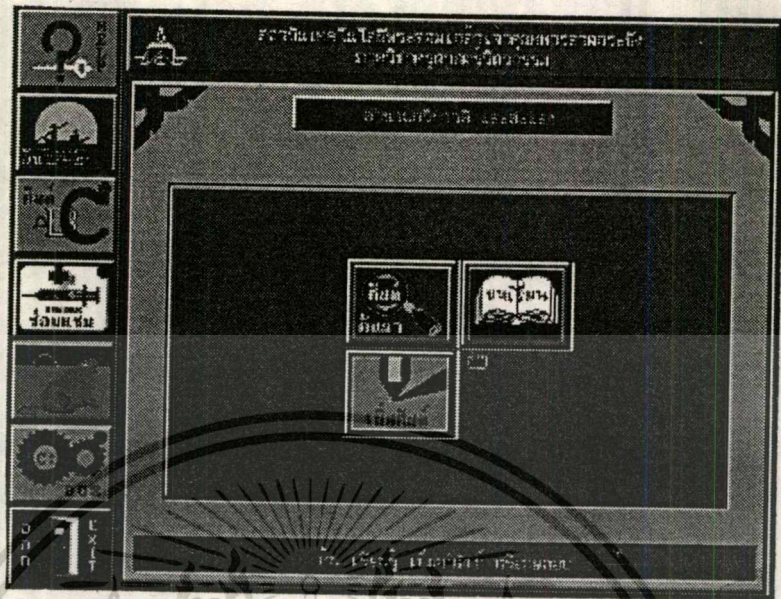
ออกจากเมนูค้นหา คลิกเมาส์ที่ EXIT จอภาพปรากฏดังรูปที่ 4.20



รูปที่ 4.20 ต้องการออกจากเมนูค้นหาคำสั่งหรือไม่

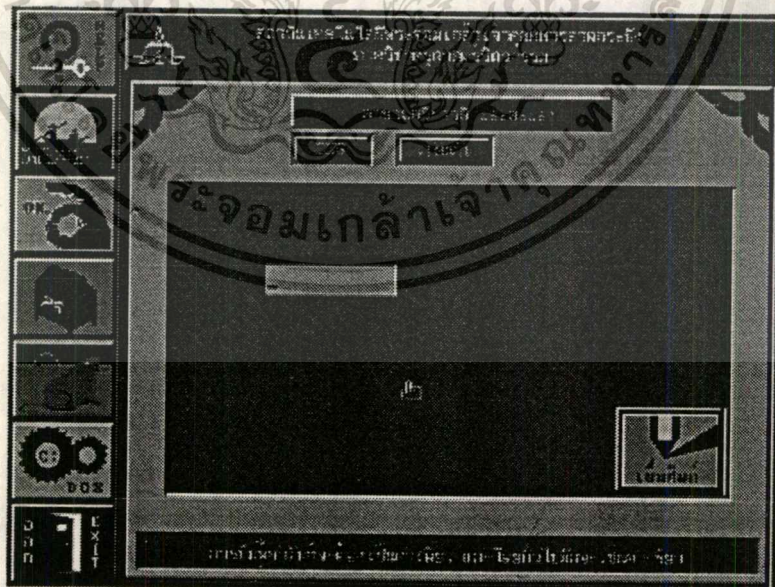
คลิกเมาส์ที่ "ใช่" กลับไปสู่โปรแกรมของเมนูศัพท์เริ่มต้นใหม่จอภาพปรากฏดังรูปที่ 4.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 เมนูศัพท์

4.3.2 เพิ่มศัพท์เป็นส่วนที่ทำหน้าที่เพิ่มคำศัพท์ลงไปยัง โปรแกรมพจนานุกรมกริยาวลีและ
สะแลง ตามที่ผู้ใช้ต้องการคลิกเมาส์ที่เพิ่มศัพท์ จอภาพปรากฏดังรูปที่ 4.22

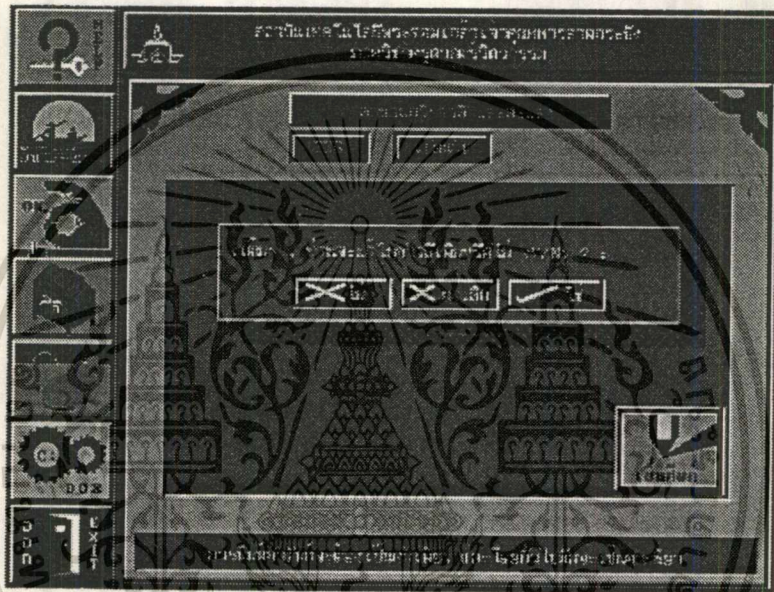


รูปที่ 4.22 เมนูของการเพิ่มศัพท์

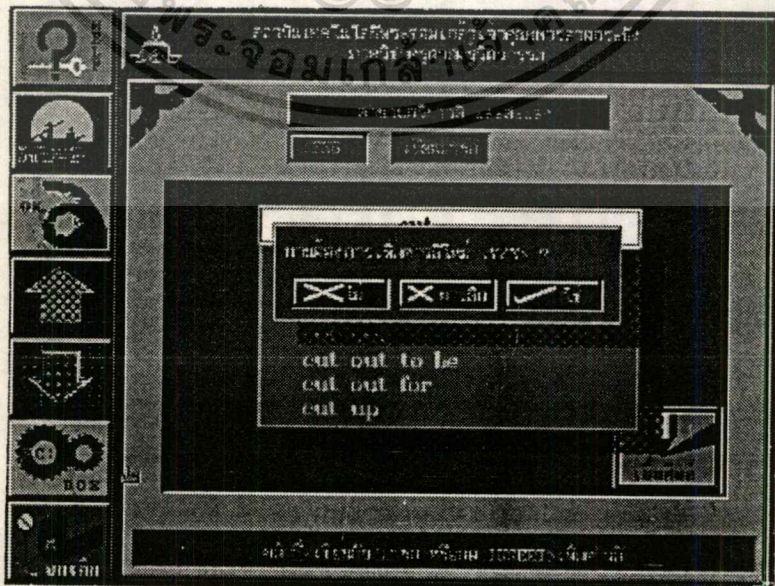
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เหมือนกับเมนูค้นหาศัพท์ คำศัพท์ที่ผู้ใช้จะใส่เพิ่มเติมมี 3 รูปแบบ คือ รค่า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. แบบที่เป็นคำศัพท์ที่เต็มแล้ว
2. แบบที่คำศัพท์มีอยู่แล้วแต่ยังไม่เต็ม
3. แบบที่เป็นคำศัพท์ใหม่

เมื่อเพิ่มคำศัพท์ลงไปยังโปรแกรม ปรากฏว่าเป็นแบบที่เป็นคำศัพท์ที่เต็มแล้ว จอภาพปรากฏดังรูปที่ 4.23

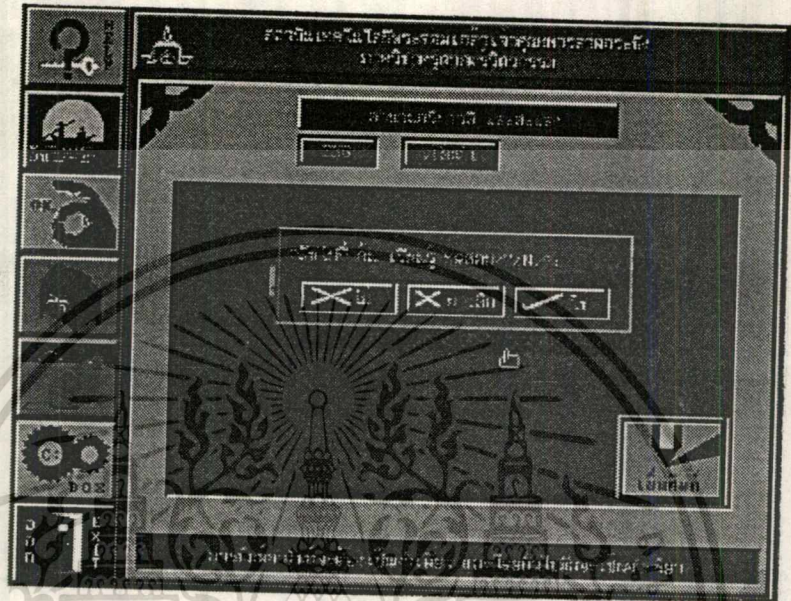


รูปที่ 4.23 กรณีคำศัพท์ที่ไม่สามารถเพิ่มศัพท์

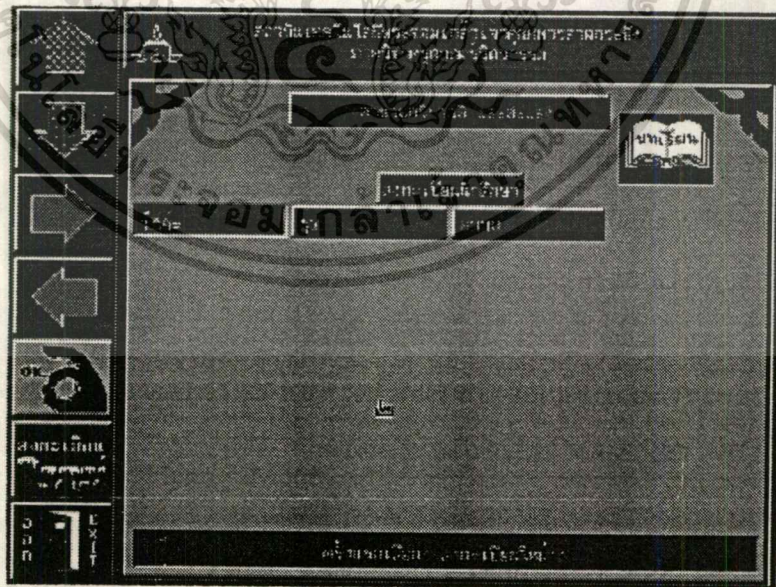


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 4.24 คำถามของการเพิ่มศัพท์ อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ละครั้ง สามารถที่จะบันทึกการเรียนว่าได้เรียนถึงคำใด เมื่อจะมาเรียนในวันหลังจะสามารถเรียนต่อจากที่เคยเรียนค้างไว้ได้



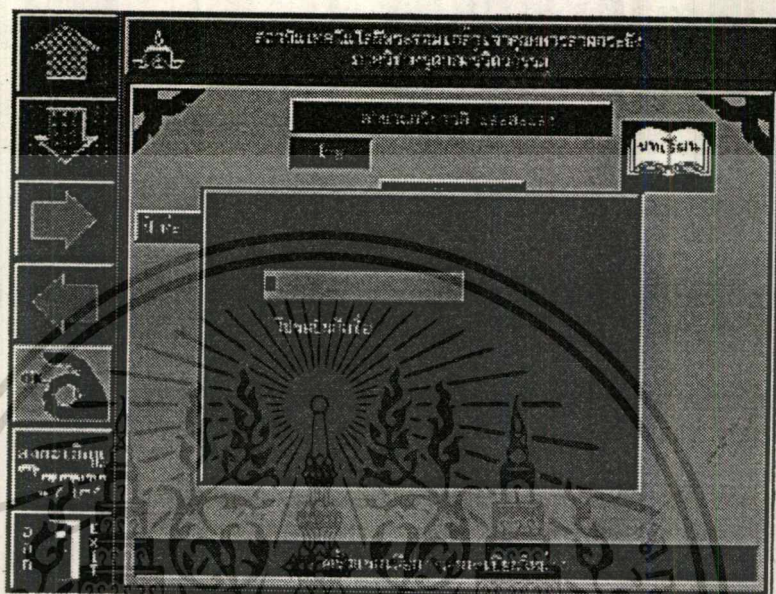
รูปที่ 4.26 ออกจากเมนูการเพิ่มศัพท์



รูปที่ 4.27 เมื่อเริ่มเข้าเมนูเรียนรู้

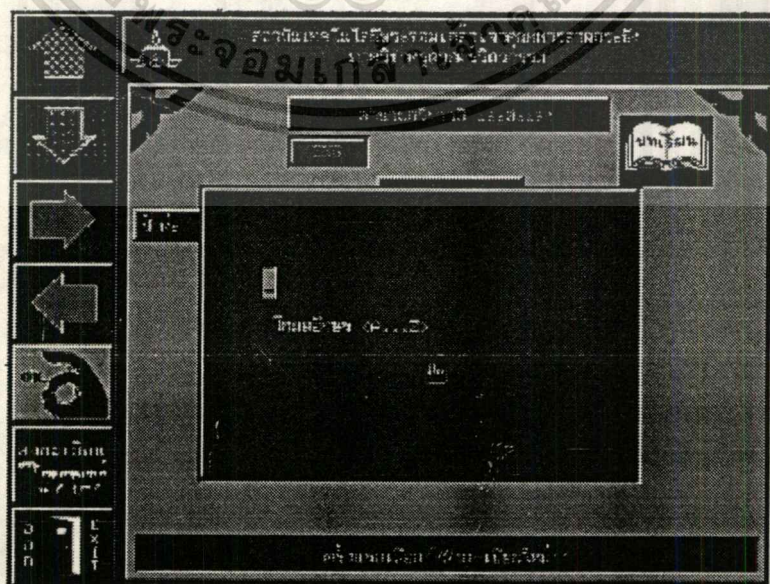
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเริ่มเรียนครั้งแรกผู้เรียนต้องทำการลงทะเบียน โดยเลือกเมนูลงทะเบียนโดยใช้เมาส์หรือคีย์บอร์ด Insert จะปรากฏที่จอภาพดังรูปที่ 4.30



รูปที่ 4.30 เมนูลงทะเบียน

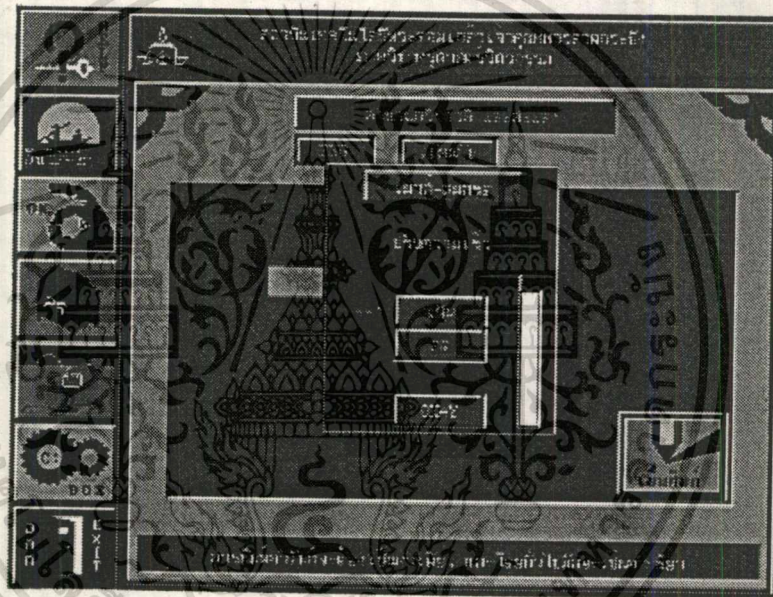
ผู้ใช้งานต้องพิมพ์ชื่อของตนเอง โดยกด Enter แล้วโปรแกรมจะสั่งให้เลือกตัวอักษรที่ต้องการจะเรียน ซึ่งที่จอภาพจะปรากฏรูปดังรูปที่ 4.31



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการรูปที่ 4.31 เมนูเลือกตัวอักษร ไปอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กด Enter โปรแกรมจะแสดงคำศัพท์คำแรกของนั้นออกมา เราสามารถเปิดดูคำต่อไปได้ โดยการใช้เมาส์คลิกที่รูปหนังสือ จะได้ภาพเคลื่อนไหวเป็นภาพหน้ากระดาษกำลังเปิดไปมา เราสามารถยกเลิกการใช้งานเมนูนี้ได้โดยการกด ESC จะมีหน้าต่างให้เราเลือกออกจากโปรแกรม หากตอบใช่เป็นการออกโปรแกรมจะบันทึกสถานะปัจจุบันที่ได้เรียนคำศัพท์เพื่อที่จะเรียนวันอื่นๆต่อไป

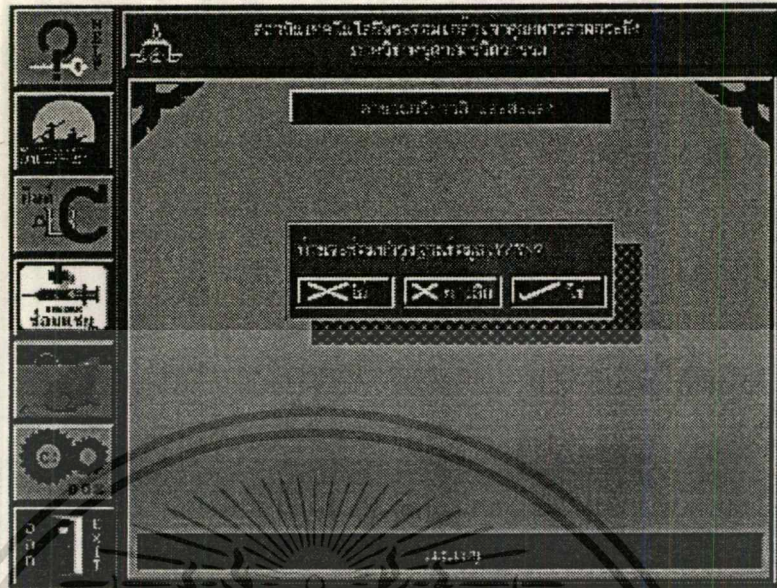
4.4 หนูปรับความเร็วของเมาส์เพิ่มลดได้ตามความต้องการของผู้ใช้คลิกเมาส์ที่หนูจอภาพปรากฏดังรูปที่ 4.32



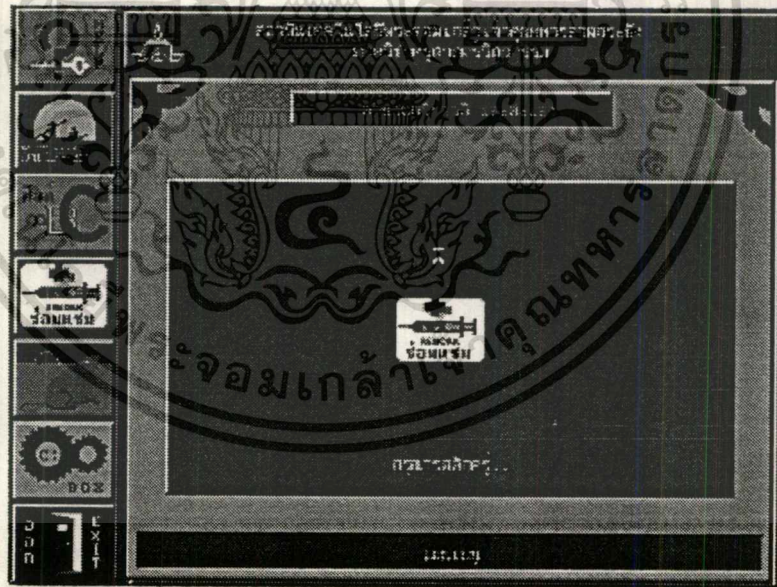
รูปที่ 4.32 เมนูของเมาส์

4.5 ซ่อมแซม ช่วยซ่อมแซมบางไฟล์ของโปรแกรมที่เสียหายให้ดีขึ้นโดยการคลิกเมาส์ที่ซ่อม จะทำการซ่อมแซมข้อมูลที่เสียหายของโปรแกรม จอภาพปรากฏดังรูปที่ 4.33

ดับเมาส์คลิกเค้าที่ ใช่ เป็นการซ่อมแซม จอภาพปรากฏดังรูปที่ 4.34



รูปที่ 4.33 ความพร้อมของการซ่อมแซม



รูปที่ 4.34 กำลังซ่อมแซม

4.6 DOS เป็นส่วนที่ทำการออกสู่ระบบ DOS และเข้ากลับโปรแกรมพจนานุกรม กริยาวลี และ
 สะแลงแบบเดิม พิมพ์คำสั่ง "EXIT" จะเป็นการเข้าไปสู่โปรแกรมที่จุดที่ผู้ใช้ออกมาตอนแรก

4.7 EXIT การออกจากโปรแกรมพจนานุกรมกริยาวลี และสะแลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและวิจารณ์

ในโปรแกรมพจนานุกรมกริยาวลีและสะแลง ได้รวบรวมคำศัพท์กริยาวลีและสะแลงต่างๆ มากกว่า 2,200 คำ สามารถค้นหาคำต่างๆ ได้รวดเร็วมีโปรแกรมการเรียนรู้คำศัพท์ การทดสอบความสามารถของผู้เรียนรู้ ผู้ใช้สามารถใช้งานได้สะดวกตลอดจนความถูกต้องของข้อมูลประสิทธิภาพในการทำงานของโปรแกรมพจนานุกรมกริยาวลีและสะแลงนี้สามารถทำงานได้อย่างง่ายๆ ทั้งยังมีส่วนของโปรแกรมช่วยเหลือต่างๆ ให้ผู้ใช้ได้ใช้งานอย่างสะดวก แต่งโครงการนี้ยังมีข้อบกพร่องและปัญหาเกิดขึ้นหลายประการทางกลุ่มผู้จัดทำ โครงการนี้ได้เขียนข้อเสนอแนะวิธีการแก้ไขปรับปรุงในแต่ละส่วน เพื่อที่จะเป็นได้ประโยชน์ให้กับผู้ที่นำโปรแกรมพจนานุกรมกริยาวลีและสะแลงนี้ไปพัฒนาในภายหน้า พร้อมทั้งสรุปผลของ โครงการนี้ ดังรายละเอียดต่อไปนี้

5.1 ปัญหาที่พบในการทำโครงการ

- 5.1.1. การทำงานของโปรแกรมช้า เมื่อมีการโหลดโปรแกรมจากแผ่นดิสก์ (เนื่องจากการโหลดรูปภาพด้วย)
- 5.1.2. ใช้เวลาเขียน SOFTWARE นานพอสมควร เพราะต้องใช้เวลาและประสบการณ์
- 5.1.3. ความหมายของคำศัพท์บางคำมีความหมาย
- 5.1.4. ใช้เวลาเก็บข้อมูลคำศัพท์เป็นเวลานาน

5.2 การแก้ปัญหา

- 5.2.1. การทำงานของโปรแกรมจะเร็วขึ้น ถ้าโหลดโปรแกรมจากฮาร์ดดิสก์
- 5.2.2. ปรึกษาเรื่องเทคนิค ในการเขียนโปรแกรมจากผู้รู้เพิ่มเติม
- 5.2.3. ดูจากตำราหลายเล่ม เพื่อให้ได้ความหมายที่แท้จริง

5.3 แนวทางในการพัฒนา

- 5.3.1. พัฒนาโปรแกรมในส่วนของกริยาวลีให้สามารถบรรจุคำศัพท์ได้มากยิ่งขึ้น
- 5.3.2. มีบททดสอบการเรียนรู้ให้มีความเข้าใจยิ่งขึ้น
- 5.3.3. เพิ่มในส่วนของกริยาวลีที่พ้องเสียงในตัวเอง

เอกสาร 5.3.4. พัฒนาระบบข้อมูลเชิงสัมพันธ์ให้ดียิ่งขึ้น ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 5.3.5. ลดหน่วยความจำที่ โปรแกรมใช้ให้น้อยลง
- 5.3.6. เพิ่มความสามารถในการนำหน่วยความจำขยายให้เป็นประโยชน์
- 5.3.7. พัฒนาขีดความสามารถของส่วนทดสอบความก้าวหน้าในการเรียน
- 5.3.8. พัฒนาความเร็วในการทำงานในระบบกราฟฟิก
- 5.3.9. จัดการในเรื่องเมาส์พอยท์เตอร์ให้ใช้งานได้ดีกับทุกระบบ
- 5.3.10. ปรับเปลี่ยนสื่ที่ใช้ใน โปรแกรมให้สอดคล้องกันยิ่งขึ้น

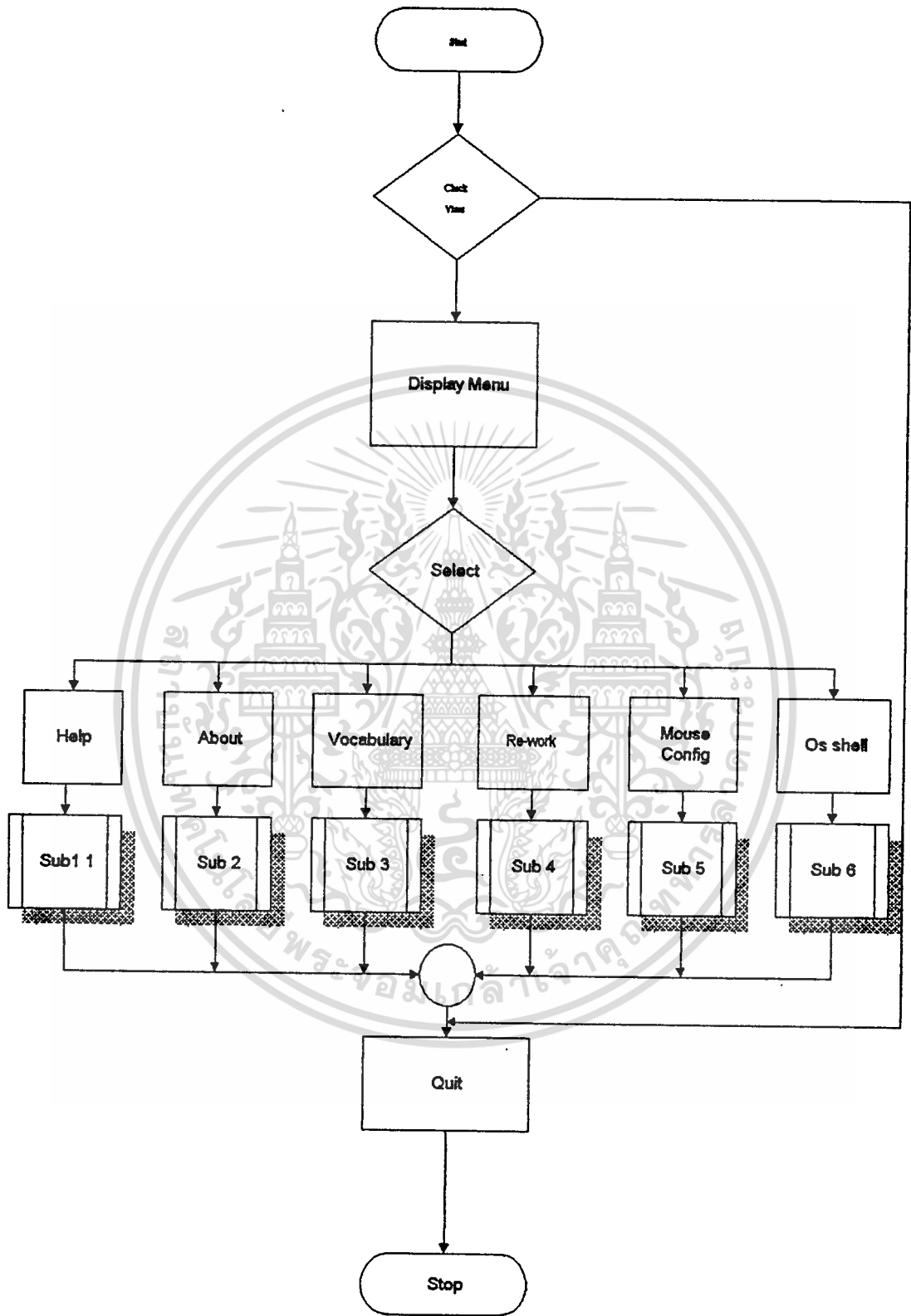
5.4. สรุปผลของโครงการ

โครงการโปรแกรมพจนานุกรมกริยาวลีและสะแลงที่ได้สร้างขึ้นนี้ผลของโครงการนี้ สามารถบรรลุวัตถุประสงค์ที่ได้ตั้งไว้กล่าวคือ

- 5.4.1 ได้รู้และเข้าใจการสร้าง โปรแกรมประยุกต์พจนานุกรมกริยาวลี
- 5.4.2 สามารถออกแบบระบบ โปรแกรมแบบ TOP DOWN DESIGN ได้
- 5.4.3 ได้รู้และเข้าใจการทำงาน โปรแกรมประมวลผลคำภาษาไทย และภาษาอังกฤษ
- 5.4.4 สามารถนำภาษาโปรแกรมซี มาประยุกต์ใช้งานได้
- 5.4.5 ได้รู้และเข้าใจระบบของฐานข้อมูลแบบ ไบนารีทรีและระบบจัดการฐานข้อมูลเชิงสัมพันธ์
- 5.4.6 สามารถนำระบบกราฟฟิกส์ มาใช้สื่อความหมายระหว่างเครื่องคอมพิวเตอร์กับผู้ใช้คอมพิวเตอร์ได้
- 5.4.7 สามารถสร้าง โปรแกรมช่วยสอนได้
- 5.4.8 สามารถประมวลผลความรู้ที่ผู้เรียนได้เรียนรู้จากโปรแกรมได้
- 5.4.9 สามารถนำไปใช้แก้ปัญหาการสร้างประโยคภาษาอังกฤษให้สมบูรณ์และเป็นที่นิยมใช้กัน โดยทั่วไปได้

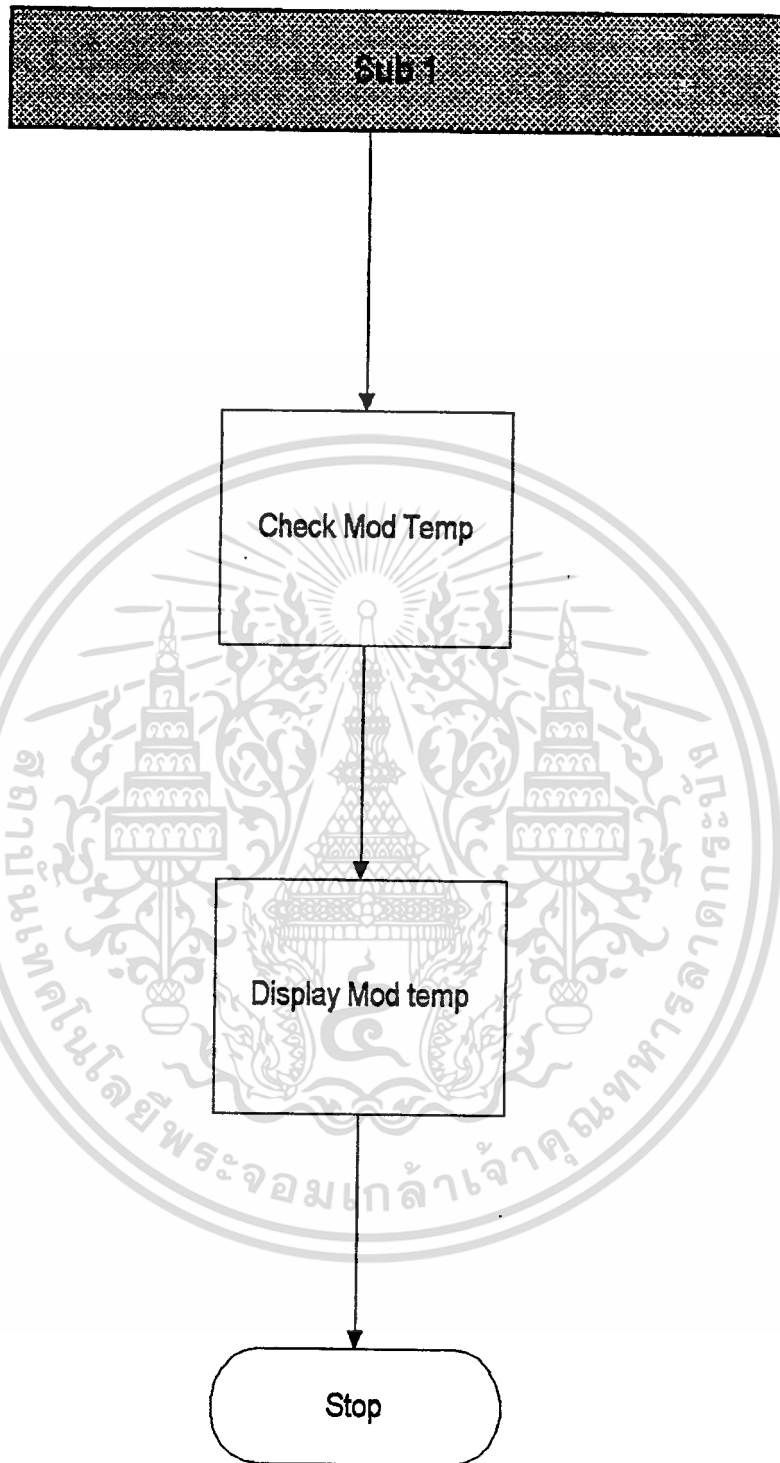


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



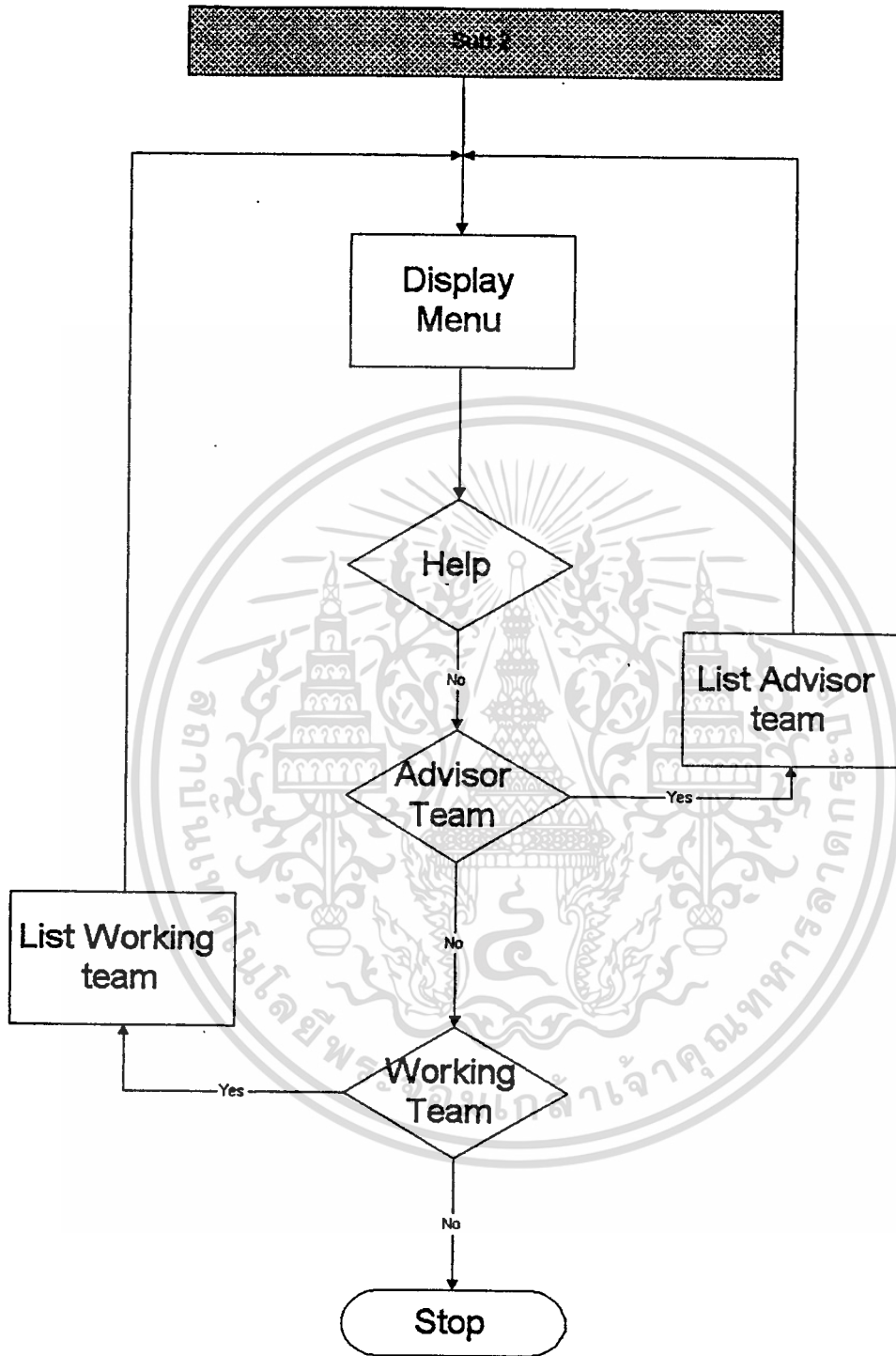
Main Menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



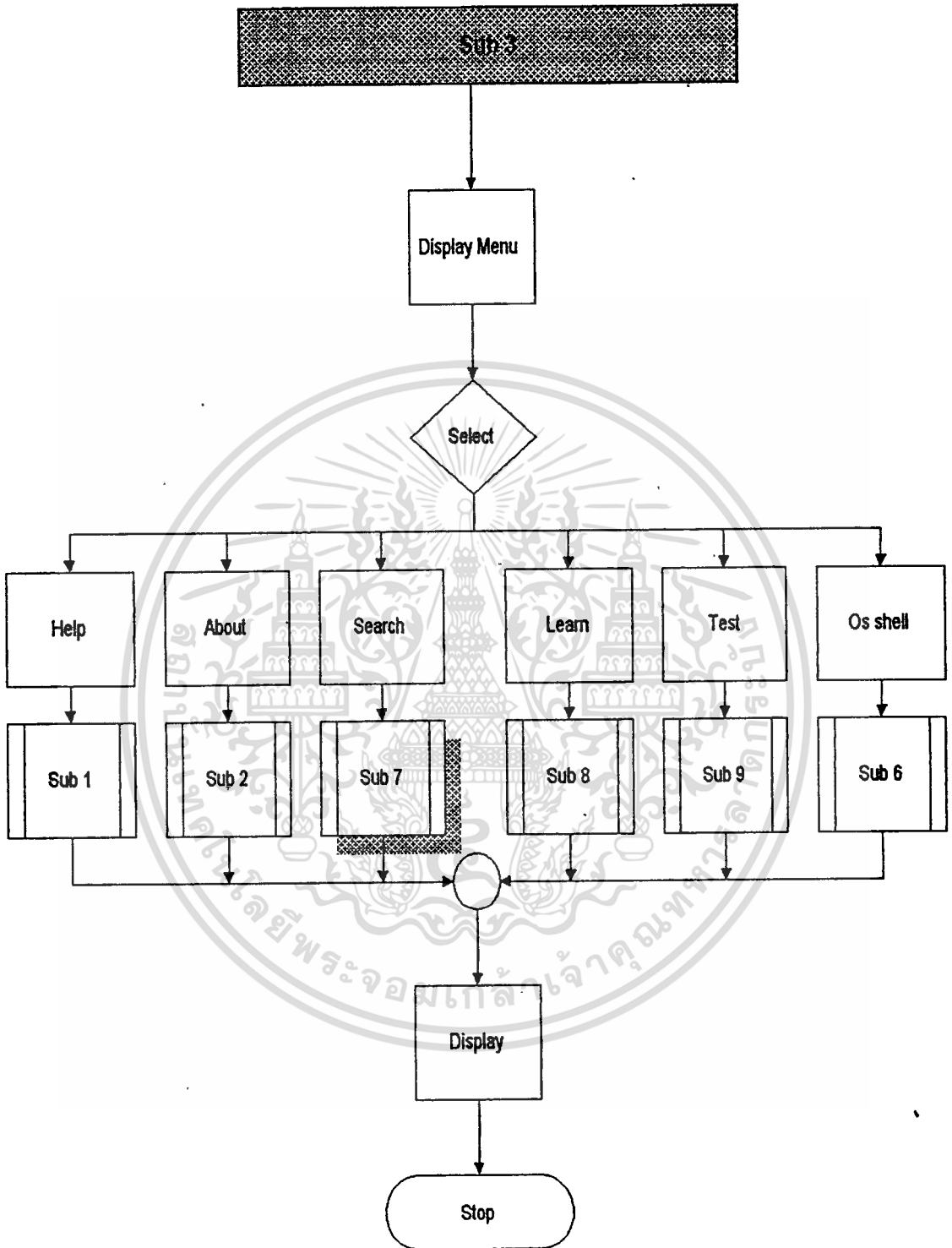
Help

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



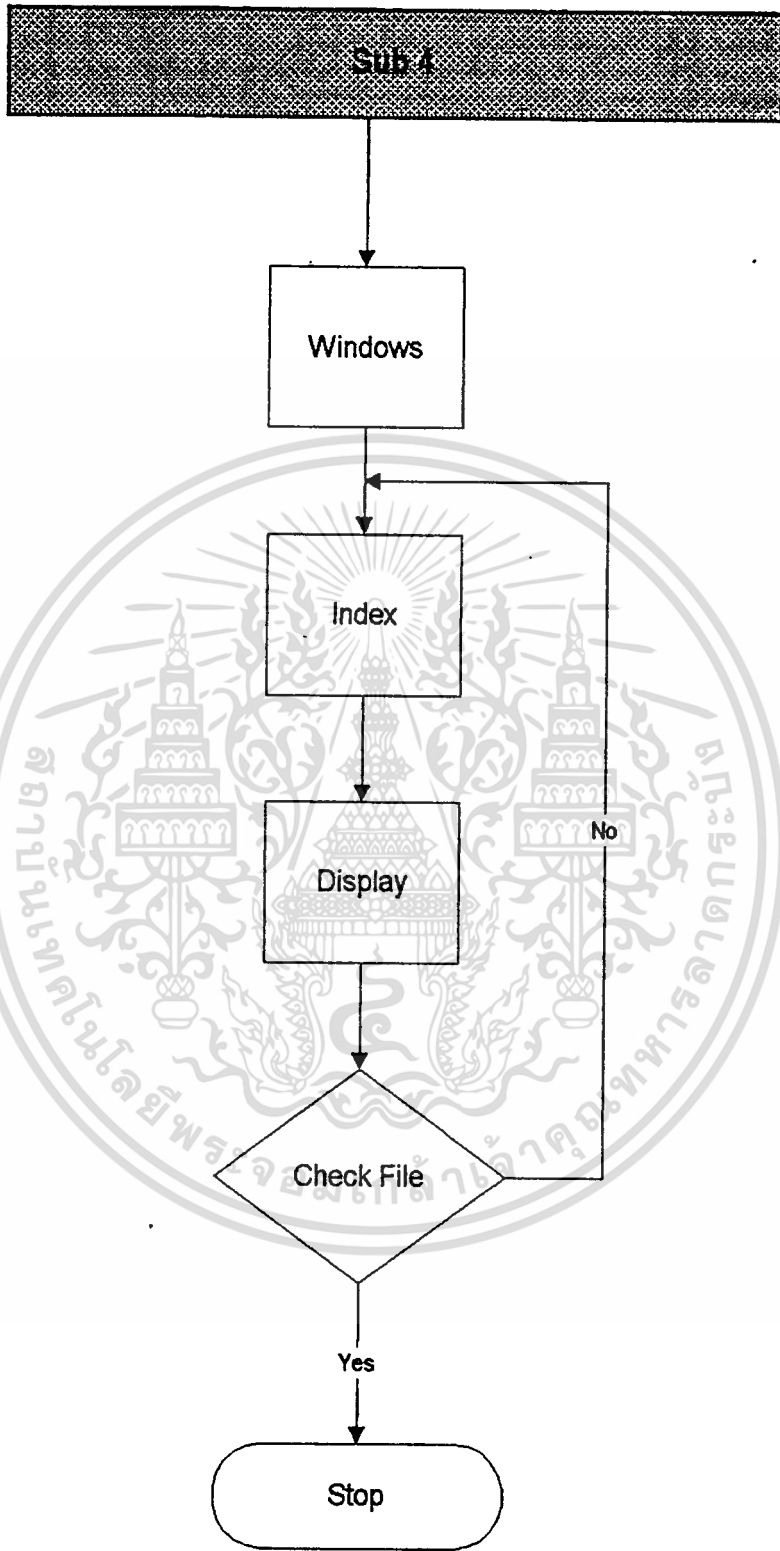
About

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



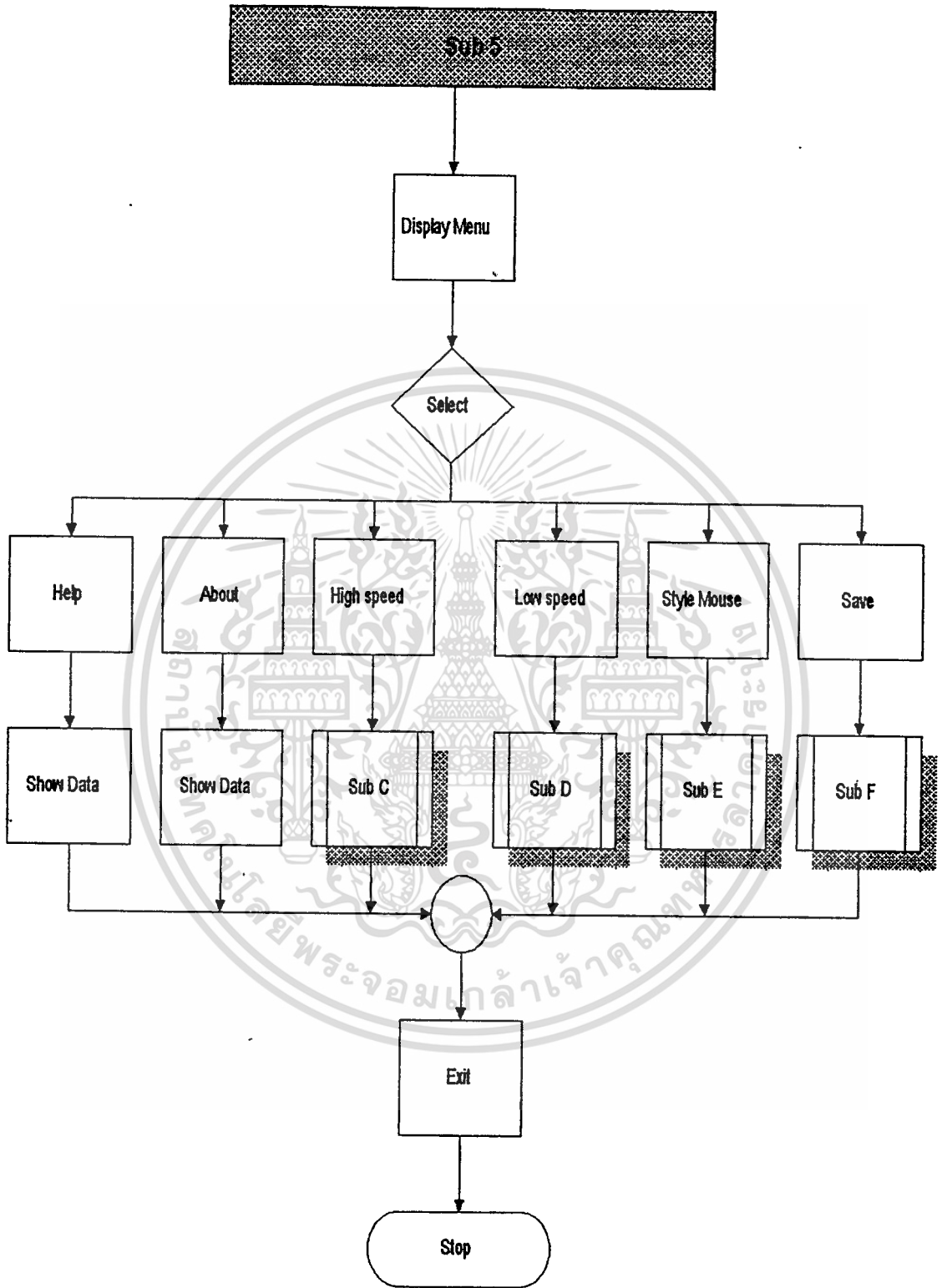
Main Phases Menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



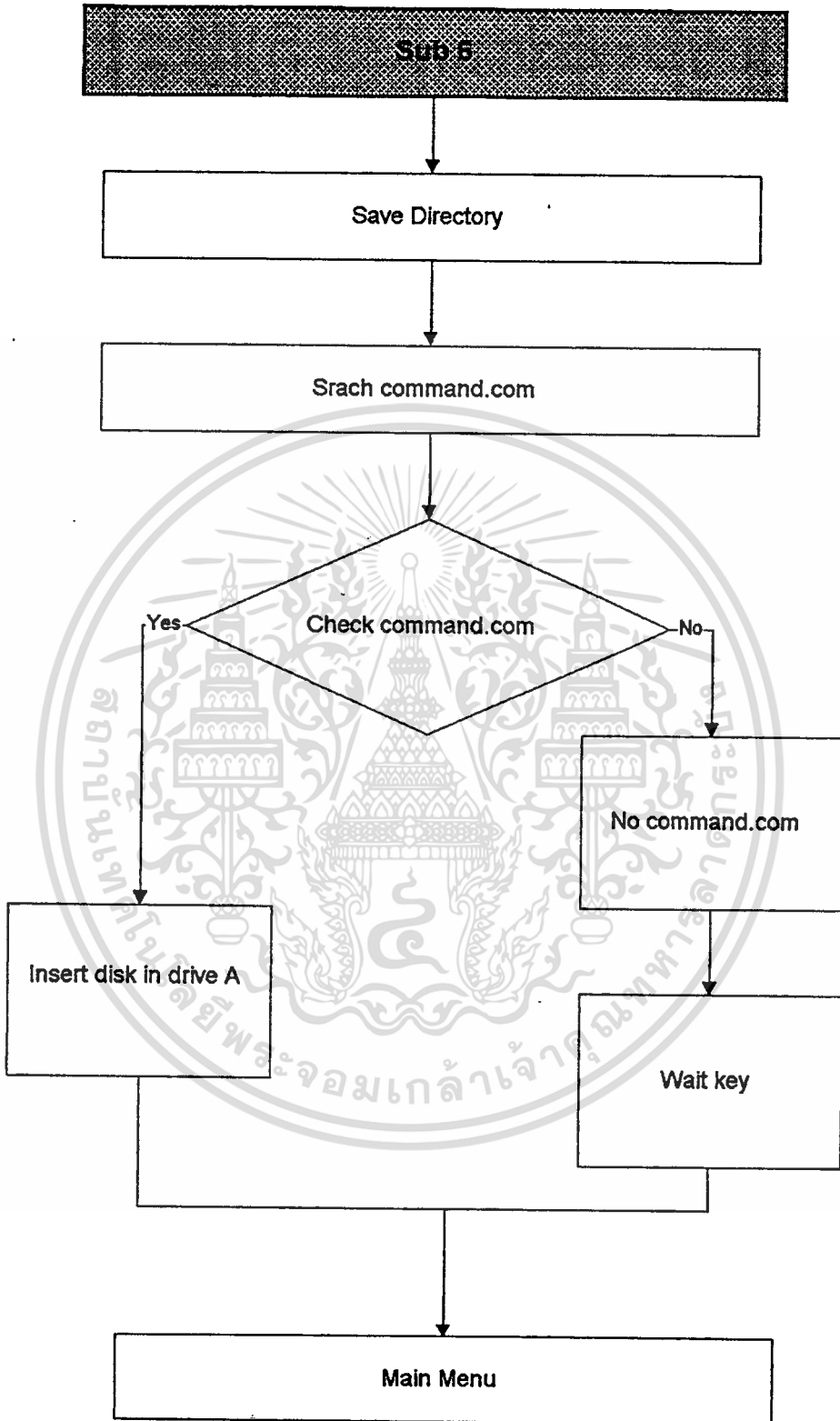
Re-Work

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



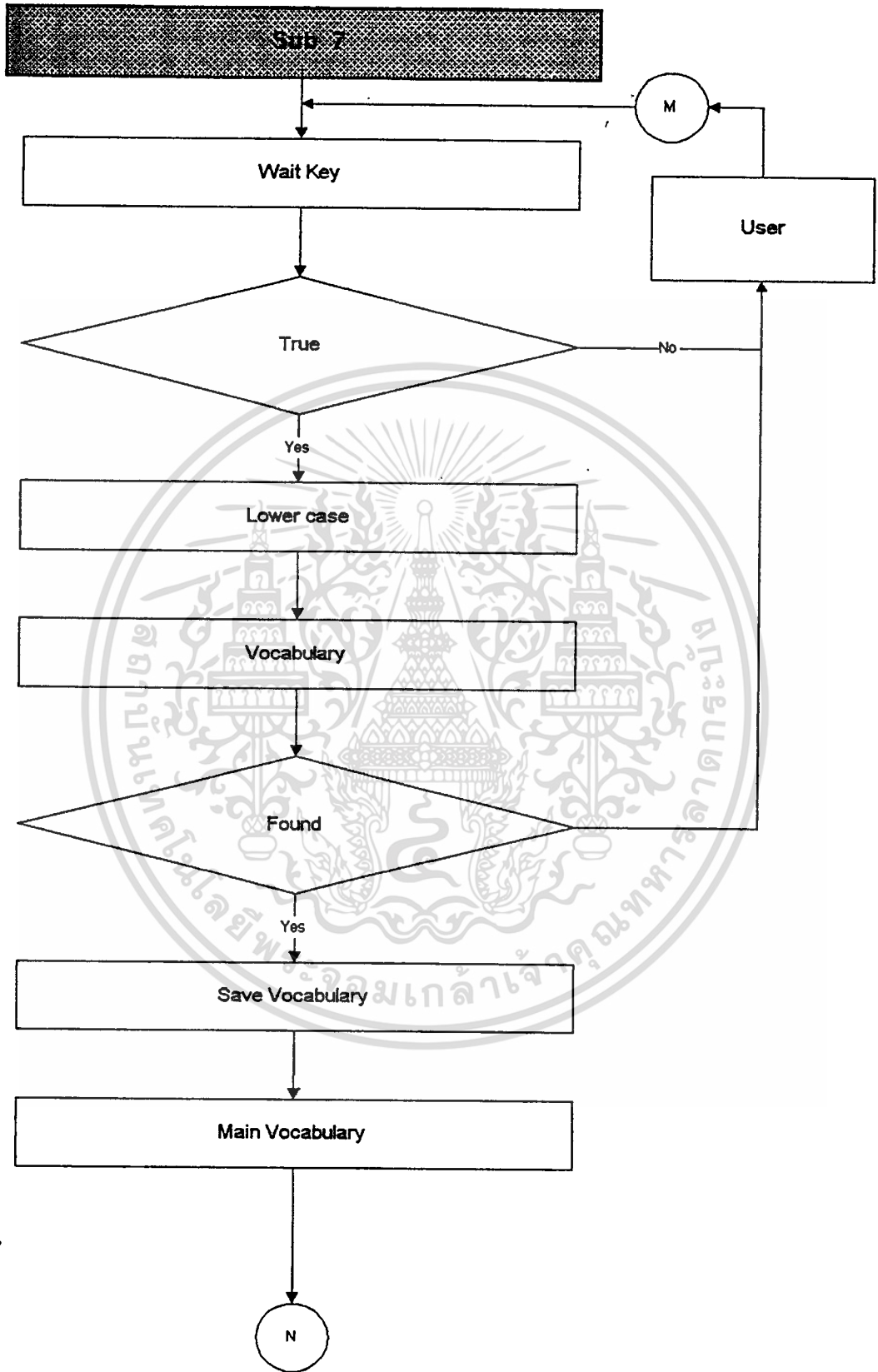
Mouse Config

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



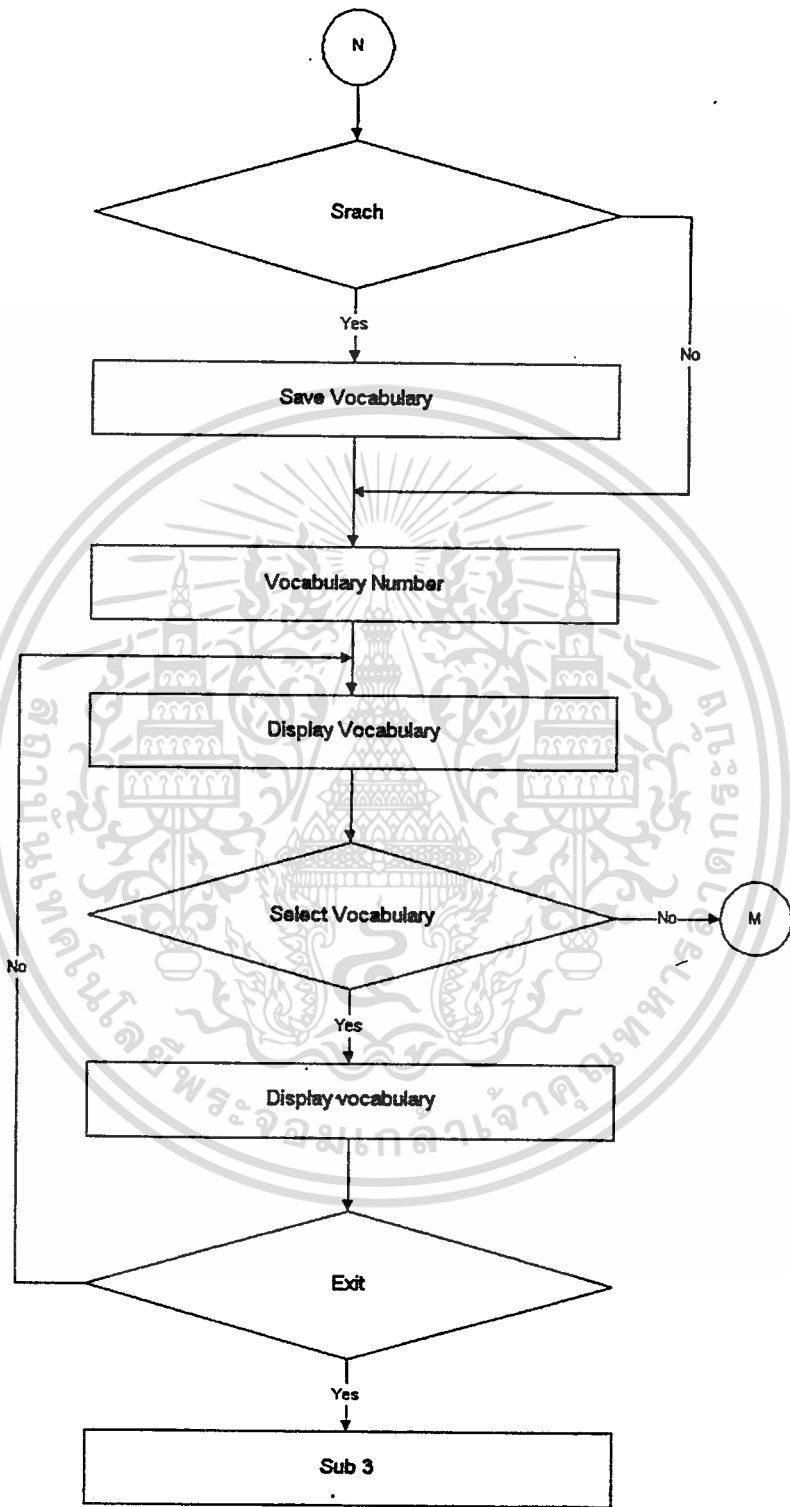
Os shell

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



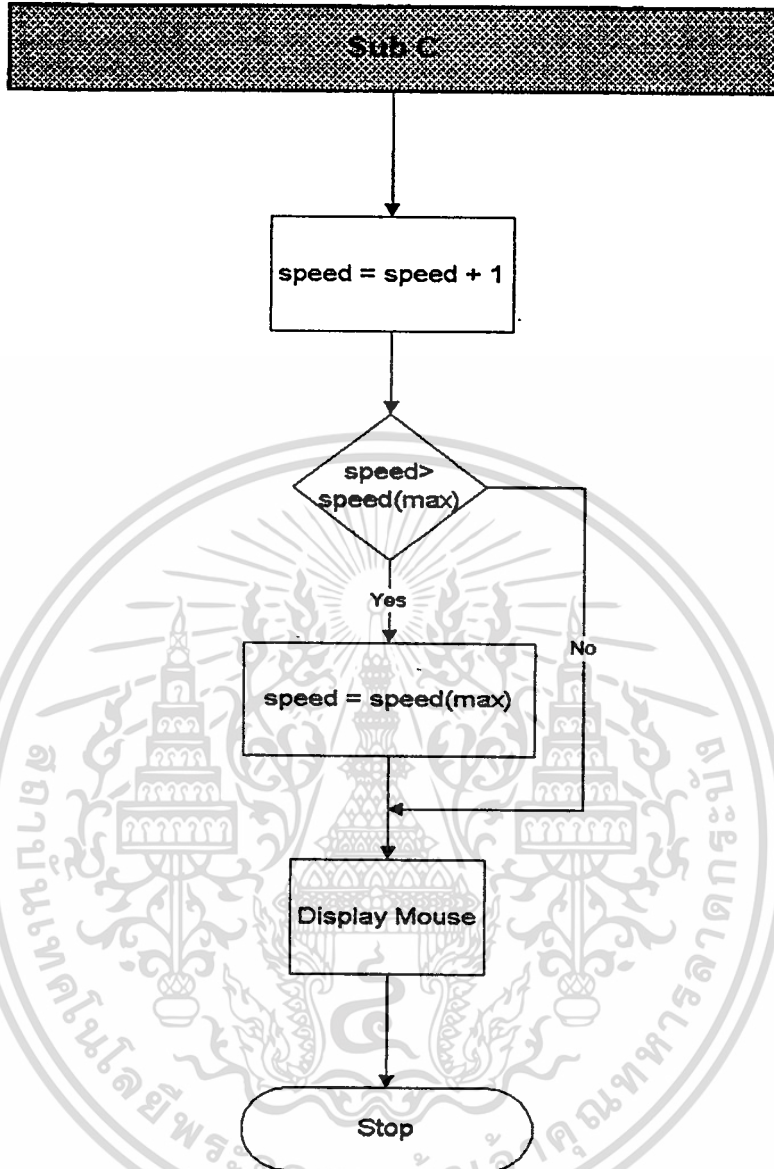
Strach Phases

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



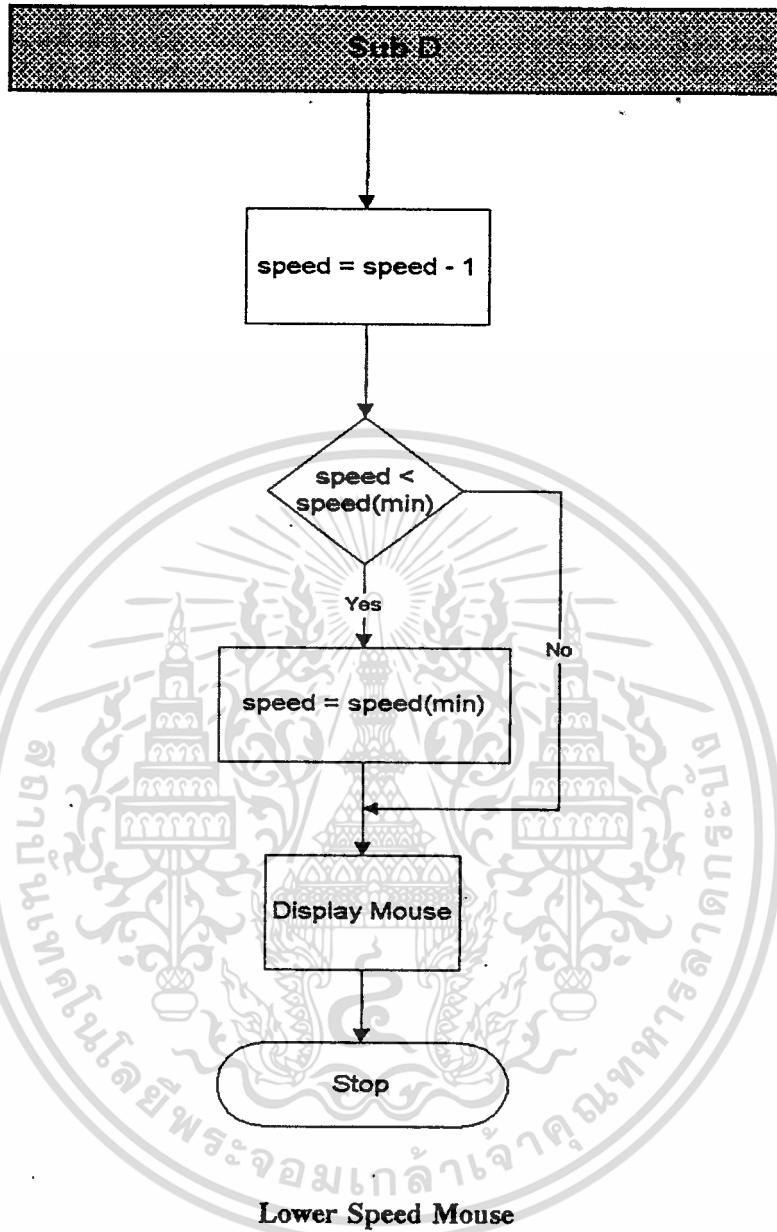
Srach Phases

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

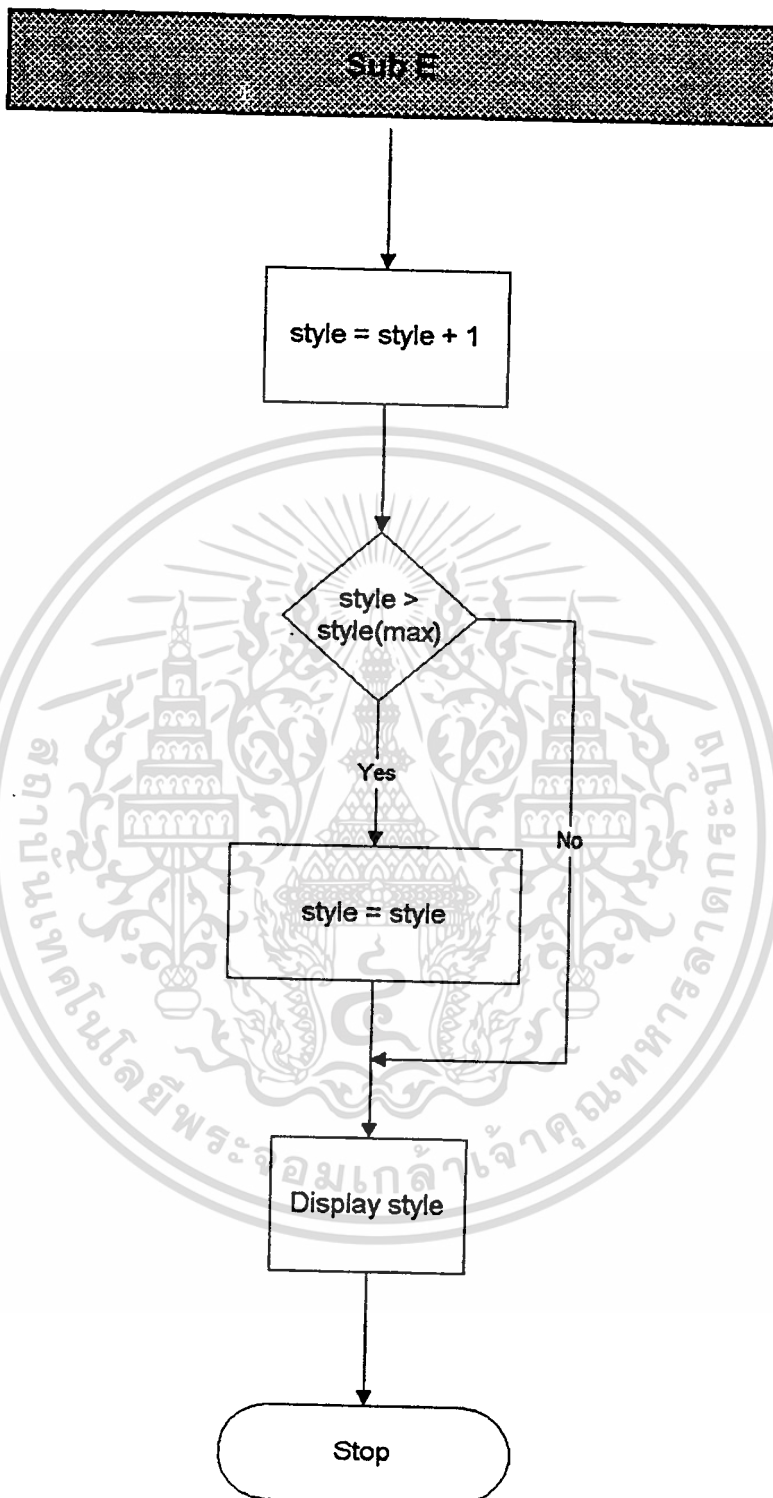


High Speed Mouse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

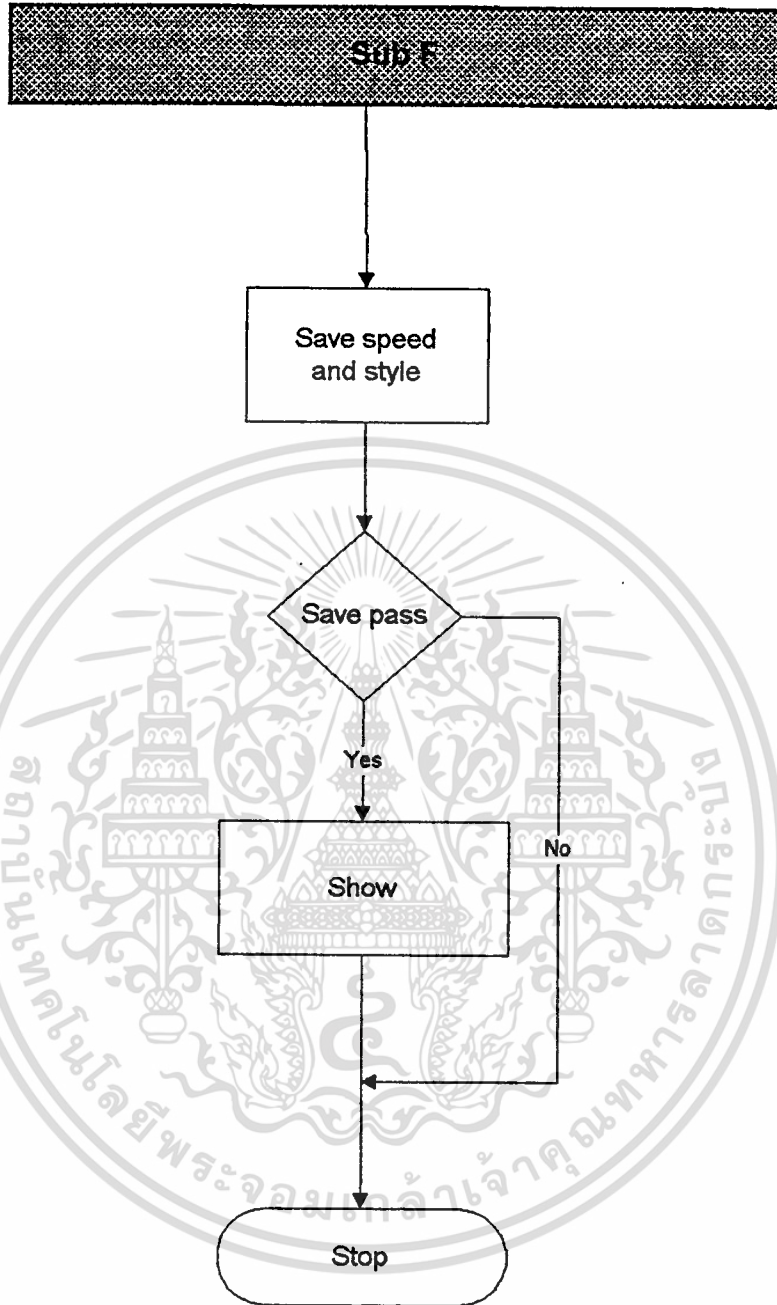


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Style Mouse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Save

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงปุ่มฟังก์ชันที่ควบคุมการใช้งานโปรแกรมพจนานุกรม กริยาวลีและสะแลง

ฟังก์ชัน F1 (ส่วนช่วยเหลือของโปรแกรม)

เป็นฟังก์ชันที่ทำหน้าที่แสดงรายละเอียดของหน้าที่ของปุ่มฟังก์ชันที่อยู่ในโปรแกรมพจนานุกรมกริยาวลีและสะแลง

ฟังก์ชัน F2

ทำหน้าที่บันทึกข้อมูลจากการใช้งานที่เมนูของการเพิ่มคำศัพท์

ฟังก์ชัน F4

ทำหน้าที่ตอบตกลง เป็นปุ่มที่ทำหน้าที่แทนการใช้เมาส์คลิกที่ OK

ฟังก์ชัน F10

เป็นปุ่มที่จะใช้งานในส่วนเมื่อเราเข้าเมนูเพิ่มศัพท์ ใช้เปลี่ยนการพิมพ์อักษรภาษาไทยเป็นการพิมพ์อักษรภาษาอังกฤษหรือเปลี่ยนจากการพิมพ์ภาษาอังกฤษเป็นการพิมพ์ภาษาไทย ESC

เพื่อยกเลิกการทำงานในแต่ละส่วนของโปรแกรมพจนานุกรมกริยาวลีและสะแลง โดยจะสั่งโปรแกรมให้ย้อนการทำงานกลับหลังมา 1 ขั้นตอน

Tab

ใช้เลื่อนแถบสว่างภายในเมนูไปยังตำแหน่งต่างๆ แทนการใช้เมาส์

Backspace

เพื่อลบตัวอักษรทางซ้ายของเคอร์เซอร์

Del

เพื่อลบตัวอักษรทางขวาของเคอร์เซอร์

Int

เพื่อเลือกการทำงานระหว่างการพิมพ์แทรกตัวอักษรระหว่างข้อความหรือพิมพ์ทับอักษร

ปุ่มลูกศรขวา

เพื่อเลื่อนเคอร์เซอร์ไปทางขวา 1 ตัวอักษร

ปุ่มลูกศรซ้าย

เพื่อเลื่อนเคอร์เซอร์ไปทางซ้าย 1 ตัวอักษร

ปุ่มลูกศรขึ้น

เพื่อเลื่อนเคอร์เซอร์ขึ้นทีละ 1 พิลด์

ปุ่มลูกศรลง

เพื่อเลื่อนเคอร์เซอร์ลงทีละ 1 พิลด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดกลุ่มไฟล์ตามหน้าที่การทำงาน

1. ไฟล์ที่ทำงานด้วยตัวเอง
 - DICMOUSE.COM เป็นเมาส์ไดรเวอร์แบบ 3 บุ่มที่ให้มาพร้อมกับโปรแกรม
 - PHASES.EXE เป็นโปรแกรมหลักจัดการระบบทั้งหมด
 - PH_VRS1.EXE เป็นโปรแกรมที่ใช้สร้างไฟล์ข้อมูลเก็บไว้ตรวจเช็คไฟล์เสียหาย
 - PH_UTIL.EXE เป็นโปรแกรมที่ใช้ซ่อมไฟล์ฐานข้อมูล
 - PH_SCHEM.EXE เป็นโปรแกรมที่ใช้สร้างพจนานุกรมข้อมูลให้กับฐานข้อมูล
 - *.BAT เป็นโปรแกรมที่ช่วยจัดการงานต่าง ๆ ให้ง่ายและเร็วขึ้น
2. ไฟล์ที่ทำงานโดยถูกเรียกผ่านทางโปรแกรมอื่น (child process)
 - PH_VRS2.PWA เป็นไฟล์ที่ใช้ตรวจไฟล์ที่เกิดการเสียหาย สร้างจากไฟล์สกุล .COM
3. ไฟล์ที่ใช้เป็นฐานข้อมูลหรือข้อมูลสำหรับอ้างอิง
 - เป็นไฟล์ที่มีนามสกุล .DAT
4. ไฟล์โปรแกรม
 - ไฟล์สกุล .C เป็นไฟล์ซอร์สโปรแกรมหลัก
 - ไฟล์สกุล .H เป็นไฟล์ที่ใช้ผนวกกับซอร์สไฟล์หลัก ซึ่งมีมากับคอมไพเลอร์
 - ไฟล์สกุล .I, .I1, .I2, .I3 เป็นไฟล์ที่ผนวกกับซอร์สไฟล์หลักสร้างขึ้นมาใช้เอง
 - ไฟล์สกุล .VOC เป็นไฟล์ต้นแบบสำหรับสร้างพจนานุกรมข้อมูลโดยใช้โปรแกรม PH_SCHEM.EXE
5. ไฟล์ตัวอักษร
 - PHASETAI.FON เป็นไฟล์ตัวอักษรภาษาไทย/อังกฤษแบบบิตแมพ
 - ไฟล์สกุล .CHR เป็นไฟล์ตัวอักษรภาษาอังกฤษแบบสโตรก
6. ไฟล์รูปภาพ
 - ไฟล์สกุล .DRW เป็นไฟล์ที่ใช้แสดงภาพกราฟฟิกส์ และ ไอคอน
7. ไฟล์กราฟฟิกส์ไดรเวอร์
 - ไฟล์สกุล EGAVGA.BGI เป็นไฟล์ที่ใช้เพื่อให้โปรแกรมทำงานในโหมดกราฟฟิกส์แบบ VGA (เนื่องจาก โปรแกรมหลักมีขนาดใหญ่ จึงไม่ได้คอมไพล์ไฟล์นี้รวมกับโปรแกรมหลัก)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
/*          ph_qd.c          */
/*          Function : Main Program          */
/*-----*/

```

```

#define CLIC 1
#define No 0
#define Yes 1
#define Cancel 2
#define PH_NO_LIMIT 5
#define MAX_PH_LIST 25
#define MAX_STU_NAM 40
#define SV_USER 11
#define DIC_PH 9
#define GET_STU 12
#define GET_MAIN 0
int noload_kb=Yes;

char *rb; // record buffer
char *hb; // hold buffer
char *sc; // screen buffer
int len; // record length
char *malloc( );
int fl[ ] = {0, -1};
int *els;
char *fileuser = "          ",
        *UserName="          ",
        *UsersAlpha=" ", *UsersItemUnit=" ",
        *UsersOrgUnit=" ", *UsersEndUnit=" ";

int FILE_CHG=0;
int ADD_PHASE=0;

```

```

int FILE_TEM=No;
int HELPABOUT=0;

struct viewporttype view1,view2,view3,view4,vp;
int ModeTemp,VMode, ErrorCode=0, xold, yold,
    exit_time = 0,CAPNUM=0, VocMode,IconHi=68,
    bar_height = 50, VStep, box_width,
    NumPress,NumChoose,FontStyle=5,FontSize=4;

void *kng_image;
void query( ), rcdin( ), rcdout( ), set_trap( ), clear_record( );
int key_entry( );
void initial( ), ExitProg( ), VocabStyle( ),
    VocBarAtt( ), OutTextXYShade( ), Sh_ChooseButton( ),
    Sh_ReleaseButton( ), ClearMemData( ),
    ButtonTab( ), ButtonTakeOver( ), KeyMouRelease( ),
    ButtonMidTakeOver( ),WinButton( ),Shadow( ),
    HelpAbout( ), ShPhaseDic( ), OpenDBase( ), Sh_NextPage( );
char *GetString( ), *IncDecNumStr( );

#include <ph_incl.i>
m_result *rodent;
int iplist [MXELE+1];

int LookForPhase( );
void BarPulldown( ), PhaseSupport( ),
    BarLine( ), CursorShape( );
int TakeOverDbFile( ), ToSys( );
void label_box( );
int GetKeyMou(int gx,int gy);
void CapsNum(void);
void bar_color(int l, int t, int r, int b, int colorgnd,
    int colorbox,int boxmode);

```

```
void gr_cursor(int col, int row, int mode);
```

```
void CheckSys(void), DoVocab( );
```

```
static int file;
```

```
int existing_record;
```

```
void main(void)
```

```
{
```

```
    initial( );
```

```
    CheckSys( );
```

```
    ExitProg( );
```

```
}
```

```
void ExitProg(void) {
```

```
    if(image_voc!=NULL)
```

```
        free(image_voc);
```

```
    m_hide( );
```

```
    ToView(1);
```

```
    WindowHide(0,0,maxx,maxy,BLACK,BLACK,0);
```

```
    cleardevice( );
```

```
    closegraph( );
```

```
    clrscr( );
```

```
    exit(0);
```

```
}
```

```
/* ----- process the query ----- */
```

```
static void query( )
```

```
{
```

```
    int (*validfunct)( );
```

```
    int term = 0, i,j;
```

```
    char *bfptr11="00000", *bfptr22="00000", *Inc_Last_Ph_No="00000";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char *bfptr1, *bfptr2, *dic_list;
*fl = file;

db_open("", fl);
clrrec(sc, els);
set_trap( );
do {
    term = data_entry( );
    switch (term){
        case DIC_PH:
            clear_record( );
            TakeOverDbFile( TellFileNo(MAIN_WORD,0) );
            bfptr1 = bf + epos(elist[0],elist);
            validfunct = sb[0].edits;
            ModeTemp=GET_WORD;
            break;
        case ADD_PH:
            rcdout( );
            if(last_rcd(file, 1, rb)==ERROR) {
                rcdout( );
                for(i=0; i<strlen(bfptr1); i++)
                    *(bf+i) = '0';
                add_rcd(file, bf);
            }
            rcdin( );
            strcpy(bfptr11,bf);
            TakeOverDbFile( TellFileNo(MAIN_WORD,1) );
            bfptr22 = bf + epos(elist[0],elist);
            validfunct = sb[0].edits;
            strcpy(bfptr22,bfptr11);
            strcpy(bfptr22,IncDecNumStr(bfptr11,1));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    bf=bfptr22;
    strcpy(Inc_Last_Ph_No,bfptr22);
    wr_screen=0;
    (*validfunct)(bfptr11);
    wr_screen=1;
    ModeTemp = ADD_PH;
    break;
case -6:
    NumChoose=5;
    ButtonTakeOver( );
    View2CPY( );
    ToSys(Yes);
    if(ModeTemp==GET_WORD)
        clear_record( );
    break;
case -7: if(ModeTemp!=EDIT_PH && ModeTemp!=ADD_PH)
        goto Exit;
    else
        term=F2; //goto case F2
case F2:
    if(ModeTemp==GET_WORD){
        clear_record( );
        break;
    }
    NumChoose=6;
    ButtonTakeOver( );
    View2CPY( );
    m_hide( );
    WindowHide(5,maxy/5,maxx-5,maxy-maxy/6,bkcolor,RED,1);
    WindowExpand(maxx/18,maxy/5,maxx-maxx/18,maxy-maxy/6,
    EGA_CYAN,0,3);
    Make_Screen(11,0);

```



```

if(ModeTemp==ADD_PH) {
    rcnout( );
    TakeOverDbFile( TellFileNo(MAIN_WORD,0) );
    bfptr1 = bf + epos(elist[0],elist);
    validfunct = sb[0].edits;
    strcpy(bfptr1,MAIN_WORD);
    wr_screen=0;
    (*validfunct)(bfptr1);
    wr_screen=1;
    bfptr1 = ToFieldNo(bf,2);
    while(*bfptr1)
        bfptr1++;
    bfptr1--;
    while(*bfptr1 && *bfptr1!=' ')
        bfptr1--;
    bfptr1+=2;
    *bfptr1 = ' ';
    //avoid '0' at the front of number
    while(*Inc_Last_Ph_No++=='0');
    Inc_Last_Ph_No--;
    for(i=0; i<strlen(Inc_Last_Ph_No); i++)
        *(bfptr1+i) = *(Inc_Last_Ph_No+i);
    *(bfptr1+i) = ' '; // delete '\0' in phaso_list
    bfptr1 = ToFieldNo(bf,2);
    ModeTemp=EDIT_PH;
}

if(ModeTemp==EDIT_PH) {
    rcnout( );
    FILE_CHG = Yes;
    if(FILE_TEM==Yes)
        ADD_PHASE ^= Yes;
    nlock_kb=ADD_PHASE==Yes?No:Yes;
}

```

```

        ModeTemp=GET_WORD;
        term=DEPART;
    }
    else
        rcldout( );
    break;
case DEPART:
    FILE_TEM=Yes;
    rcldout( );
    FILE_CHG = Yes;
    if(FILE_TEM==Yes)
        ADD_PHASE ^= Yes;
    nolock_kb=ADD_PHASE==Yes?No:Yes;
    if(ModeTemp==EDIT_PH || ModeTemp==ADD_PH ||
    ModeTemp==NO_ADD)
        ModeTemp=GET_WORD;
    if(ModeTemp==DIC_PH) {
        FILE_TEM=Yes;
        break;
    }
    display_template( );
    break;
case ESC:
    Exit:
    NumChoose=6;
    ButtonTakeOver( );
    View2CPY( );
    term=ESC;
    if(ModeTemp==ADD_PH || ModeTemp==EDIT_PH) {
        post_notice("F2 => บันทึกแล้วออก..");
        term=!ESC;
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น. ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else {
    if((i=ChkYN("กลับไปที กั้น เร็วรู้ ทดสอบ<Y/N>?:",
    Yes)) != Yes) {
        clear_record( );
        term =!ESC;
    }
    else
        FILE_CHG=!1; //for exit from dovocab( )
    }
    break;
    default: break;
}
term=term==DEPART ? ESC:term;
} while (term != ESC);
db_cls( );
}

/* ----- clear out the record area ----- */
static void clear_record( )
{
    int i = 0;
    while (index_ele [file] [0] [i])
        protect(index_ele[file][0][i++],FALSE);
    clrscr(sc, cls);
    existing_record = FALSE;
}

/* ----- get the data base file record ----- */
static void rdin( )
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int i = 0;

    if(empty(rb, rlen(file)) == 0){
        rcd_fill(rb, sc, file_ele [file], els);
        mov_mem(rb, hb, rlen(file));
        existing_record = TRUE;
        while (index_ele [file] [0] [i])
            protect(index_ele[file][0][i++],TRUE);
    }
}

/* ----- add or update the data base file record ----- */
static void rcdout( )
{
    if (empty(sc, len) == 0){
        rcd_fill(sc, rb, els, file_ele[file]);
        if (existing_record){
            if (same( ) == 0){
                post_notice("บันทึก");
                rtn_rcd(file, rb);
            }
        }
        else{
            post_notice("บันทึก");
            if (add_rcd(file, rb) == ERROR)
                dberror( );
        }
        clear_record( );
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* ----- test for an empty record buffer ----- */
static int empty(b, l)
char *b;
int l;
{
    while (l--)
        if (*b && *b != ' ')
            return FALSE;
        else
            b++;
    return TRUE;
}

/* ----- test two record buffers for equality ----- */
static int same( )
{
int ln = rlen(file);

while (--ln)
    if (*(rb + ln) != *(hb + ln))
        break;
return (*(rb + ln) == *(hb + ln));
}

/* ----- set the query screen's key element trap ----- */
static void set_trap( )
{
int i = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while (index_elo [file] [0] [i])
    i++;
edit(index_elo [file] [0] [i-1], key_entry);
}

/* --- come here when the primary key has been entered ---- */
static int key_entry(s)
char *s;
{
char key [MXKEYLEN];
int i;

if (spaces(s))
    return OK;
*key = '\0';
i = 0;
while (index_elo [file] [0] [i]) {
    protect(index_elo[file][0][i],TRUE);
    strcat(key, sc + epos(index_elo[file][0][i++], els));
}
if (find_red(file, 1, key, rb) == ERROR){
    existing_record = FALSE;
    return OK;
}
redin( );
tally( );
return OK;
}

void initial( )
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

clrscr( );
opengraph( );
}

void OpenDBase(char *msg, int mode, int YN) { //1=Y 0=N of db_open
//msg=string used to check first alpha
//mode0=dicword family
//mode1=vocphase family
//mode2=student family
//YN=control db_open( )
char *fileopen[8] = {
    "dicwords",
    "dicword2",
    "dicword3",
    "vocphase",
    "vocphas2",
    "vocphas3",
    "students"
};

if ((file = filename( fileopen[TellFileNo(msg,mode)] )) == ERROR){
    printf("ERROR FILENAME");
    exit(1);
}

len = rlen(file);
cls = file_elo [file];
sc = malloc(len);
rb = malloc(rlen(file));
hb,= malloc(rlen(file));
if (sc == (char *) 0 ||

```

```

rb == (char *) 0 ||
hb == (char *) 0){
    printf("\nOut of memory");
    exit(1);
}
init_rcd(file, rb);
init_rcd(file, hb);
init_screen(fileopen[TellFileNo(msg,mode)] , els, sc);
if(YN==1) {
    *fl=file;
    db_open("", fl);
    clrscr(sc, els);
    set_trap( );
}
}

char *IncDecNumStr(char *num_str, int mode) {
//mode0==Decrement mode1==Increment
int i;
char *num=" ";
if(mode==1)
    itoa((i=(atoi(num_str)+1)),num,10);
else
    itoa(i=((i=(atoi(num_str)-1))<0)?0:i,num,10);
if(i<10)
    i=(i==0)?5:4;
else if(i<1000)
    i=(i<100)?3:2;
else
    i=1;
mov_mem(num, num+i,strlen(num));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

while(i-->0)
    *(num+i)='0';
*(num+5)='\0';
return num;
}

```

```

int PhasePullDown(char *bfptr, int maxphase, char *ExitCode) {
//ExitCode=="0" or "-1"(ESC) , give return ModeTemp
//ExitCode=="1" count item phasal-verb
int (*validfunct)( , i, j, k, m, n, p=0, c, //maxphase2=maxphase,
    max_lim=7, maxx2, maxy2, NumPullDown=0, repulldown=0, takeover,
    oldmodetemp=ModeTemp;
int maxphase2=maxphase;
char *bfptr2;
char *PhasNam;

if((PhasNam=(char *)malloc(850))==NULL) {
    //800=MAX+PH_LIST * 32+50=25*32+50
    error_message("Not enough memory");
    exit(1);
}
strcpy(ExitCode,"0"); //if(exitcode==1 ,exit end of function
bfptr = bf + epos(elist[0],elist);
validfunct = sb[0].edits;
if(maxphase !=NEW_PH) {
    //clear memory
    for(i=0; i<850; i++)
        *(PhasNam+i)=' ';
do { //RePullDown
    say_command(VocMode==WR_PH?"คลิกที่<เขียนทับ/แทรก>"
        "หรือกด<INSERT>เพิ่มคำวลี"

```

```

:"เลือกทำมีคศัพท์");
maxx2=view2.right-view2.left;
maxy2=view2.bottom-view2.top;
setviewport(view2.left+mx5+3, view2.top+my5+20,
view2.right-mx5+(maxx/25),
view2.top+5+my4+maxy/7+(maxy/22)+(20*(maxphase>max_lim?
max_lim:maxphase)),1);
getviewsettings(&view3);
maxx=view3.right-view3.left;
maxy=view3.bottom-view3.top;
if(m_events->yaxis>view3.top-30 &&
m_events->yaxis<view3.bottom-30 &&
m_events->xaxis>view3.left-30 && m_events->xaxis<view3.right+30)
    m_moveto(view3.left-50,view3.top);
    WindowExpand(0, 31, maxx-(maxx2/25), maxy-(maxy2/22),
    DARKGRAY, YELLOW, 1);
    WindowExpand(0,0,maxx-(maxx2/25),30,15,EGA_CYAN,1);
    Shadow(maxx2/25,maxy2/22,maxx,maxy,maxy-(maxy2/22)-1,
    maxx-(maxx2/25)+1);
    ///setview in window3
    setviewport(view3.left, view3.top,
    view3.right-(maxx2/25)-4, //r-4
    view3.bottom,1);
    maxx=view3.right-(maxx2/25)-4 - view3.left;
    settxtstyle(TRIPLEX_FONT,HORIZ_DIR,1);
    settxtjustify(CENTER_TEXT, CENTER_TEXT);
    setcolor(BLUE);
    outtextxy(maxx/2+(maxx2/25),11, MAIN_WORD);
    settxtjustify(LEFT_TEXT, CENTER_TEXT);
    if(repulldown==0) {
        i=j=0;
        setcolor(YELLOW);

```

```

do {
    bfptr2 = &phase_no[maxphase--][0];
    strcpy(bfptr,bfptr2);
    wr_screen=0;
    (*validfunct)(bfptr);
    wr_screen=1;
    strcpy(PhasNam+(i*32),ToFieldNo(bfptr,2));
    *(PhasNam+(i*32)+31)='\0';
    if(i<=max_lim)
        outtxxy(maxx/8, 43+(20*j),PhasNam+(i*32));
    else {
        ToView(3);
        showchar(15,maxy-22,(unsigned char)252);
        //DN arrow
    }
    i++;
    if(j==NumPullDown) {
        setwritemode(XOR_PUT);
        setcolor(RED);
        BarLine(37, 36+(j*20), maxx, 56+(j*20));
        setwritemode(COPY_PUT);
        setcolor(YELLOW);
    }
    j++;
    k=0;
} while(maxphase>=0);
} //IF
else {
    ToView(3);
    setwritemode(COPY_PUT);
    bar_del(3,34,maxx,maxy,DARKGRAY);
    setcolor(YELLOW);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

settextjustify(LEFT_TEXT, CENTER_TEXT);
for(n=0,m=p*(max_lim+1); m<=(p+1)*(max_lim+1); m++){
    outtextxy(maxx/8, 43+(20*n++),PhasNam+(m*32));
    if(m==NumPullDown) {
        ToView(2);
        BarPullDown(k);
        setwriteMode(COPY_PUT);
        setColor(YELLOW);
        ToView(3);
    }
    ToView(3);
    if(m==maxphase2) {
        setColor(DARKGRAY);
        showchar(15,maxy-22,(unsigned char)252); //DN arrow
        break;
    }
    showchar(15,maxy-22,(unsigned char)252); //DN arrow
}
}
m_hide( );
Make_Screen(NumChoose==274:9,0); //check return from DIC-Mode
ToView(2);
j=NumPullDown;
ModeTemp=GET_PH;
Set_Graphic_Cursor(GLOVE);
ModeTemp2=GET_PH; //check for m_hide in sh_cursor( )
NumChoose=NumPress=-1; //for check first time
for(;;) {
    m_show( );
    i=GetKeyMou(-20,-20);
    BarPullDown(k);
    switch(i) {

```

```

case '\r':
    ButtonTab(6);
    while(kbhit( ))
        getch( ); //clear kb_buffer
    break;
case -3: NumChoose=2;
case '\r':
    i='\r';
    switch(NumChoose) {
        case 0:
        case 1:
            i=NumChoose==0?-1:-2;
            goto H_A1;
        case 2:
            NumChoose=2;
            ButtonTakeOver( );
            i='\r';
            break;
        case 3:
        case 4:
            i=NumChoose==3?-14:-15;
            goto CURSOR1;
        case 5:
            goto SYS1;
        case 6:
            i=ESC;
            break;
        default:break;
    }
    break;
case UP:
case DN:

```

case -14:

case -15:

```

CURSOR1:
if(i==DN || i==-15) {
    i=DN;
    if(++j>maxphase2) {
        j=maxphase2;
        break;
    }
    if(j>max_lim && j%(max_lim+1)==0) {
        k=0;
        p++;
    }
    else
        k++;
    NumChoose=4;
}
else{
    i=UP;
    if(--j<0)
        k=j=0;
    else if(--k<0) { //j>=0
        k=max_lim;
        p--;
    }
    NumChoose=3;
}
ButtonTakeOver( );
break;

```

case F1:

case -1:

case -2:

```

H_A1:
NumChoose=i--1?0:1;
ButtonTakeOver( );
ToView(2);
BarPulldown(k);
ModeTemp2=!GET_PH;
ModeTemp=GET_PH;
HelpAbout(i);
ModeTemp=oldmodetemp;
ModeTemp2=GET_PH;
BarPulldown(k);
break;
case -6:
SYS1:
NumChoose=5;
ButtonTakeOver( );
ToView(2);
BarPulldown(k);
ModeTemp2=!GET_PH;
ModeTemp=GET_PH;
if(ToSys(No)==No) {
    ModeTemp2=GET_PH;
    ModeTemp=oldmodetemp;
    BarPulldown(k);
    i=0;
    break;
}
Make_Screen(6,Yes);
WindowExpand(maxx/18,maxy/5,maxx-maxx/18,maxy-maxy/6,
EGA_CYAN,0,3);
sh_image_voc(mx_4+3,maxy-maxy/6-79);
ModeTemp=oldmodetemp;

```

```

ModeTemp2=GET_PH;
NumPullDown=j;
maxphase=maxphase2;
insert_status( );
saymode(ENG);
repulldown=1;
i=-6;
break;
case INS: //add a new phase
    if(VocMode==DIC_PH) {
        i=0; //clear kb-code
        break;
    }
    VocBarAtt(7); //ADD-PH area
    ModeTemp=oldmode+temp;
    ToView(2);
    BarPullDown(k);
    ModeTemp2=!GET_PH;
    if(ModeTemp==FULL_PH) {
        tone( );
        post_notice("<เต็ม>");
        i=INS;
    }
    else {
        if(ChkYN("ท่านต้องการเพิ่มคำศัพท์ใหม่ <Y/N> ?","No")!=Yes)
            i=INS;
    }
    ModeTemp2=GET_PH;
    BarPullDown(k);
    break;
case -7:
    i=ESC;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

case ESC: //exit the current mainword-vocab

    NumChoose=6;

    ButtonTakeOver( );

    ToView(2);

    break;

default: break;

}

if(i=='\r' || i==6) break;

if((j>max_lim && j%(max_lim+1)==0 && i==DN) ||
(j+1>max_lim && (j+1)%(max_lim+1)==0 && i==UP)) {
    //window used view(3)
    ToView(3);

    setwritemode(COPY_PUT);

    bar_del(3,34,maxx,maxy,DARKGRAY);

    setcolor(YELLOW);

    settextjustify(LEFT_TEXT, CENTER_TEXT);

    for(n=0,m=p*(max_lim+1); m<=(p+1)*(max_lim+1); m++){
        outtextxy(maxx/8, 43+(20*n++),PhasNam+(m*32));
        if(m==maxphase2) {
            setcolor(DARKGRAY);

            showchar(15,maxy-22,(unsigned char)252); //DN arrow

            break;
        }

        showchar(15,maxy-22,(unsigned char)252); //DN arrow
    }

}

ToView(2);

}

ToView(2);

    BarPullDown(k);

    while(m_events->flag && m_events->flag==0x02);

    ModeTemp=GET_PH;

    if(i==INS || i==ESC || i=='\r' || i==6)

```

```

        break;
    }
}while(i==6); //i==6 , goto RePullDown
free(PhasNam);
} //maxphase!=NEW_PH

ModeTemp=oldmode&temp;
ModeTemp2=lGET_PH;
setwriteMode(COPY_PUT);

if(i==INS || maxphase==NEW_PH) {
    if(ModeTemp==FULL_PH) {
        m_hide( );
        Make_Screen(3,0);
        ToView(2);
        return FULL_PH;
    }
    if(maxphase==NEW_PH) {
        PhaseSupport(No);
        ModeTemp=NEW_PH;
        return NEW_PH;
    }
    else {
        ModeTemp = ADD_PH;
        PhaseSupport(Yes);
        return ADD_PH;
    }
}

//-----case of ENTER, ESC ----
NumPullDown=j; // for return from DIC-mode
Set_Graphic_Cursor(PAUSED);
ToView(2);

```

```

m_hide( );
WindowHide(maxx/18,maxy/12+4,maxx-maxx/18+2,maxy-maxy/6,
bkcolor,bkcolor,0);
if(i==ESC) { //exit from DIC-mode and WR-mode
    WindowExpand(maxx/18,maxy/5,maxx-maxx/18,maxy-maxy/6,EGA_CYAN,0,3);
    sh_image_voc(mx_4+3,maxy-maxy/6-79);
    Make_Screen(10,0);
    ModeTemp=DIC_PH;
    return DIC_PH;
}
strcpy(ExitCode,"1");
return j;
}

int lineptr[ ]={0,1,2,3,4};

int LookForPhase(int maxphase, int mode) { //mode0=dic & add-dic
// the 4 phases==> maxphase=3 //mode1==study
//mode2==game-test
int (*validfunc)( ), j, c,
maxx2, maxy2, takeover,
oldmodetemp=ModeTemp, field_ctr, el,
field_ptr=0;

char *bfptr, *bfptr2, *killerror, *BuffToSave, *ExitCode, *msg,
*MsgLine[ ] = {
"คำศัพท์หลัก",
"จำนวนกริยาวิเศษณ์ หรือสะแลง",
"ความหมายทั่วไป",
"ตัวอย่างการใช้งาน",
"คำแปล"

```

```

};
char user_name[30], user_alpha[2], user_itemunit[5],user_orgunit[5],
    user_endunit[5];
NumPress=-1;
NumChoose=-2;

if(mode==0)
    takeover=TakeOverDbFile( TellFileNo(MAIN_WORD,1));
else {
    msg="#####";
    msg=File2Message("usertemp.dat");
    *(msg+30)='\0';
    *(msg+32)='\0';
    *(msg+38)='\0';
    UsersName="#####";
    UsersAlpha="##";
    UsersItemUnit="#####";
    strcpy(user_name,msg);
    strcpy(user_alpha,msg+31);
    strcpy(user_itemunit,msg+33);
    GetOrgEndUnit(user_alpha,user_orgunit,user_endunit);
    OpenDBase(user_alpha,1,Yes); //l=vocphase
}

if(mode==0 && takeover==Yes) {
    if(mode==0) { //case of No New phase
        bfptr = bf + epos(elist[0],elist);
        el = *(elist + field_ptr)-1;
        validfunct = sb[0].edits;

        RePullDown:
        strcpy(ExitCode,"0");
        j=PhasePullDown(bfptr,maxphase,ExitCode);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(atoi(ExitCode)==0) //ESC or ModeTemp
            return j;          // exit and return ModeTemp
    }

    //ESLE RETURN J FOR CALCULATE A NUMBER OF PHASE-LIST
    bfptr2 = &phase_no[maxphase-j][0];
    strcpy(bfptr,bfptr2);
} //////IF(mode==0 && maxphase!=NEW_PH)

if( VocMode==DIC_PH) { //VocMode=DIC_PH : Study & Dic
    if(mode==0) { //VocMode=WR_PH : Edit Dbase
        m_hide( );
        Make_Screen(8,No);
    }
    if(mode==1) { //mode1
        ModeTemp=GET_STU;
        field_ctr=no_flds( );
        field_ptr=field_ctr;
        bfptr = bf + epos(elist[0],elist);
        el = *(elist + field_ptr)-1;
        validfunct = sb[0].edid;
        data_coord(el+1);
        strcpy(bfptr,user_itemunit);

        m_hide( );
        Make_Screen(8,No);
        sh_draw1(maxx-maxx/18-95,maxy/5-59,"learn.drw");
        ////TEMP
        say_command("ไปรคพทลคคหนำถคไป");
    }
}
do { //if(mode==1 && c==80) ==> LOOPING
    wt_screen=0;

```

```

(*validfunc)(bfptr);
wr_screen=1;
NumChoose=NumPress=-1; //for check first time
ToView(2);
m_hide( );
Set_Graphic_Cursor(GLOVE);
settextjustify(LEFT_TEXT, BOTTOM_TEXT);
do {
    ShPhaseDic(bfptr,mode); //j used to calculate line of "=>"
j=0;
    ThaiStrShade(9,lineptr[j]-21, "=>",WHITE,BLACK,2,No);
do {
    m_show( );
    switch(c=GetKeyMon(-20,-20)) {
        case UP:
        case DN:
        case -4:
        case -5:
            CURSOR:
            ToView(2);
            ThaiStrShade(9,lineptr[j],"=>",EGA_CYAN,EGA_CYAN,2,No);
            ThaiStrShade(9,(j==2||j==4)?lineptr[j]:lineptr[j]-21,
            "=>",EGA_CYAN,EGA_CYAN,2,No);
            if(c==DN || c==5) {
                j=++j>4?0:j;
                NumChoose=4;
            }
            else{
                j=--j<0?4:j;
                NumChoose=3;
            }
            ThaiStrShade(9,(j==2||j==4)?lineptr[j]:lineptr[j]-21,

```

```

"=>",WHITE,RED,2,No);
say_command((j==0 && mode==1)?"":MsgLine[j]);
ButtonTakeOver( );
ToView(2);
break;

case ' ':
case '\t':
    ButtonTab(6);
    break;

case F1:
case -1:
case -2:
    H_A:
    NumChoose=c==2?1:0;
    ButtonTakeOver( );
    HelpAbout(c);
    break; //goto the line of switch( )

case BS:
case FWD: c=c==FWD?-91:-90;
case -90:
case -91: //learn.drw
    if(mode==1) {
        Sh_NextPage(c);
        if(strcmp(bfptr, (c==91)?user_endunit:user_orgunit)==0) { //chk org/end unit? 0=
end

        if(ChkYN("หมด! กด'Y'หรือเลือก'1'เลือกอักษรใหม่",No)==Yes) {
            c=-91; //use -91 replace -90 , for check exit only
            break;
        }
    }
    else {
        c=0;

```

```

        break;
        //goto INPUT;
    }
}
else {
    strcpy(bfptr,IncDecNumStr(bfptr,(c==91)?1:0));
    strcpy(UsersItemUnit,bfptr);
    c=-80;
    break;
    //goto NextStudy;
}
}
else //mode==0
    c=0; //goto INPUT
break;
case '\r':
    switch(NumChoose) {
        case 0:
        case 1:
            c=NumChoose==0?-1:-2;
            goto H_A;
        case 2:
            c=-3;
            goto FONT;
        case 3:
        case 4:
            c=NumChoose==3?-4:-5;
            goto CURSOR;
        case 5:
            goto SAVEDIC;
        case 6:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

        c=ESC;
        break;
    default:
        break;
}
break;

case -3:
    FONT:
    NumChoose=2;
    ButtonTakeOver( );
    ToView(2);
    switch(FontStyle) {
        case 0: FontStyle=5; //script
            FontSize=4; //n=fontsize
            break;
        case 5: FontStyle=1; //trip
            FontSize=3;
            break;
        case 1: FontStyle=10; //bold
            FontSize=1;
            break;
        case 10: FontStyle=0; //default
            FontSize=1;
            break;
    }
    break;

case -6:
    SAVEDIC:
    NumChoose=5;
    ButtonTakeOver( );
    if((BuffToSave=(char *) malloc(400)) == NULL) {
        error_message("หน่วยความจำไม่พอ");
    }

```

```

        break;
    }
    sprintf(BuffToSave, "\n\n%s ==> %s\n%s ==> %s\n%s"
        " ==> %s\n%s ==> %s\n%s\n%s ==> %s\n",
        MsgLine[0], MAIN_WORD, MsgLine[1], ToFieldNo(bfptr,2),
        MsgLine[2], ToFieldNo(bfptr,3), MsgLine[3],
        ToFieldNo(bfptr,4), ToFieldNo(bfptr,5), MsgLine[4],
        ToFieldNo(bfptr,6));
    SaveDataUser(BuffToSave,0,Yes); //0==phase-dic-user
    free(BuffToSave);
    break;
case -7:
case ESC:
    NumChoose=6;
    ButtonTakeOver( );
    if(mode==1) { //study
        if((c=ChkYN("ออก..และบันทึกบทเรียน<Y/N> ?",
            No))==Yes) {
            UnitUserFile(user_name,user_alpha,bfptr);
            //bfptr==current unit
c=-91; // if c=-90/-91, give c=-91 for check exit
            break;
        }
        else if(c==Cancel) {
            c=0; // goto INPUT;
            break;
        }
        else if(c==ESC) {
            c=0;
            break;
        }
        else {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        c=-91; //goto ExitFunc;
        break;
    }

}

default: break;
}

}while(c!=ESC && c!=-7 && c!=-3 && c!=-91 && c!=-80);
m_hide( );
} while(c==3); //check c==3 ?
}while(mode==1 && c==80); //check c==80
    if(mode==1 && c==91 && c==ESC && c==7) {
        if(image_voc!=NULL)
            free(image_voc);
        ToView(2);
        bar_del(2,maxy/5,maxx-2,maxy-maxy/6+2,LIGHTGRAY);
        return 1; //EXIT FUNCTION
    }
    ToView(2);
    if(mode==0) {
        WindowHide(5,maxy/5,maxx-5,maxy-maxy/6,bkcolor,bkcolor,0);
        WindowExpand(maxx/18,maxy/5,maxx-maxx/18,maxy-maxy/6,EGA_CYAN,0,3);
        sh_image_voc(mx_4+3,maxy-maxy/6-79);
        ModeTemp=oldmodetemp;
        goto RePullDown;
    }
} //..IF(VocMode==DIC_PH)
else { //VocMode=WR_PH
    PhaseSupport(No);
    display_template( );
    (*validfunc)(bfptr);
    ToView(2);
}

```

```

ModeTemp=EDIT_PH;
return EDIT_PH;
} //IF(MAXPHASE!=NEW_PH)
ToView(2);
bar_del(0,maxy/5,maxx-4,maxy-maxy/6+2,LIGHTGRAY);
return 1;
}

```

```

void ShPhaseDic(char *bfptr,int mode) { //mode0=DIC mode1=Study
int i, j, y; //, lineptr[ ]={0,1,2,3,4};
char *bfptr3, *bfptr4, examp[3],
*MsgLine[ ] = {
    "คำศัพท์หลัก",
    "จำนวนกริยาวิ หรือสะกด",
    "ความหมายทั่วไป",
    "ตัวอย่างการใช้งาน",
    "คำแปล"
};

setviewport(view2.left,view2.top,view2.right,view2.bottom,1);
bar_color(5,maxy/5,maxx-5,maxy-maxy/6,EGA_CYAN,WHITE,2);
setviewport(view2.left,view2.top,view2.right-8,view2.bottom,1);
settextstyle(FontStyle,HORIZ_DIR,FontSize);

y=25;
m_hide( );

if(mode==0) //MAIN_WORD is shown in DIC-mode only
    OutTextXYShade(maxx/23+10,lineptr[0]=maxy/6+y+20,
    MAIN_WORD,EGA_LIGHTCYAN,BLACK,2);
y+=30;
bfptr3=ToFieldNo(bfptr,2);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        i=strlen(bfptr3);
        do --i; while(*(bfptr3+i) == ' ' && *(bfptr3+i)!='\0');
if(i>26) {
    i=28;
    do --i; while(*(bfptr3+i) != ' ' && *(bfptr3+i) != '\0');
    examp[0] = *(bfptr3+i);
    *(bfptr3+i)='\0';
    OutTextXYShade(maxx/23+10,lineptr[1]=maxy/6+y+20,bfptr3,
    EGA_LIGHTCYAN,BLACK,2);
    *(bfptr3+i) = examp[0];
    bfptr4=bfptr3+i+1;
    y+=30;
    OutTextXYShade(maxx/23+10,maxy/6+y+20,bfptr4,
    EGA_LIGHTCYAN,BLACK,2);
}
else {
    OutTextXYShade(maxx/23+10,lineptr[1]=maxy/6+y+20,bfptr3,
    EGA_LIGHTCYAN,BLACK,2);
    lineptr[2]=3;
}
y+=30;
ThaiStrShade(maxx/23+10,lineptr[2]=maxy/6+y+3,
ToFieldNo(bfptr,3),WHITE,BLACK,2,No);
y+=30;
bfptr3=ToFieldNo(bfptr,4);
i=strlen(bfptr3);
do --i; while(*(bfptr3+i) == ' ' && *(bfptr3+i)!='\0');
if(i>26) {
    i=28;
    do --i; while(*(bfptr3+i) != ' ' && *(bfptr3+i) != '\0');
    examp[0] = *(bfptr3+i);
    *(bfptr3+i)='\0';

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

OutTextXYShade(maxx/23+10,lineptr[3]=maxy/6+y+20,bfptr3,
EGA_LIGHTCYAN,BLACK,2);
y+=30;
*(bfptr3+i) = examp[0];
bfptr4=(bfptr3+i+1);
strcpy(bfptr4,ToFieldNo(bfptr,5));
i=strlen(bfptr4);
do --i; while(*(bfptr4+i) == ' ' && *(bfptr4+i)!='\0');
if(i>26) {
    i=28;
    do --i; while(*(bfptr4+i) != ' ' && *(bfptr4+i)!='\0');
    examp[0] = *(bfptr4+i);
    *(bfptr4+i)='\0';
    OutTextXYShade(maxx/23+10,maxy/6+y+20,bfptr4,
EGA_LIGHTCYAN,BLACK,2);
    *(bfptr4+i) = examp[0];
    bfptr3=(bfptr4+i+1);
    y+=30;
    OutTextXYShade(maxx/23+10,maxy/6+y+20,bfptr3,
EGA_LIGHTCYAN,BLACK,2);
}
else
    OutTextXYShade(maxx/23+10,maxy/6+y+20,bfptr4,
EGA_LIGHTCYAN,BLACK,2);
}
else // < 26
    OutTextXYShade(maxx/23+10,lineptr[3]=maxy/6+y+20,
ToFieldNo(bfptr,4),EGA_LIGHTCYAN,BLACK,2);
y+=30;
ThaiStrShade(maxx/23+10,lineptr[4]=maxy/6+y+5,
ToFieldNo(bfptr,6),WHITE,BLACK,2,No);
if(mode==0) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

sh_image_voc(mx_4+28,maxy/5+5);
say_command(MsgLine[0]);
}
ModeTemp=DIC_PH;
j=0;
ThaiStrShade(9,lineptr[j]-21, "=>",WHITE,BLACK,2,No);
}

```

```

int NoCountSpaceEnd(char *msg) {
//start 1..2..3...
int i;
i=strlen(msg)-1;
while(*(msg+i-1)==' ')
--i;
return(i);
}

```

```

void Sh_ChooseButton(int NumPress, int NumChoose) {
ToView(1);
if(NumPress!=-1) //no-press
BarAttrib(1,NumPress*IconHi,box_width,
(NumPress+1)*IconHi,15,XOR_PUT);
BarAttrib(1,NumChoose*IconHi,box_width,
(NumChoose+1)*IconHi,15,XOR_PUT);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void Sh_ReleaseButton(int NumChoose) {
    if(NumChoose!=-1) { //this status ==> no-press
        ToView(1);
        BarAttrib(1,NumChoose*IconHi,box_width,(NumChoose+1)*IconHi,15,XOR_PUT);
    }
}

```

```

void ButtonTakeOver(void) { //out
    if(NumPress!=NumChoose){
        Sh_ChooseButton(NumPress,NumChoose);
        DelayChkKey(0.12);
        KeyMouRelease( );
        NumPress=NumChoose;
    }
    Sh_ReleaseButton(NumChoose);//show animation
    NumPress=NumChoose-1;
}

```

```

void ButtonTab(int max) {
    NumPress=(++NumPress>max)?0:NumPress;
    Sh_ChooseButton(NumPress,NumChoose);
    NumChoose=NumPress;
}

```

```

void KeyMouRelease(void) {
    m_show( );
    while(kbhit( )!=0)
        getch( ); //clear kb_buffer
    //wait for release mouse
}

```



```

while(m_events->flag && m_events->flag==0x02);
m_hide( );
}

```

```

void ButtonMidTakeOver(int oldbutt,int newbutt) {
int j,i,m;
  ToView(1);
  for(j=0; j<(m=oldbutt==newbutt?1:3); j++) {
    if(m==1)
      i=oldbutt;
    else {
      if(j==0)
        i=oldbutt;
      else
        i=newbutt;
    }
    switch(i) {
  case 2:
    case 3:
      BarAttrib(i<3?maxx/2-43:maxx/2+52, maxy/3+41,
        i<3?maxx/2+51:maxx/2+147,maxy/3+116,WHITE,XOR_PUT);
      break;
    case 4:
  case 5:
      BarAttrib(i<5?maxx/2-43:maxx/2+52, maxy/3+115,
        i<5?maxx/2+51:maxx/2+147,maxy/3+190,WHITE,XOR_PUT);
      break;
    case 0:
    case 1:
    case 6:

```

```

BarAttrib(1,i*IconHi,box_width,(i+1)*IconHi,WHITE,XOR_PUT);
break;
default:
break;
}
if(j>=1)
KeyMouRelease( );
}
NumChoose=NumPress=-1;
}

void OutTextXYShade(int x, int y, char *msg,int color,
int narrow, int num) {
int i;
for(i=0; i<=num; i++) {
setcolor(i<num?narrow:color);
outtextxy(x-i,y-i,msg);
}
}

void PhaseSupport(int hide) {
m_hide( );
if(hide!=0) {
WindowHide(view3.left-view2.left, view3.top-view2.top,
view3.right-view2.left, view3.bottom-view2.top, EGA_CYAN,YELLOW,1);
WindowHide(maxx/18,maxy/5,maxx-maxx/18+2,maxy-40,bkcolor,0,3);
}
WindowExpand(5,maxy/5,maxx-5,maxy-maxy/6,EGA_CYAN,WHITE,2);
setcolor(CH_TEMPLATE);

```

```

ThaiStr(maxx/4-18,maxy/5+5,"ไฟล์ข้อมูล"); //filename
ThaiStr(14,maxy/4+20,"คำศัพท์หลัก");
ThaiStr(maxx/4-18,maxy/4+20,MAIN_WORD);
Make_Screen(3,0);
ToView(2);
}

```

```

void BarPulldown(int j) {
    setwritemode(XOR_PUT);
    setcolor(RED);
    BarLine(mx5+40,my5+56+(j*20),mx_5-4,my5+76+(j*20));
}

```

```

int TellFileNo(char *mainword,int groupfile) {
//dicword*:groupfile==0
//vocphas*:groupfile==1
//students:groupfile==2
int c;
if(((c=(int)(tolower(*(mainword) ))) >=97 && c<=100) ||
(c>=119 && c<=122)) //a,b,c,d,w,x,y,z
    c=0;
else if((c>=108 && c<=112) || (c>=115 && c<=118)) //l,m,n,o,p,s,t,u,v
    c=1;
else    c=2; //b,f,g,h,i,j,k,q,r
return(c=groupfile==0?c:x+3);
}

```

```

int TakeOverDbFile(int file_no) {
char *fileopen[8] = {

```

```

"dicwords",
"dicword2",
"dicword3",
"vocphase",
"vocphas2",
"vocphas3",
"students"
};

clear_record( );

db_cls( ); // close the previous db_file
ClearMemData( );

if ((file = filename(fileopen[file_no])) == ERROR) {
    error_message("Invalid filename");
    return No;
}
else {
    len = rlen(file);
    els = file_els [file];
    sc = malloc(len); /* screen buffer */
    rb = malloc(rlen(file)); /* record buffer */
    hb = malloc(rlen(file)); /* hold buffer */
    if (sc == (char *) 0 ||
        rb == (char *) 0 ||
        hb == (char *) 0) {
        printf("\nOut of memory");
        abort( );
    }

    init_rcd(file, rb);
    init_rcd(file, hb);
    init_screen(fileopen[file_no], els, sc);
    *fl = file;
    db_open("", fl);
}

```

```

    clrcred(sc, cla);
    return Yes;
}
}

```

```
void BarLine( l, t, r, b)
```

```
register t, b, l, r;
```

```

{
    setlinestyle(0,0,1);
    for(--t; t++<=b;)
        line(l,t,r,t);
}

```

```
void CursorShape(top, bottom)
```

```
int top, bottom;
```

```

{
    union REGS r;
    r.h.ch = top;
    r.h.cl = bottom;
    r.h.ah = 1;
    int86(0x10, &r, &r);
}

```

```
int GetKeyMou(int gx,int gy)
```

```

{
    register int x,y,step,mx,my;
    int c;
    do {
        sh_cursor(gx,gy+14,say_mode);

```

```
-
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(kbhit( )) {
    c=(c=getch( ))==0 ? getch( )|128:c;
    break;
}
if(m_events->flag && m_events->button==1 ) {
    ToView(1);
    if(m_events->xaxis<=box_width) {
        if(m_events->yaxis<VStep) {
            c=-1;
            break;
        }
        else if(m_events->yaxis<2*VStep) {
            if(ModeTemp!=EDIT_PH && ModeTemp!=ADD_PH)
                c=-2;
            else
                c=-25; //UP
            break;
        }
        else if(m_events->yaxis<3*VStep) {
            if(ModeTemp==GET_VOC)
                break;
            c=-3;
            break;
        }
        else if(m_events->yaxis<4*VStep) {
            switch(ModeTemp) {
                case DIC_PH:
                case GET_MAIN:
                    c=-4;
                    break;
                case GET_VOC:
                    break;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    case GET_WORD:
    case GET_PH:
        c=-14;
        break;
    default:
        c=-44;
        break;
}
}
else if(m_events->yaxis<5*VStep) {
    switch(ModeTemp) {
    case GET_VOC:
        break;
    case DIC_PH:
        c=-5;
        break;
    case GET_MAIN:
    case GET_WORD:
    case GET_PH:
        c=-15;
        break;
    default:
        c=-55;
        break;
    }
}
else if(m_events->yaxis<6*VStep) {
    if(ModeTemp==GET_VOC)
        break;
    c=-6;
    break;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if(m_events->yaxis>6*VStep) {
    c=-7;
    break;
}
else { //clicked out of area
    ToView(2);
    moveto(gx,gy);
}
}

if(ModeTemp==DIC_PH) {
    if(m_events->yaxis>=84 && m_events->yaxis<=144 &&
        m_events->xaxis>=504 && m_events->xaxis<=584) {
        if(m_events->xaxis<544)
            c=-90;
        else
            c=-91;
        break;
    }
}

if(ModeTemp==GET_VOC && HELPABOUT!=1) {
    ToView(1);
    if(m_events->yaxis>=maxy/3+42 &&
        m_events->yaxis<=maxy/3+115 &&
        m_events->xaxis>=maxx/2-42 &&
        m_events->xaxis<=maxx/2+147) {
        if(m_events->xaxis < maxx/2+53) {
            c=-3; //DIC-mode
            break;
        }
        else {
            c=-4; //LEARN-mode
            break;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

    }
}
if(m_events->yaxis>=maxy/3+116 &&
    m_events->yaxis<=maxy/3+189 &&
    m_events->xaxis>=maxx/2-42 &&
    m_events->xaxis<=maxx/2+147) {
    if(m_events->xaxis < maxx/2+53) {
        c=-5; //ADD-mode
        break;
    }
    else {
        c=-6; //TEST-mode
        break;
    }
}
else
    c=0;
    break;
} //end GET-VOC

if(VocMode==WR_PH && HELPABOUT!=1) {
    if(m_events->yaxis>=104 &&
        m_events->yaxis<=125 &&
        m_events->xaxis>=321 &&
        m_events->yaxis<=395) {
        c=INS; //ADD-PHASE AREA
        break;
    }
}

ToView(2);
y=maxy/3-20;
if(m_events->yaxis>=y+maxy/8-7+view2.top &&

```

```

m_events->yaxis<=y+maxy/5-12+view2.top &&
HELPABOUT!=1 ) { //check Yes/No/Cancel
    if(m_events->xaxis>=maxx/4+4+view2.left &&
        m_events->xaxis<=maxx/4-3+maxx/6+view2.left) {
        c=-99;
        break;
    }
    else if(m_events->xaxis>=maxx/4+4+maxx/6+view2.left &&
        m_events->xaxis<=maxx/4-3+(2*maxx/6)+view2.left) {
        c=-98;
        break;
    }
    else if(m_events->xaxis>=maxx/4+4+(2*maxx/6)+view2.left &&
        m_events->xaxis<=maxx/4-3+(3*maxx/6)+view2.left) {
        c=-97;
        break;
    }
    else { //clicked out of area
        ToView(2);
        moveto(gx,gy);
    }
}
if(HELPABOUT==1) { //check HelpAbout
    ToView(4);
    if(m_events->xaxis>=view4.left+maxx/3-14 &&
        m_events->xaxis<=view4.right-maxx/3-9) {
        if(m_events->yaxis>view4.top+106 &&
            m_events->yaxis<view4.top+134) {
            c=-92; //Datafile input
            break;
        }
        if(m_events->yaxis>view4.top+134 && m_events->yaxis<view4.top+162) {

```

```

        c=-95;
        break;
    }
    if(m_events->yaxis>view4.top+162 && m_events->yaxis<view4.top+190) {
        c=-94;
        break;
    }
    if(m_events->yaxis>view4.top+190 && m_events->yaxis<view4.top+218) {
        c=-93;
        break;
    }
}
}
}
}while(c==0);
ToView(2);
moveto(gx,gy);
return c;
}

void bar_color(int l, int t, int r, int b, int colorgnd,
int colorbox,int boxmode)
{
    int c;
    c=getcolor( );
    setcolor(colorgnd);
    setfillstyle(1,colorgnd);
    setlinestyle(0,0,1);
    bar(l, t, r, b);
    setwriteMode(COPY_PUT);
}

```

```

if(boxmode==1) {
    setcolor(colorbox);
    rectangle(l+1, t+1, r-1, b-1);
}
else if(boxmode==2) {
    setcolor(WHITE);
    line(l,t,r,t);
    line(l+1,t+1,r-1,t+1);
    line(l,t,l,b);
    line(l+1,t+1,l+1,b-1);
    setcolor(BLACK);
    line(r,t,r,b);
    line(r-1,t+1,r-1,b-1);
    line(l,b,r,b);
    line(l+1,b-1,r-1,b-1);
}
else if(boxmode==3) {
    setcolor(BLACK);
    line(l,t,r,t);
    line(l,t+1,r,t+1);
    line(l,t,l,b);
    line(l+1,t,l+1,b);
    setcolor(WHITE);
    line(r,t+1,r,b);
    line(l+1,b,r,b);
}
else if(boxmode==4) {
    setcolor(BLACK);
    line(l,t,r,t);
    line(l+1,t+1,r-1,t+1);
    line(l,t,l,b);
    line(l+1,t+1,l+1,b-1);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

line(l,b-1,r,b-1);
line(l+1,b-2,r-1,b-2);
line(l+2,b-3,r-2,b-3);
line(r-2,t,r-2,b-2);
line(r-3,t+1,r-3,b-3);
line(r-4,t+2,r-4,b-4);
setcolor(WHITE);
line(l+2,t+2,r-4,t+2);
line(l+3,t+3,r-5,t+3);
line(l+2,t+2,l+2,b-3);
line(l+3,t+3,l+3,b-4);
line(l+1,b,r,b);
line(r,t+1,r,b);
line(r-1,t+2,r-1,b-1);
}
setcolor(c);
}

```

```

void Make_Screen(int mode_icon, int screen)

```

```

{
int i,j,n;
char *item = " ";
char *NoItem = " ";
int voc_drw[ ]={17,18,6,19};

```

```

char *myicon[ ] = {
    "help.drw", //0 0
    "aboutdic.drw", //1 1
    "rework.drw", //3 2
    "instdat.drw", //4 3
    "system.drw", //5 4

```



```

"out.drw", //6 5
"addmw.drw", //2 6
"okay2.drw", //2 7
"save.drw", //1 8
"cancel.drw", //6 9
"arrowdn.drw", //3 10 //A
"thaieng.drw", //3 11 //B
"insowr.drw", //4 12 //C
"drawers.drw", //5 13 //D
"arrowup.drw", //4 14 //E
"rodent.drw", //5 15 //F
"vocab.drw", //2 16 //G
"findmw.drw", //4 17 //H
"learn.drw", //3 18 //I
"game.drw", //4 19 //J
"fonts.drw", //2 20 //K
"garbage.drw", //3 21 //L
"registra.drw", //5 22 //M
"arrowri.drw", //2 23 //N
"arrowlf.drw" //3 24 //O
};
ToView(1);
if(screen!=0) {
    WindowExpand(box_width+1,0,maxx+1,maxy+1,EGA_CYAN,0,0);
    bar_color(box_width+7,bar_height+8,maxx-6,maxy-6,bkcolor,0,4);
    ShowView2( );
    //shade around get_word_box
    if(ModeTemp!=GET_PH && ModeTemp!=GET_MAIN)
    WindowExpand(maxx/18,maxy/5,maxx-maxx/18,maxy-maxy/6,EGA_CYAN,0,3);
    ToView(1);
    /// show institute-name
    setwritemode(COPY_PUT);

```

```

WindowExpand(box_width+5,5,maxx-5,bar_height+1,LIGHTRED,0,0);
bar_color(box_width+5,5,maxx-5,bar_height+1,LIGHTRED,0,2);
sh_draw1(box_width+10,bar_height/3-2,"kng.drw");
ThaiStrShade(mx2+box_width/2,11,"สถาบันเทคโนโลยีพระจอมเกล้าเจ้า
"อาภุคทหารลาดกระบัง",WHITE,BLACK,3,Yes);
ThaiStrShade(mx2+box_width/2,maxy/22+7,"ภาควิชาครุศาสตร์วิศวกรรม",
WHITE,BLACK,3,Yes);
bar_color(0,0,box_width,maxy,GREEN,GREEN,0); //box of icon
setcolor(BLACK);
rectangle(1,1,box_width+1,maxy);
setcolor(LIGHTGREEN);
rectangle(0,0,box_width,maxy-1);
}
switch(mode_icon) {
case 20://show Valeo_image // see upper line
case 1://Main option
strcpy(item,"01G2F45");
break;
case 2://Get Mainword
strcpy(item,"017LF45");
break;
case 3://Edit Phase
strcpy(item," E BC48");
break;
case 4://Choose Phase list
strcpy(item," EA 9");
break;
case 5://exit DOS to Edit Phase
strcpy(item,"0E7BC48");
break;
case 6://exit DOS to choose phase list
//the rest will be made in LookForPhase( ) //"017 4 "

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        strcpy(item,"017EA49");
break;
case 7: //choose vocab-mode style
        //the rest will be make in LookForPhase( )
        strcpy(NoItem," **** ");
        strcpy(item,"01G2F45");
ToView(2);
        WindowExpand(maxx/18,maxy/5,maxx-maxx/18,maxy-maxy/6,
        EGA_CYAN,WHITE,2);
        bar_color(maxx/18,maxy/5,maxx-maxx/18,maxy-maxy/6,
        EGA_CYAN,WHITE,4);
ToView(1);
m_hide( );
for(i=0; i<3; i++) { //creat vocab-option /// TEMPORARY
        bar_color(i%2==0?maxx/2-42:maxx/2+53,
        i<2?maxy/3+42:maxy/3+116, i%2==0?maxx/2+52:maxx/2+147,
        i<2?maxy/3+115:maxy/3+189,EGA_CYAN,WHITE,4);
        sh_draw1(i%2==0?maxx/2-35:maxx/2+60,
        i<2?maxy/3+49:maxy/3+123,myicon[voc_drw[i]]);
}
m_show( );
break;
case 8: //exit DOS to DIC-MODE
strcpy(item,"01KEAD9"); //the rest will be make in LookForPhase( )
        break;
case 9: //return From DIC-MODE to Choose phase list
        strcpy(item,"017 4 ");
        break;
case 10:
        strcpy(item," LF 5"); //from Choose phase list to MainWord
        break;
case 11:

```



```

strcpy(item," 1 LF 5"); //from EDIT_PH to MainWord
break;
case 12:
strcpy(item,"EANO7M5"); //from EDIT_PH to MainWord
break;

default:
break;
}

if(mode_icon!=0) { // ignore display icon
if(m_events->xaxis<box_width) m_hide( );
for(i=0; i<7; i++) {
if(*(item+i)!=' ') { //buttons of icons
bar_color(4, 3+VStep*i, box_width-2, VStep*(i+1)-1,
EGA_CYAN,BLACK,2);
setwriteMode(COPY_PUT);
setcolor(BLACK);
rectangle(7,5+VStep*i,box_width-4,VStep*(i+1)-3);
}
if(*(item+i) !=' ' && *(item+i)!='*') {
j=(j=(int)(*(item+i)))>=65?j-55:j-48; //j-55=cap j-48=num
sh_draw1(6,5+VStep*i,myicon[j]);
}
if(*(NoItem+i)=='*') { //Delete icon
setwriteMode(COPY_PUT);
setlinestyle(0,0,1);
setcolor(YELLOW);
pieslice(box_width-12, 12+VStep*i,0,180,5);
pieslice(box_width-12, 12+VStep*i,180,360,5);
setcolor(BLACK);
circle(box_width-12, 12+VStep*i,5);
line(box_width-17, 12+VStep*i,box_width-7, 12+VStep*i);
}
}
}

```

```

    }
}
strcpy(NoItem," "); //clear data
}
ToView(2);
}

```

```

void CheckSys(void)
{
int c;

setviewport(0,0,maxx,maxy,1);
getviewsettings(&view1);
setviewport(maxx/3+box_width/2,maxy/4+5,maxx-maxx/3+box_width/2,
maxy-maxy/4+5,1);
getviewsettings(&view4);
setviewport(view1.left,view1.top,view1.right,view1.bottom,1);
sh_draw1(maxx/3+20,maxy/3+20,"phasetit.drw");

VStop = maxy/7;
m_eventz = MK_FP(getpsp( ), 0x00C0);
initialize_cursors( );
rodent = m_reset( );
if(rodent->present) {
    m_inst_task(EVENTMASK, FP_SEG(m_handler), FP_OFF(m_handler));
    Set_Graphic_Cursor(GLOVE);
    m_moveto(maxx/8,2);
    m_show( );
    m_xlimit(1,maxx);
    m_ylimit(1,maxy);
    m_move_ratio(HorFast,VerFast);
}
}

```

```

}
IsDamage( ); //check files damage, if found, asked user to exit
m_moveto(maxx/2,maxy-maxy/8);
Pause( );
m_hide( );
WindowHide(0,0,maxx+1,maxy+1,bkcolor,RED,1);
WindowExpand(0,0,maxx+1,maxy+1,BLACK,0,0);
setviewport(box_width+12,bar_height+12, maxx-10,maxy-10,1);
getviewsotings(&view2);
m_hide( );
Make_Screen(20,1);
ToView(1);
m_hide( );
sh_draw1(box_width+10,bar_height/3-2,"kng.drw");
ToView(2);
say_command("MAIN MENU"); //MAIN MENU
ModeTemp=GET_MAIN;
NumPress=NumChoose=-1;

do {
m_show( );
switch(c=GetKeyMou(-30,-30)) {
case '\':
ButtonTab(6);
break;
case '\r':
switch(NumChoose) {
case 0:
case 1:
case 4: //mouse
if(NumChoose<2)

```

```

        c=NumChoose==0?-1:-2;

    else
        c=-15;
        goto H_ABOUT;

    case 2:
    case 3:
        c=NumChoose==2?-3:-4; // -3=query and -4=rework
        goto DB;

    case 5:
        goto SYS;
    case 6:
        goto EXIT;
    default:
        break;
}
break;

case F1:
case -1:
case -2:
case -15: //mouse

H_ABOUT:
    if(c==2)
        NumChoose=1;
    else if(c==15)
        NumChoose=4;
    else
        NumChoose=0;

ButtonTakeOver( );
HelpAbout(c);

break;

```

```

//FINE, LEARN, TEST
case -3: //j=0
case -4:
    DB:
    NumChoose=c==3?2:3;
    ButtonTakeOver( );
    if(DB_Manager(c)!=-1) { //check cancel
        // ValueImage( );
        ToView(2);
    }
    break;
case -6:
    SYS:
NumChoose=5;
    ButtonTakeOver( );
    ModeTemp=GET_MAIN;
    if(ToSys(No)==Yes) {
        m_hide( );
        Make_Screen(20,1);
        say_command("เมนเมนู"); //MAIN MENU
    }
    m_show( );
    break;
case ESC:
case -7:
    EXIT:
        c=ESC;
        NumChoose=6;
        ButtonTakeOver( );
    if(ChkYN("ท่านจะออกจากโปรแกรม<Y/N>?",No)!=Yes)
        c=0;
        break;

```

```

default: break;
}
}while(c!=ESC);
}

```

```

int DB_Manager(int mode) {
    m_hide( );
    switch(mode) {
    case -3: ///QUARY
        if(ChkYN("ท่านจะเข้าเมนูหลักทันที <Y/N> ?",No)==Yes) {
            m_hide( );
            VocabStyle( );
        }
        else
            mode=-1;
        break;
    case -4: ///REWORK
        if(ChkYN("ท่านจะซ่อมปรุฐานข้อมูล<Y/N>?",No)==Yes){
            m_hide( );
            WindowExpand(maxx/18,maxy/5,maxx-maxx/18,maxy-maxy/6,
            LIGHTRED,BLACK,2);
            ThaiStrShade(maxx/2,maxy/2+maxy/4,"กรุณารอดักครู่..",
            YELLOW,BLACK,2,Yes);
            sh_draw1(maxx/2-50,maxy/2-30,"rework.drw");
            m_moveto(maxx-maxx/3,maxy/2);
            Set_Graphic_Cursor(PAUSED);
            m_show( );
            system("ph_util all");
            m_hide( );
            WindowHide(maxx/18,maxy/5,maxx-maxx/18,maxy-maxy/6,
            bkcolor,bkcolor,0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Set_Graphic_Cursor(GLOVE);
m_show( ); //set m_hide( ) in reindex( ) before
}
else
    mode=-1;
break;
default:
    break;
}
return(mode);
}

```

```

int ToSys(int MkCr) {
    int i,j,c,c2, curdrive;
    char curdir[124],oldpath[128],comspec[124];
    if(ChkYN("ท่านจะออกสู่ DOS ชั่วคราว<Y/N>?",
(ModeTemp==GET_WORD? Yes:No )==Yes) {
        m_hide( );
        curdrive=getdisk( );
        getcwd(curdir, 124);
        getviewsettings(&vp);
        VMode=getgraphmode( );
        restorecrtmode( );

        for(i=0; i++<79;)
            printf("*");
        printf("\n*          Type 'EXIT' to return to Phasal verb "
"database. . .          *\n");
        for(i=0; i++<79;)
            printf("*");
        system("COMMAND");
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setdisk(curdrive);
sprintf(oldpath,"cd\ %s",curdir);
system(oldpath);
setgraphmode(VMode);
switch(ModeTemp) {
    case GET_WORD:
        i=2; //i=icon
        break;
    case EDIT_PH:
    case ADD_PH:
        i=5;
        break;
    default: //GET_MAIN
        i=1;
        break;
}
m_hide( );
if(MkCrtl!=No) {
    Make_Screen(i,1);
    if(ModeTemp==EDIT_PH || ModeTemp==ADD_PH) {
        m_hide( );
        PhaseSupport(1);
        m_show( );
    }
}
Set_Graphic_Cursor(GLOVE);
return Yes;
}
return No;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

void DoVocab(void)
{
    ModeTemp=GET_WORD;
    ToView(2);
    //box around get_word
    m_hide( );
    WindowHide(maxx/18,maxy/5,maxx-maxx/18-1,maxy-maxy/6-1,BLACK,BLACK,0);
    WindowExpand(maxx/18,maxy/5,maxx-maxx/18,maxy-maxy/6,EGA_CYAN,0,3);

    //SHOW DIC-WR-MODE-PIC
    WindowExpand(mx_4+3,maxy-maxy/6-79,maxx-maxx/16-2,
        maxy-maxy/6-5,EGA_LIGHTBLUE,WHITE,4);
    sh_drawl(mx_4+11,maxy-maxy/6-71,VocMode==DIC_PH?"findmw.drw":
        "addmw.drw");
    if(image_voc!=NULL)
        free(image_voc);
    sv_image_voc(mx_4+3,maxy-maxy/6-79,maxx-maxx/16-2,maxy-maxy/6-5);
    Make_Screen(2, 0);
    ModeTemp=GET_WORD;
    do{
        OpenDBase(MAIN_WORD,ModeTemp==GET_WORD?0:1,No);
        query( );
        ClearMemData( );
    } while(FILE_CHG==1);
    ModeTemp=GET_VOC;
    Make_Screen(7, 0);
    m_hide( );
    bar_del(maxx/4,maxy/12+5,maxx-maxx/4,maxy/5-10,bkcolor);
    say_command("ค้นหา เรียงรู้ เพิ่มศัพท์ หรือทดสอบกันดีปะ");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char *GetString(unsigned char field_type, char *str_msk,
char *str_bfr, char *exit_code) {
    int c,oldmode=ModeTemp;

    strnset(str_bfr,(int)' ',strlen(str_bfr));
    ModeTemp=GET_STU;
    ToView(2);
    bar_del(mx5,maxy/3+5,maxx-maxx/4,maxy/3+45,CYAN);
    bar_color(mx5+4,maxy/3+11,mx5+8+(strlen(str_msk)*8),
maxy/3+37,CH_INVER,WHITE,3);
    do {
        prev_col=Xmsk;
        prev_row=Ymsk/* +3 */;
        cursor(prev_col,prev_row);
    } while((c=read_element((unsigned char)field_type,
(char *)str_msk,(char *)str_bfr))!='\r' && c!=ESC);
    if(c==ESC)
        strcpy(exit_code,"1");
    else
        strcpy(exit_code,"0");
    ModeTemp=oldmode;
    return(str_bfr);
}

```

```

void VocabStyle( ) {
    int c,i;
    char *vocmsg="ค้น เรียนรู้ เพิ่มคำศัพท์ หรือทดสอบ";
    char *temp;

    ModeTemp=GET_VOC;
    HELPABOUT=1;

```

```

Make_Screen(7, 0);
say_command(vocmsg);
NumPress=NumChoose=-1;
do {
m_show( );
switch(c=GetKeyMou(-30,-30)) {
case '\':

    i=NumPress=(++NumPress>6)?0:NumPress;
    if(i<2)
        Sh_ChooseButton(NumPress,NumChoose);
    else if(i==2) {
        ButtonMidTakeOver(i,-1); //-1==> cancel
        Sh_ReleaseButton(1);
    }
    else if(i>2 && i<=5) {
ButtonMidTakeOver(i-1,-1);
        ButtonMidTakeOver(i,-1);
    }
    else if(i==6) {
        ButtonMidTakeOver(i-1,-1);
        Sh_ReleaseButton(i);
        NumChoose=6;
    }
    NumChoose=NumPress=i;
    break;
case '\r':
    switch(NumChoose) {
        case 0:
        case 1:
            c=NumChoose==0?-1:-2;
            NumPress=c==2?1:0;

```

```

        goto H_ABOUT;
    case 2:
    case 4:
        if(NumChoose==2) {
            c=-3;
            goto RD;
        }
        else {
            c=-5;
            goto WR;
        }
    case 3:
        goto LEARN; //4=learn
    case 5:
        goto TEST; //6=test
    case 6:
        goto EXIT;
    default:
        break;
}
break;
case F1:
case -1:
case -2:
    H_ABOUT:
    NumChoose=c==2?1:0;
    ButtonMidTakeOver(NumPress,NumChoose);
    HELPABOUT=1;
    HelpAbout(c);
    HELPABOUT=!1;
    break;

```

```

//////////
//SEARCH//
//////////
case -3:
RD:
NumChoose=2;
ButtonMidTakeOver(NumPress,NumChoose);
VocMode = DIC_PH;
DoVocab( );
break;

//////////
//LEARN//
//////////
case -4:
LEARN:
NumChoose=3;
ButtonMidTakeOver(NumPress,NumChoose);
m_hide( );
VocMode=DIC_PH;
StudyUnit( );
m_hide( );
bar_del(maxx-maxx/18-95,maxy/5-59, //clear learn.drw
maxx-maxx/18-15,maxy/5-1,LIGHTGRAY);
ModeTemp=GET_VOC;
Make_Screen(7,0);
say_command(vocmsg);
break;

/////
//ADD//
/////

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case -5:
WR:
NumChoose=4;
ButtonMidTakeOver(NumPress,NumChoose);
VocMode = WR_PH;
DoVocab( );
break;

```

```

//////

```

```

//TEST//

```

```

//////

```

```

case -6:
TEST:
NumChoose=5;
ButtonMidTakeOver(NumPress,NumChoose);
m_hide( );
m_hide( );
ModeTemp=GET_VOC;
Make_Screen(7,0);
say_command(voormsg);
break;
case ESC:
case -7:
EXIT:
NumChoose=6;
ButtonMidTakeOver(NumPress,NumChoose);
c=ESC;
if(ChkYN("ท่านจะออกไปเมนู <Y/N> ?",No)!=Yes)
c!=ESC;
default: break;
}
}while(c!=ESC);

```

```

m_hide( );
WindowHide(maxx/18,maxy/5,maxx-maxx/18,maxy-maxy/6,bkcolor,bkcolor,0);
ModeTemp=GET_MAIN;
Make_Screen(1, 0);
say_command("เมนูเมนู");
m_show( );
ToView(1);
}

```

```

void VocBarAtt(int i) { //highlight choosing option
int j,k;

k=(NumPress==NumChoose && i<=6)?1:2;
ToView(1);
for(j=0; j<k; j++) {
if(j>0 || k==1) {
DelayChkKey(0.12);
KeyMouRelease( );
}
switch(i) {
case 0:
case 1:
case 6:
BarAttrib(1,i*IconHi,box_width,(i+1)*IconHi,15,XOR_PUT);
break;

case 2:
case 3:
BarAttrib(i<3?maxx/2-43:maxx/2+52, maxy/3+41,
i<3?maxx/2+51:maxx/2+147,maxy/3+116,15,XOR_PUT);
break;

```

```

case 4:
case 5:
BarAttrib(i<5?maxx/2-43:maxx/2+52, maxy/3+115,
          i<5?maxx/2+51:maxx/2+147,maxy/3+190,15,XOR_PUT);
break;
case 7:
    ToView(2);
    if(linsert_mode) { //Insert button ==> WR_PH-mode
        insert_mode ^=TRUE;
        insert_status();
    }
break;
case 8: //No
case 9: //Cancel
case 10: //Yes
    BarAttrib( 236+((i-8)*87)+(i==10?1:0),219,317+((i-8)*88),
              247,15,XOR_PUT);
    break;
case 11:
case 12:
case 13:
case 14:
    BarAttrib(314,311-((i-11)*27),391,311-((i-11)*27)+29,15,XOR_PUT);
    break;
} //SWITCH
}
if(i>=11 && i<=14)
    ToView(4);
NumPress=NumChoose=-1;
}
void Sh_NextPage(int mode) {
int i, Rpage[ ]={544,122,544, 94,562, 72,562,109,544,122},

```



```

Lpage[ ]={544,122,544, 94,530, 72,530,109,544,122};
ToView(1);
for(i=0; i<2; i++) {
    m_hide( );
    sv_image1(529,71,563,123);
    setwritemode(COPY_PUT);
    setlinestyle(0,0,1);
    setfillstyle(SOLID_FILL,WHITE);
    setcolor(DARKGRAY);
    if(i==(mode===-91)?0:1)
        fillpoly(sizeof(Rpage)/(2*sizeof(int)),Rpage);
    else
        fillpoly(sizeof(Lpage)/(2*sizeof(int)),Lpage);
    m_show( );
    Delay(0.05);
    m_hide( );
    sh_image1(529,71);
    m_show( );
Delay(0.05);
}
ToView(2);
}

void ClearMemData(void) {
    if(hb!=NULL)
        free(hb);
    if(rb!=NULL)
        free(rb);
    if(sc!=NULL)
        free(sc);
}

```

```

/* ----- ph_schem.c ----- */

/* Read a file from the standard input.
 * Write one of three schema files to the standard output
 * depending on the runtime switch.
 * If -1 is on the command line,
 *
 *     the #define statements are built.
 * If -2 is on the command line,
 *
 *     the file of ascii strings for
 *     file and data element names is built.
 * If -3 is on the command line,
 * the data base schema array source file is built.
 */

/*****
/* RUN WITH COMMAND LINE ONLY */
*****/

#include <stdio.h>
#define SCHEMA
#include <cdatadic.i>

#define MXCAT 3      /* maximum elements concatenated per index */
#define NAMLEN 31

struct dict {
    // data element dictionary
    char dename [NAMLEN+1];    // name
    char dtype;                // type
    int dlen;                  // length
    char *demaask;             // display mask
} dc [MXELE];

int dectr = 0;                // data elements in dictionary
int fctr = 0;                  // files in data base
char filename [MXFILS] [NAMLEN+1]; // file name strings

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int filele [MXFILS] [MXELE];           // elements in files
int ndxle  [MXFILS] [MXINDEX] [MXCAT]; // indices

char word[NAMLEN+1];

char *malloc( );

int lnctr = 0; /* input stream line counter */
char ln [400]; /*160*/

/* ----- error messages ----- */
char *ers[ ] = {
    "invalid name",
    "invalid length",
    "comma missing",
    "invalid data type",
    "quote missing",
    "#schoma missing",
    "#<command> missing",
    "unexpected end of file",
    "duplicate file name",
    "unknown data element",
    "too many data elements",
    "out of memory",
    "unknown file name",
    "too many indices in file",
    "too many elements in index",
    "duplicate data element",
    "too many files",
    "invalid command line switch"
};

void de_dict( ), files( ), keys( ), schout( ), defout( ),
    lcase( ), strout( ), error( ), get_line( ), skip_white( ),
    name_val( ), numb_val( ), expect_comma( ), depart( );

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char *get_word( );

#define iswhite(c) ((c)=='\l(c)=='\t')
#define REMARK ';'

/* ----- main program ----- */
void main(argc, argv)

int argc;

char *argv[ ];

{

    get_line( );
    if (strcmp(ln, "#schema ", 8))
        error(6);
    else
    {
        get_word(ln + 8);
        name_val( );
        printf("\n");
        printf("----- %s ----- *\n", word);
    }
    get_line( );
    while (strcmp(ln, "#end schema", 11)) {
        if (strcmp(ln, "#dictionary", 11) == 0)
            de_dict( );
        else if (strcmp(ln, "#file ", 6) == 0)
            files( );
        else if (strcmp(ln, "#key ", 5) == 0)
            keys( );
        else
            error(7);
        get_line( );
    }
    if (argc > 1) {
        if (strcmp(argv[1], "-1") == 0)
            defout( );

```

```

else if (strcmp(argv[1], "-2") == 0)
    strout( );
else if (strcmp(argv[1], "-3") == 0)
    schout( );
else
    error(18);
}
else
    error(18);
depart(0);
}

/* ----- build the data element dictionary ----- */
static void do_dict( )
{
    char *cp, *cpl;
    int el;
    while (TRUE) {
        get_line( );
        if (strcmp(ln, "#end dictionary", 15) == 0)
            break;
        if (dectr == MXELE) {
            error(11);
            continue;
        }
        cp = get_word(ln);
        name_val( );
        for (el = 0; el < dectr; el++)
            if (strcmp(word, dc[el].dename) == 0) {
                error(16);
                continue;
            }
        strcpy(dc[dectr].dename, word);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

expect_comma(&cp);
skip_white(&cp);
switch (*cp)    {
    case 'A':
    case 'B':
    case 'E':
    case 'Z':
    case 'C':
    case 'N':
    case 'D': break;
    default : error(4);
                continue;
}
dc[dctr].dctype = *cp++;
expect_comma(&cp);
cp = get_word(cp);
numb_val( );
dc[dctr].dolen = atoi(word);
expect_comma(&cp);
skip_white(&cp);
if (*cp != '') {
    error(5);
    continue;
}
}
cp1 = cp + 1;
while (*cp1 != "" && *cp1 && *cp1 != '\n')
    cp1++;
if (*cp1++ != "") {
    error(5);
    continue;
}
*cp1 = '\0';
if ((dc[dctr].demask = malloc((cp1-cp)+1)) == 0) {

```

```

        error(12);
        depart(1);
    }
    strcpy(dc[dctr].demask, cp);
    dctr++;
}
}

/* ----- build the file definitions ----- */
static void files( )
{
    int i, el = 0;
    if (fctr == MXFILS)
        error(17);
    get_word(ln + 6); /* get the file name */
    name_val( ); /* validate it */
    for (i = 0; i < fctr; i++) /* already assigned? */
        if (strcmp(word, filename[i]) == 0)
            error(9);
    strcpy(filename[fctr], word);
    /* ----- process the file's data elements ----- */
    while (TRUE) {
        get_line( );
        if (strcmp(ln, "#end file", 9) == 0)
            break;
        if (el == MXELE) {
            error(11);
            continue;
        }
        get_word(ln); /* get a data element */
        for (i = 0; i < dctr; i++) /* in dictionary? */
            if (strcmp(word, dc[i].dename) == 0)
                break;
    }
}

```

```

        if (i == dectr)
            error(10);
        else if (fctr < MXFILS)
            /* post to file */
            fileole [fctr] [el++] = i + 1;
    }
    if (fctr < MXFILS)
        fctr++;
}

/* ----- build the index descriptions ----- */
static void keys( )
{
    char *cp;
    int f, el, x, cat = 0;
    cp = get_word(ln + 5);          /* get the file name */
    for (f = 0; f < fctr; f++) /* in the schema? */
        if (strcmp(word, filename[f]) == 0)
            break;
    if (f == fctr) {
        error(13);
        return;
    }
    for (x = 0; x < MXINDEX; x++)
        if (*ndxele [f] [x] == 0)
            break;
    if (x == MXINDEX) {
        error(14);
        return;
    }
    while (cat < MXCAT) {
        cp = get_word(cp);          /* get index name */
        for (el = 0; el < dectr; el++) /* in dictionary? */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

        if (strcmp(word, dc[el].dename) == 0)
            break;
    .if (el == dectr) {
        error(10);
        break;
    }
    ndxelo [f] [x] [cat++] = el + 1; /* post element */
    skip_white(&cp);
    if (*cp++ != ',') /* concatenated index? */
        break;
    if (cat == MXCAT) {
        error(15);
        break;
    }
}
}

/* ----- write the schema source language ----- */
static void schout( )
{
    int f, el, x, xl, cat, fel;
    char name [NAMLEN+1];

    /* ----- data element lengths ----- */
    printf("\n\nint ellen [ ] = {");
    for (el = 0; el < dectr; el++) {
        if ((el % 25) == 0)
            printf("\n\t");
        printf("%d", dc[el].dolen);
    }
    printf("\n};\n");

    /* ----- write the file contents arrays ----- */
    for (f = 0; f < fctr; f++) {

```

```

lcase(name, filename [f]);
printf("\n\nint f_%s [ ] = {", name);
el = 0;
while ((fel = fileole[f] [el++]) != 0)
    printf("\n\n%s", dc[fel-1].dename);
printf("\n\n");
}
/* ----- write the file list pointer array ----- */
printf("\n\nint *file_ole [ ] = {");
for (f = 0; f < fctr; f++) {
    lcase(name, filename [f]);
    printf("\n\nf_%s", name);
}
printf("\n\n");
/* ----- write the index arrays ----- */
for (f = 0; f < fctr; f++) {
    lcase(name, filename [f]);
    for (x = 0; x < MXINDEX; x++) {
        if (*ndxole [f] [x] == 0)
            break;
        printf("\n\nint x%d_%s [ ] = {", x + 1, name);
        for (cat = 0; cat < MXCAT; cat++)
            if (ndxole [f] [x] [cat])
                printf("\n\n%s",
                    dc[ndxole [f] [x] [cat] - 1].dename);
        printf("\n\n");
    }
    printf("\n\nint *x_%s [ ] = {", name);
    for (x1 = 0; x1 < x; x1++)
        printf("\n\nx%d_%s", x1 + 1, name);
    printf("\n\n");
}
printf("\n\nint **index_ole [ ] = {");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (f = 0; f < fctr; f++) {
    lcase(name, filename [f]);
    printf("\n\tx_%s,", name);
}
printf("\n\0\n};\n");
}

/* -- write the schema #defines and struct definitions -- */
static void defout( )
{
    int f, el, fel;
    char name [NAMLEN+1];

    /* ----- data element defines ----- */
    for (el = 0; el < dectr; el++)
        printf("\n#define %s %d", dc[el].dename, el + 1);
    putchar('\n');
    /* ----- write the file #define statements ----- */
    for (f = 0; f < fctr; f++)
        printf("\n#define %s %d", filename [f], f);
    putchar('\n');
    /* ----- write the record structures ----- */
    for (f = 0; f < fctr; f++) {
        lcase(name, filename [f]);
        printf("\nstruct %s {", name);
        el = 0;
        while ((fel = fileolo[f] [el++]) != 0) {
            lcase(name, dc[fel-1].dename);
            printf("\n\tchar %s [%d];",
                name, dc[fel-1].delen + 1);
        }
        printf("\n};\n");
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

```

```
/* ----- write the file and data element ascii strings ----- */

```

```
static void strout( )

```

```
{

```

```
    int el, f;

```

```
    /* ----- data element ascii names ----- */

```

```
    printf("\nchar *denames [ ] = {");

```

```
    for (el = 0; el < dectr; el++)

```

```
        printf("\n\\\"%s\\\",", dc[el].dename);

```

```
    printf("\n\\0\n");

```

```
    /* ----- data element types ----- */

```

```
    printf("\nchar eltype [ ] = \";

```

```
    for (el = 0; el < dectr; el++)

```

```
        putchar(dc[el].detype);

```

```
    printf("\n");

```

```
    /* ----- data element display masks ----- */

```

```
    printf("\nchar *elmask [ ] = {");

```

```
    for (el = 0; el < dectr; el++)

```

```
        printf((el < dectr-1 ?

```

```
            "\n\\\"%s,":

```

```
            "\n\\\"%s"),dc[el].demask);

```

```
    printf("\n");

```

```
    /* ----- write the ascii file name strings ----- */

```

```
    printf("\nchar *dbfiles [ ] = {");

```

```
    for (f = 0; f < fctr; f++)

```

```
        printf("\n\\\"%s\\\",", filename [f]);

```

```
    printf("\n\\0\n");

```

```
}

```

```
/* ----- convert a name to lower case ----- */

```

```
static void lcase(s1, s2)

```

```
char *s1, *s2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    while (*s2)    {
        *s1 = tolower(*s2);
        s1++;
        s2++;
    }
    *s1 = '\0';
}

```

```

/* ----- errors ----- */
static void error(n)
int n;
{
    static int erct = 0;
    static int erlin = 0;

    if (erlin != lnctr) {
        erlin = lnctr;
        fprintf(stderr, "\nLine %d: %s", lnctr, ln);
    }
    fprintf(stderr, "\n Error %d: %s", n, ers[n-1]);
    if (erct++ == 5) {
        erct = 0;
        fprintf(stderr, "\nContinue? (y/n) ... ");
        if (tolower(getc(stderr)) != 'y')
            depart(1);
    }
}

/* --- get a line of data from the schema input stream --- */
static void get_line( )
{
    *ln = '\0';

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while (*ln == '\0' || *ln == REMARK || *ln == '\n') {
    if (fgets(ln,120,stdin) == 0) {
        error(8);
        depart(1);
    }
    lnctr++;
}
}

```

```

/* ----- skip over white space ----- */

```

```

static void skip_white(s)

```

```

char **s;

```

```

{

```

```

    while (iswhite(**s))

```

```

        (*s)++;

```

```

}

```

```

/* ----- get a word from a line of input ----- */

```

```

static char *got_word(cp)

```

```

char *cp;

```

```

{

```

```

    int wl = 0, fst = 0;

```

```

    skip_white(&cp);

```

```

    while (*cp && *cp != '\n' &&

```

```

        *cp != ',' &&

```

```

            iswhite(*cp) == 0){

```

```

        if (wl == NAMLEN && fst == 0) {

```

```

            error(1);

```

```

            fst++;

```

```

        }

```

```

    else

```

```

        word [wl++] = *cp++;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    word [wl] = '\0';
    return cp;
}

/* ----- validate a name ----- */
static void name_val( )
{
    char *s = word;
    if (isalpha(*s)) {
        while (isalpha(*s) || isdigit(*s) || *s == '_') {
            *s = toupper(*s);
            s++;
        }
        if (*s == '\0')
            return;
    }
    error(1);
}

/* ----- validate a number ----- */
static void numb_val( )
{
    char *s = word;
    do {
        if (isdigit(*s++) == 0) {
            error(2);
            break;
        }
    } while (*s);
}

/* ----- expect a comma next ----- */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

static void expect_comma(cp)
char **cp;
{
    skip_white(cp);
    if ((*cp)++ != ',')
        error(3);
}

/* ----- terminate schema program ----- */
static void depart(n)
int n;
{
    int el;
    for (el = 0; el < dectr; el++)
        free(dc[el].demask);
    exit(n);
}

#if COMPILER == WIZARD
/* --- strcmp function to replace non-standard version --- */
int strcmp(s1, s2, n)
char *s1, *s2;
{
    int i;
    while (n--)
        if (i = (*s1++ - *s2++))
            return i;
    return 0;
}
#endif

```



```

/*-----*/
/*          ph_util.c          */
/*-----*/

int box_width;

#include <graphics.h>
#include <stdio.h>
#include <ctype.h>
#include <process.h>
#include <ph_thai.i>
#include <cdataadic.i>
#include <ph_pvd.i1>
#include <ph_pvd.i2>
#include <ph_pvd.i3>
#include <databas2.i>
#include <ph_datfi.i>
#include <btrec.i>

//----- indexing -----
#define BELL 7
extern RPTR curr_a []; /* current record file address */

void main (int argc, char *argv[]) {
    int f = 1, i;
    static int fs [MXFILES+1];
    char *malloc(), *bf, *path;
    char *errmsg="\nCannot run this file.";
    if (argc < 2)    {
        printf(errmsg);
        exit(1);
    }

    if(stremp("all", argv [1]) == 0) { /* index all */
        if (stremp("all", argv [1]) == 0) /* index all */
            for (f = 0; dbfiles [f]; f++) /* put files in list */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        fs [f] = f;
    else if ((*fs = filename(argv[1])) == ERROR)
        exit(1);
    fs [f] = (-1);    /* terminator (file,file,...,-1) */
    path = argc > 2 ? argv[2] : "";
    /* delete and rebuild indexes
        in the data base files being indexed */
    for (i = 0; (f = fs [i]) != (-1); i++)
        build_index(path, f);
    /* Open the data base files. */
    db_open(path, fs);
    for (i = 0; (f = fs [i]) != (-1); i++) {
        bf = malloc(strlen(f));
        while (seqred(f, bf) != ERROR)
            add_indexes(f, bf, curr_a [f]);
        free(bf);
    }
    db_cls();
}
else {
    printf(errmsg);
    exit(1);
}
}
//DUMMY FUNCTION//////////////////
int IsMidChar(char *msg) {
    return (int)msg;
}

void cursor(int x, int y, int mode) {
    x+=0;
    y+=0;
    mode+=0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
/*          PH_VRS1.C          */
/* FUNCTION : Check sum of executables */
/*          and save it in text file   */
/*-----*/

#include <dos.h>
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <string.h>

#pragma pack(1)
#define CRC32_POLYNOMIAL 0xEDB88320L

unsigned long CRCTable[256];
extern unsigned _stklen = 16384; /* 15K stack size */

void BuildCRCTable(void) {
register int i, j;
unsigned long crc;

for (i=0; i<=255; i++) {
    crc=i;
    for(j=8; j>0; --j) {
        if(crc&1) crc=(crc>>1) ^ CRC32_POLYNOMIAL;
        else crc >>= 1;
    }
    CRCTable[i] = crc;
}
}

```

```

void FatalError(char *fmt, ...) {
va_list argptr;
    va_start(argptr, fmt);
    printf("Fatal error: ");
    vprintf(fmt, argptr);
    va_end(argptr);
    exit(-1);
}

```

```

unsigned long CalculateBufferCRC(unsigned count,
                                unsigned long crc, void *buffer) {
unsigned char *p;
unsigned long temp1, temp2;

p = (unsigned char *) buffer;
while (--count != 0) {
    temp1 = (crc >> 8) & 0x00FFFFFFL;
    temp2 = CRCTable[((int) crc ^ *p++) & 0xFF];
    crc = temp1 ^ temp2;
}
return crc;
}

```

```

#define BUFFSIZE 16384U
unsigned long CalculateFileCRC(FILE *file) {
unsigned long crc;
unsigned i, count;
static char narikar[] = "\n/";
static unsigned char buffer[BUFFSIZE];

crc = 0xFFFFFFFFL;
for(i=0;;)
{
    count = fread(buffer, 1, BUFFSIZE, file);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    putchar('\b');
    putchar(narikal[i]);
    if(++i > 4) i=1;
    if(count==0)
        break;
    crc = CalculateBufferCRC(count, crc, buffer);
}
putchar('\b');
putchar(' ');
return(crc^=0xFFFFFFFF);
}

int CheckFiles(char *CRCfilename) {
FILE *crcfile, *testfile;
unsigned long crc, log_crc;
char log_name[FILENAME_MAX];
int result,error=0;

if((crcfile = fopen(CRCfilename, "r")) == NULL)
    FatalError("Couldn't open the log file: %s\n",
    CRCfilename);
for(;;) {
    result = fscanf(crcfile, "%lx %s", &log_crc, log_name);
    if(result<2)
        break;
    testfile = fopen(log_name, "rb");
    if(testfile != NULL) {
        crc = CalculateFileCRC(testfile);
        fclose(testfile);
        if(crc!=log_crc) {
            error=1;
            printf("\aFile '%-40s' has been damage..."log_name);
            if(getch() ==0)

```

```

        (void) getch();
        putchar('\n');
    }
    else
        printf("OK\n");
}
else
    printf("Could not open file %s\n", log_name);
} //for(;;)
return error;
}

void ProcessAllFiles(char *path, FILE *crc_file) {
int done;
struct find_t fileinfo;
char fullname[FILENAME_MAX];
unsigned long crc;
FILE *file;

    strcat(path, "\\");
    strcpy(fullname, path);
    strcat(fullname, "*.*");
    done = _dos_findfirst(fullname,
        _A_SUBDIR | _A_HIDDEN | _A_SYSTEM, &fileinfo);
    while(done == 0) {
        strcpy(fullname, path);
        if(strcmp(fileinfo.name, ".") && strcmp(fileinfo.name, "..")) {
            strcat(fullname, fileinfo.name);
            if(fileinfo.attrib & _A_SUBDIR)
                ProcessAllFiles(fullname, crc_file);
            else {
                file = fopen(fullname, "rb");
                if(file != NULL) {

```

```

    fprintf("%-40s  ", fullname);
    crc = CalculateFileCRC(file);
    fprintf("%08lx\n", crc);
    fprintf(crc_file, "%08lx %s\n", crc, fullname);
    fclose(file);
}
else
    printf("Could not open %s\n", fullname);
}
}
done = _dos_findnext(&fileinfo);
}
}

void BuildCRCFile(char *input_dir, char *crc_filename) {
char path[FILENAME_MAX];
FILE *crc_file;

strcpy(path, input_dir);
if(path[strlen(path)-1] == '\\')
    path[strlen(path)-1] = '\0';
if((crc_file = fopen(crc_filename, "w")) == NULL)
    FatalError("Can't open CRC log file: %s\n", crc_filename);
ProcessAllFiles(path, crc_file);
}

int main( /* int argc, char *argv[] */ ) {
int argc=4;
char *argv1[4];
    argv1[1]="-b";
    argv1[2]="c:\projec\";
    argv1[3]="chk_vr.crc";
BuildCRCTable();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(argc==2) {
    if(CheckFiles(argv[1])!=1); // no error
}
else if(argc==4 && strcmp(argv[1], "-b")!=0) {
    BuildCRCFile(argv[2], argv[3]);
}.
else {
    .puts("\nUsage: DICPHASE [-b input-dir] CRC-file\n");
    .puts("Using the -b option checks all files under the input -dir");
    puts("and appends their data to the CRC-file, Otherwise, the");
    puts("programme checks the CRC data of all of the files in the");
    puts("CRC-file and prints the results\n");
    return 1;
}
return 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

/*-----*/
/*          PH_VRS2.C          */
/* COMPILE TO: ph_vrs2.com     */
/* FUNCTION : Check file-damage for Project-files */
/*          and initialize the programme          */
/*-----*/

```

```

#include <dos.h>
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <string.h>
#include <process.h>
#include <dir.h>
#include <mem.h>
#include <fcntl.h>
#include <io.h>

#pragma pack(1)
#define CRC32_POLYNOMIAL 0xEDB88320L

unsigned long CRCTable[256];
extern unsigned _stklen = 16384; /* 15K stack size */

void BuildCRCTable(void) {
register int i, j;
unsigned long crc;

for (i=0; i<=255; i++) {
    crc=i;
    for(j=8; j>0; --j) {
        if(crc&1) crc=(crc>>1) ^ CRC32_POLYNOMIAL;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    else crc >>= 1;
}
CRCTable[i] = crc;
}
}

void FatalError(char *fmt, ...) {
va_list argptr;

va_start(argptr, fmt);
printf("Fatal error: ");
vprintf(fmt, argptr);
va_end(argptr);
exit(-1);
}

unsigned long CalculateBufferCRC(unsigned count,
    unsigned long crc, void *buffer) {
unsigned char *p;
unsigned long temp1, temp2;

p = (unsigned char *) buffer;
while (--count != 0) {
    temp1 = (crc >> 8) & 0x00FFFFFFL;
    temp2 = CRCTable[((int) crc ^ *p++) & 0xFF];
    crc = temp1 ^ temp2;
}
return crc;
}

#define BUFFSIZE 16384U
unsigned long CalculateFileCRC(FILE *file) {
unsigned long crc;

```

```

unsigned i, count;
static char narikar[ ] = "\\\\";
static unsigned char buffer[BUFSIZE];

crc = 0xFFFFFFFFL;
for(i=0;;)
{
    count = fread(buffer, 1, BUFSIZE, file);
    putchar('\b');
    putchar(narikar[i]);
    if(++i > 4) i=1;
    if(count==0)
        break;
    crc = CalculateBufferCRC(count, crc, buffer);
}
putchar('\b');
putchar(' ');
return(crc^=0xFFFFFFFFL);
}

int CheckFiles(char *CRCfilename) {
FILE *crcfile, *logfile;
unsigned long crc, log_crc;
char log_name[FILENAME_MAX];
int result,error=0;

if((crcfile = fopen(CRCfilename, "r")) == NULL)
    FatalError("Couldn't open the log file: %s\n",
    CRCfilename);

for(;;) {
    result = fscanf(crcfile, "%lx %s", &log_crc, log_name);
    if(result<2)

```

```

    break;
testfile = fopen(log_name, "rb");
if(testfile != NULL)    {
    crc = CalculateFileCRC(testfile);
    fclose(testfile);
    if(crc!=log_crc)    {
        error=-1;
        printf("\nFile '%-40s' has been damage...\n",log_name);
        putchar('\r');
    }
}
else
    printf("Could not open file %s\n", log_name);
}
return error;
}

unsigned long han_report(int handle) {
    return (unsigned long) filelength(handle);
}

int ChkFileSize(void) {
    struct fblk file;
    int i,handle,error=0;
    unsigned long fsize[ ]= {
        211232,
        8958,
        29920
    };
    char *filename[ ] = {
        "phases.exe",

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

"dicmouse.com",
"ph_vrs2.pwa"
};
for(i=0; i<3; i++) {
    if(findfirst(filename[i],&file,0xff) == 0) {
        handle = open(filename[i], O_RDONLY);
        if(fsize[i] != han_report(handle)) { //ERROR
            printf("\nERROR : The size of %s. . .",filename[i]);
            close(handle);
            error=-1;
            break;
        }
        close(handle);
    }
    else {
        printf("\nERROR : %s not found. . .",filename[i]);
        error=-1;
        break;
    }
}
if(error== -1)
    printf("a");
return error;
}

void main(int argc, char *argv[ ] ) {
int error=0; //no_error
char c, *arg1="ed-eng15";

if(argc!=2 || strcmp(arg1,argv[1])!=0) {
    printf("This file run under DB_PHASE.COM\n");
    exit(-1);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(ChkFileSize( )== -1) //error
    error=-1;
if(error!= -1) {
    BuildCRCTable( );
    if(CheckFiles("chk_vr.crc")== -1)
        error=-1;
}

if(error== -1) { // error
    printf("\n\nMaybe the virus-occupation. . .");
    exit(-1);
}
exit(0); //no-error
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* ----- btree.c ----- */
#define MXTREES 20
#define ADR sizeof(RPTR)
#define KLEN bheader[trx].keylength
#define ENTLN (KLEN+ADR)

/* ----- the btree node structure ----- */
typedef struct treenode {
    int nonleaf;    // 0 if leaf, 1 if non-leaf
    RPTR prntnode; // parent node
    RPTR lfsib;    // left sibling node
    RPTR rtsib;    // right sibling node
    int keyct;     // number of keys
    RPTR key0;     // node # of keys < 1st key this node
    char keyspace [NODE - ((sizeof(int) * 2) + (ADR * 4))];
    char spil [MXKEYLEN]; // for insertion excess
} BTREE;

/* ---- the structure of the btree header node ----- */
typedef struct treehdr {
    RPTR rootnode; /* root node number */
    int keylength;
    int m;         /* max keys/node */
    RPTR rised_node; /* next released node */
    RPTR endnode;   /* next unassigned node */
    int locked;    /* if btree is locked */
    RPTR leftmost; /* left-most node */
    RPTR rightmost; /* right-most node */
} HEADER;

HEADER bheader [MXTREES];
BTREE tnode;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#if COMPILER == TURBOC
handle [MXTREES+=0]; /* handle of each index in use */
#endif

RPTR currnode [MXTREES]; /* node number of current key */
int currkno [MXTREES]; /* key number of current key */
int trx; /* current tree */

char *malloc(), *childptr();
void redist(), adopt(), implode(),
    read_node(), write_node(), bseek();
RPTR firstkey(), lastkey(), scannext(), scanprev(),
    leaflevel(), nextnode(), fileaddr();

/* ----- initiate b-tree processing ----- */
int btree_init(ndx_name)
char *ndx_name;
{
#if COMPILER == MICROSOFT
    extern int _iomode;
    _iomode = 0x8000;
#endif
#if COMPILER == LATTICE
    extern int _iomode;
    _iomode = 0x8000;
#endif
    for (trx = 0; trx < MXTREES; trx++)
        if (handle [trx] == 0)
            break;
    if (trx == MXTREES)
        return ERROR;
    if ((handle [trx] = open(ndx_name, OPENMODE)) == ERROR)
        return ERROR;
    lseek(handle [trx], 0L, 0);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

read(handle [trx],(char *) &bheader[trx],sizeof(HEADER));
if (bheader[trx].locked) {
    close(handle [trx]);
    handle [trx] = 0;
    return ERROR;
}
bheader[trx].locked = TRUE;
lseek(handle [trx], 0L, 0);
write(handle [trx],(char *) &bheader[trx],sizeof(HEADER));
currnode [trx] = 0;
currkno [trx] = 0;
return trx;
}
/* ----- terminate b tree processing ----- */
int btree_close(tree)
int tree;
{
    if (tree >= MXTREES || handle [tree] == 0)
        return ERROR;
    bheader[tree].locked = FALSE;
    lseek(handle[tree], 0L, 0);
    write(handle[tree],(char *)&bheader[tree],sizeof(HEADER));
    close(handle[tree]);
    handle[tree] = 0;
    return OK;
}

/* -----Build a new b-tree ----- */
void build_b(name, len)
char,*name;
int len;
{
    HEADER *bhd;

```

```

    int fd;
#ifdef COMPILER == MICROSOFT
    extern int _iomode;
    _iomode = 0x8000;
#endif
#ifdef COMPILER == LATTICE
    extern int _iomode;
    _iomode = 0x8000;
#endif

    if ((bhdp = (HEADER *)malloc(NODE))==(HEADER *)NULL) {
        errno = D_OM;
        dberror();
    }
    set_mem(bhdp, NODE, '\0');
    bhdp->koylength = len;
    bhdp->m = ((NODE-((sizeof(int)*2)+(ADR*4)))/(len+ADR));
    bhdp->endnode = 1;
    unlink(name);
    fd = creat(name, CMODE);
    close(fd);
    fd = open(name, OPENMODE);
    write(fd, (char *) bhdp, NODE);
    close(fd);
    free(bhdp);
}

/* ----- Locate key in the b-tree ----- */
RPTR locate(tree, k)
int tree;
char *k;
{
    int i, fnd = FALSE;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RPTR t, ad;

char *a;

trx = tree;

t = bheader[trx].rootnode;

if (t) {
    read_node(t, &trnode);
    fnd = btreescan(&t, k, &a);
    ad = leaflevel(&t, &a, &i);
    if (i == trnode.keyct + 1) {
        i = 0;
        t = trnode.risib;
    }
    currnode [trx] = t;
    currkno [trx] = i;
}
return fnd ? ad : (RPTR) 0;
}

/* ----- Search tree ----- */
static int btreescan(t, k, a)
RPTR *t;
char *k, **a;
{
    int nl;
    do {
        if (nodescan(k, a) {
            while (compare_keys(*a, k) == FALSE)
                if (scanprev(t, a) == 0)
                    break;
            if (compare_keys(*a, k)
                scannext(t, a);
            return TRUE;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        nl = trnode.nonleaf;
        if (nl) {
            *t = *((RPTR *) (*a - ADR));
            read_node(*t, &trnode);
        }
    } while (nl);
    return FALSE;
}

```

```

/* ----- Search node ----- */

```

```

static int nodescan(keyvalue, nodeadr)
char *keyvalue, **nodeadr;
{
    int i;
    int result;
    *nodeadr = trnode.keyspace;
    for (i = 0; i < trnode.keyset; i++) {
        result = compare_keys(keyvalue, *nodeadr);
        if (result == FALSE) return TRUE;
        if (result < 0) return FALSE;
        *nodeadr += ENTLN;
    }
    return FALSE;
}

```

```

/* ----- Compare keys ----- */

```

```

static int compare_keys(a, b)
char *a, *b;
{
    int len = KLEN, cm;

    while (len--)
        if ((cm = (int) *a++ - (int) *b++) != 0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;

    return cn;
}

/* ----- Compute current file address ----- */
static RPTR fileaddr(t, a)
RPTR t;
char *a;
{
    RPTR cn, ti;
    int i;
    ti = t;
    cn = leaflevel(&ti, &a, &i);
    read_node(t, &tnode);
    return cn;
}

/* ----- Navigate down to leaf level ----- */
static RPTR leaflevel(t, a, p)
RPTR *t;
char **a;
int *p;
{
    if (tnode.nonleaf == FALSE)    { /* already at a leaf? */
        *p = (*a - tnode.keyspace) / ENTLN + 1;
        return *((RPTR *) (*a + KLEN));
    }
    *p = 0;
    *t = *((RPTR *) (*a + KLEN));
    read_node(*t, &tnode);
    *a = tnode.keyspace;
    while (tnode.nonleaf)    {
        *t = tnode.key0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        read_node(*t, &trnode);
    }
    return trnode.key0;
}

```

```

/* ----- Delete a key ----- */

```

```

int deletekey(tree, x, ad)
int tree;
char *x;
RPTR ad;
{
    BTREE *qp, *yp;
    int rt_len, comb;
    RPTR p, adr, q, *b, y, z;
    char *a;
    trx = tree;
    if (trx >= MXTREES || handle [trx] == 0)
        return ERROR;
    p = bheader[trx].rootnode;
    if (p == 0)
        return OK;
    read_node(p, &trnode);
    if (btreescan(&p, x, &a) == FALSE)
        return OK;
    adr = fileaddr(p, a);
    while (adr != ad) {
        adr = scannext(&p, &a);
        if (compare_keys(a, x))
            return OK;
    }
    if (trnode.nonleaf) {
        b = (RPTR *) (a + KLEN);
        q = *b;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if ((qp=(BTREE *) malloc(NODE))==(BTREE *) NULL) {
    errno = D_OM;
    dberror();
}
read_node(q, qp);
while (qp->nonleaf) {
    q = qp->key0;
    read_node(q, qp);
}
/* Move the left-most key from the leaf
to where the deleted key is */
mov_mem(qp->keyspace, a, KLEN);
write_node(p, &trnode);
p = q;
mov_mem(qp, &trnode, sizeof trnode);
a = trnode.keyspace;
b = (RPTR *) (a + KLEN);
trnode.key0 = *b;
free(qp);
}
currnode [trx] = p;
currkno [trx] = (a - trnode.keyspace) / ENTLN;
rt_len = (trnode.keyspace + (bheader[trx].m * ENTLN)) - a;
mov_mem(a + ENTLN, a, rt_len);
set_mem(a + rt_len, ENTLN, '\0');
trnode.keyct--;
if (currkno [trx] > trnode.keyct) {
    if (trnode.rtsib) {
        currnode [trx] = trnode.rtsib;
        currkno [trx] = 0;
    }
    else
        currkno [trx]--;
}

```

```

}
while (tnode.keyct <= bheader[trx].m / 2 &&
      p != bheader[trx].rootnode) {
    comb = FALSE;
    z = tnode.prtnode;
    if ((yp=(BTREE *) malloc(NODE))==(BTREE *) NULL) {
        orno = D_OM;
        dberror();
    }
    if (tnode.rtsib) {
        y = tnode.rtsib;
        read_node(y, yp);
        if (yp->keyct + tnode.keyct <
            bheader[trx].m && yp->prntnode == z) {
            comb = TRUE;
            implode(&tnode, yp);
        }
    }
    if (comb == FALSE && tnode.lfsib) {
        y = tnode.lfsib;
        read_node(y, yp);
        if (yp->prntnode == z) {
            if (yp->keyct + tnode.keyct <
                bheader[trx].m) {
                comb = TRUE;
                implode(yp, &tnode);
            }
        }
        else {
            redist(yp, &tnode);
            write_node(p, &tnode);
            write_node(y, yp);
            free(yp);
            return OK;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

    }
}
}
if (comb == FALSE) {
    y = tnode.rtaib;
    read_node(y, yp);
    redist(&tnode, yp);
    write_node(y, yp);
    write_node(p, &tnode);
    free(yp);
    return OK;
}
free(yp);
p = z;
read_node(p, &tnode);
}
if (tnode.keyct == 0) {
    bheader[trx].rootnode = tnode.key0;
    tnode.nonleaf = FALSE;
    tnode.key0 = 0;
    tnode.prtnode = bheader[trx].rised_node;
    bheader[trx].rised_node = p;
}
if (bheader[trx].rootnode == 0)
    bheader[trx].rightmost = bheader[trx].leftmost = 0;
write_node(p, &tnode);
return OK;
}

/* ----- Combine two sibling nodes. ----- */
static void implode(left, right)
BTREE *left, *right;
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RPTR lf, rt, p;
int rt_len, lf_len;
char *a;
RPTR *b;
BTREE *par;
RPTR c;
char *j;
lf = right->lfsib;
rt = left->rtsib;
p = left->prntnode;
if ((par = (BTREE *) malloc(NODE)) == (BTREE *) NULL) {
    errno = D_OM;
    dberror();
}
j = childptr(lf, p, par);
/* --- move key from parent to end of left sibling --- */
lf_len = left->keyct * ENTLN;
a = left->keyspace + lf_len;
mov_mem(j, a, KLEN);
set_mem(j, ENTLN, '\0');
/* --- move keys from right sibling to left --- */
b = (RPTR *) (a + KLEN);
*b = right->key0;
rt_len = right->keyct * ENTLN;
a = (char *) (b + 1);
mov_mem(right->keyspace, a, rt_len);
/* --- point lower nodes to their new parent --- */
if (left->nonleaf)
    adopt(b, right->keyct + 1, lf);
/* --- if global key pointers -> to the right sibling,
change to -> left --- */
if (currnode [trx] == left->rtsib) {
    currnode [trx] = right->lfsib;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        currkno [trx] += left->keyct + 1;
    }

    /* --- update control values in left sibling node --- */
    left->keyct += right->keyct + 1;
    c = bheader[trx].rised_node;
    bheader[trx].rised_node = left->rtsib;
    if (bheader[trx].rightmost == left->rtsib)
        bheader[trx].rightmost = right->lfsib;
    left->rtsib = right->rtsib;
    set_mem(right, NODE, '\0');
    right->prntnode = c;

    /* --- point the deleted node's right brother
    to this left brother --- */
    if (left->rtsib) {
        read_node(left->rtsib, right);
        right->lfsib = lf;
        write_node(left->rtsib, right);
    }

    /* --- remove key from parent node --- */
    par->keyct--;
    if (par->keyct == 0)
        left->prntnode = 0;
    else {
        rt_len = par->keyspace + (par->keyct * ENTLN) - j;
        mov_mem(j + ENTLN, j, rt_len);
    }

    write_node(lf, left);
    write_node(rt, right);
    write_node(p, par);
    free(par);
}

/* ----- Insert key ----- */
int insertkey(tree, x, ad, unique)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int tree;
char *x;
RPTR ad;
int unique;
{
    char k [MXKEYLEN + 1], *a;
    BTREE *yp;
    BTREE *bp;
    int nl_flag, rt_len, j;
    RPTR t, p, sv;
    RPTR *b;
    int lshft, rshft;
    trx = tree;
    if (trx >= MXTREES || handle [trx] == 0)
        return ERROR;
    p = 0;
    sv = 0;
    nl_flag = 0;
    mov_mem(x, k, KLEN);
    t = bheader[trx].rootnode;
    /* ----- Find insertion point ----- */
    if (t) {
        read_node(t, &tmode);
        if (btreescan(&t, k, &a)) {
            if (unique)
                return ERROR;
            else {
                leaflevel(&t, &a, &j);
                curkno [trx] = j;
            }
        }
    }
    else
        curkno [trx] = ((a - tmode.keyspace) / ENTLN)+1;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    currnode [trx] = t;
}
/* ----- Insert key into leaf node ----- */
while (t) {
    ni_flag = 1;
    rt_len = (trnode.keyspace+(bheader[trx].m*ENTLN))-a;
    mov_mem(a, a + ENTLN, rt_len);
    mov_mem(k, a, KLEN);
    b = (RPTR *) (a + KLEN);
    *b = ad;
    if (trnode.nonleaf == FALSE) {
        currnode [trx] = t;
        curkno [trx] = ((a - trnode.keyspace) / ENTLN)+1;
    }
    trnode.keyct++;
    if (trnode.keyct <= bheader[trx].m) {
        write_node(t, &trnode);
        return OK;
    }
    /* --- Redistribute keys between sibling nodes --- */
    lshft = FALSE;
    rshft = FALSE;
    if ((yp=(BTREE *) malloc(NODE))==(BTREE *) NULL) {
        errno = D_OM;
        dberror();
    }
    if (trnode.lfsib) {
        read_node(trnode.lfsib, yp);
        if (yp->keyct < bheader[trx].m &&
            yp->prntnode == trnode.prntnode) {
            lshft = TRUE;
            redist(yp, &trnode);
            write_node(trnode.lfsib, yp);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}
if (lshft == FALSE && trnode.rtsib) {
    read_node(trnode.rtsib, yp);
    if (yp->keyct < bheader[trx].m &&
        yp->prntnode == trnode.pmntnode) {
        rshft = TRUE;
        redist(&trnodes, yp);
        writb_node(trnode.rtsib, yp);
    }
}
free(yp);
if (lshft || rshft) {
    writb_node(t, &trnode);
    return OK;
}
p = nextnode();
/* ----- Split node ----- */
if ((bp = (BTREE *) malloc(NODE)) == (BTREE *) NULL) {
    errno = D_OM;
    dberror();
}
set_mem(bp, NODE, '\0');
trnode.keyct = (bheader[trx].m + 1) / 2;
b = (RPTR *)
    (trnode.keyspace + ((trnode.keyct + 1) * ENTLN) - ADR);
bp->key0 = *b;
bp->keyct = bheader[trx].m - trnode.keyct;
rt_len = bp->keyct * ENTLN;
a = (char *) (b + 1);
mov_mem(a, bp->keyspace, rt_len);
bp->rtsib = trnode.rtsib;
trnode.rtsib = p;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

bp->lfsib = t;
bp->nonleaf = trnode.nonleaf;
a -= ENTLN;
mov_mem(a, k, KLEN);
set_mem(a, rt_len + ENTLN, '\0');
if (bheader[trx].rightmost == t)
    bheader[trx].rightmost = p;
if (t == currnode [trx] &&
    currkno [trx]>trnode.keyct) {
    currnode [trx] = p;
    currkno [trx] -= trnode.keyct + 1;
}
ad = p;
sv = t;
t = trnode.prtmode;
if (t)
    bp->prtmode = t;
else
    {
    p = nextnode();
    trnode.prtmode = p;
    bp->prtmode = p;
    }
write_node(ad, bp);
if (bp->rtsib)
    {
    if ((yp=(BTREE *)malloc(NODE))==(BTREE *) NULL)
        {
        errno = D_OM;
        dberror();
        }
    read_node(bp->rtsib, yp);
    yp->lfsib = ad;
    write_node(bp->rtsib, yp);
    free(yp);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (bp->nonleaf)
    adopt(&bp->key0, bp->keyct + 1, ad);
write_node(sv, &trnode);
if (t) {
    read_node(t, &trnode);
    a = trnode.keyspace;
    b = &trnode.key0;
    while (*b != bp->lfaib) {
        a += ENTLN;
        b = (RPTR *) (a - ADR);
    }
}
free(bp);
}
/* ----- new root ----- */
if (p == 0)
    p = nextnode();
if ((bp = (BTREE *) malloc(NODE)) == (BTREE *) NULL) {
    errno = D_OM;
    dberror();
}
set_mem(bp, NODE, '0');
bp->nonleaf = nl_flag;
bp->prntnode = 0;
bp->rtaiib = 0;
bp->lfaib = 0;
bp->keyct = 1;
bp->key0 = sv;
*((RPTR *) (bp->keyspace + KLEN)) = ad;
mov_mem(k, bp->keyspace, KLEN);
write_node(p, bp);
free(bp);
bheader[trx].rootnode = p;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

if (nl_flag == FALSE) {
    bheader[trx].rightmost = p;
    bheader[trx].leftmost = p;
    currnode [trx] = p;
    currkno [trx] = 1;
}
return OK;
}

/* ----- redistribute keys in sibling nodes ----- */
static void redist(left, right)
BTREE *left, *right;
{
    int n1, n2, len;
    RPTR z;
    char *c, *d, *e;
    BTREE *zp;
    n1 = (left->keyct + right->keyct) / 2;
    if (n1 == left->keyct)
        return;
    n2 = (left->keyct + right->keyct) - n1;
    z = left->prntnode;
    if ((zp = (BTREE *) malloc(NODE)) == FALSE) {
        errno = D_OM;
        dberror();
    }
    c = childptr(right->lfsib, z, zp);
    if (left->keyct < right->keyct) {
        d = left->keyspace + (left->keyct * ENTLN);
        mov_mem(c, d, KLEN);
        d += KLEN;
        e = right->keyspace - ADR;
        len = ((right->keyct - n2 - 1) * ENTLN) + ADR;
        ...
    }
}

```

```

mov_mem(e, d, len);
if (left->nonleaf)
    adopt(d, right->keyct - n2, right->lfaib);
e += len;
mov_mem(e, c, KLEN);
e += KLEN;
d = right->keyspace - ADR;
len = (n2 * ENTLN) + ADR;
mov_mem(e, d, len);
set_mem(d + len, e - d, '\0');
if (right->nonleaf == 0 &&
    left->rtsib == currnode [trx])
    if (currkno [trx] < right->keyct - n2) {
        currnode [trx] = right->lfaib;
        currkno [trx] += n1 + 1;
    }
    else
        currkno [trx] -= right->keyct - n2;
}
else
{
e = right->keyspace + ((n2 - right->keyct) * ENTLN) - ADR;
mov_mem(right->keyspace - ADR, e,
    (right->keyct * ENTLN) + ADR);
e -= KLEN;
mov_mem(c, e, KLEN);
d = left->keyspace + (n1 * ENTLN);
mov_mem(d, c, KLEN);
set_mem(d, KLEN, '\0');
d += KLEN;
len = ((left->keyct - n1 - 1) * ENTLN) + ADR;
mov_mem(d, right->keyspace - ADR, len);
set_mem(d, len, '\0');
if (right->nonleaf)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        adopt(right->keyspace - ADR,
                left->keyct - n1, left->rtsib);
    if (left->nonleaf == FALSE)
        if (right->lfsib == currnode [trx] &&
                currkno [trx] > n1)
    {
        currnode [trx] = left->rtsib;
        currkno [trx] -= n1 + 1;
    }
    else if (left->rtsib == currnode [trx])
        currkno [trx] += left->keyct - n1;
}
right->keyct = n2;
left->keyct = n1;
write_node(z, zp);
free(zp);
}

/* ----- assign new parents to child nodes ----- */
static void adopt(ad, kct, newp)
RPTR *ad;
int kct;
RPTR newp;
{
    char *cp;
    BTREE *tmp;

    if ((tmp = (BTREE *) malloc(NODE)) == (BTREE *) NULL) {
        errno = D_OM;
        dberror();
    }
    while (kct--) {
        read_node(*ad, tmp);

```

```

        tmp->prntnode = newp;
        write_node(*ad, tmp);
        cp = (char *) ad;
        cp += ENTLN;
        ad = (RPTR *) cp;
    }
    free(tmp);
}

/* ----- compute node address for a new node -----*/
static RPTR nextnode()
{
    RPTR p;
    BTREE *nb;
    if (bheader[trx].rised_node) {
        if ((nb = (BTREE *) malloc(NODE)) == (BTREE *) NULL) {
            errno = D_OM;
            dberror();
        }
        p = bheader[trx].rised_node;
        read_node(p, nb);
        bheader[trx].rised_node = nb->prntnode;
        free(nb);
    }
    else
        p = bheader[trx].endnode++;
    return p;
}

/* ----- next sequential key ----- */
RPTR nextkey(tree)
int tree;
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    trx = tree;
    if (currnode [trx] == 0)
        return firstkey(trx);
    read_node(currnode [trx], &trnode);
    if (currkno [trx] == trnode.keyct) {
        if (trnode.rtsib == 0) {
            return (RPTR) 0;
        }
        currnode [trx] = trnode.rtsib;
        currkno [trx] = 0;
        read_node(trnode.rtsib, &trnode);
    }
    else
        currkno [trx]++;
    return *((RPTR *)
        (trnode.keyspace+(currkno[trx]*ENTLN)-ADR));
}

/* ----- previous sequential key ----- */
RPTR prevkey(tree)
int tree;
{
    trx = tree;
    if (currnode [trx] == 0)
        return lastkey(trx);
    read_node(currnode [trx], &trnode);
    if (currkno [trx] == 0) {
        if (trnode.lfsib == 0)
            return (RPTR) 0;
        currnode [trx] = trnode.lfsib;
        read_node(trnode.lfsib, &trnode);
        currkno [trx] = trnode.keyct;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
    currkno [trx]--;
return *((RPTR *)
    (tnode.keyspace + (currkno [trx] * ENTLN) - ADR));
}

/* ----- first key ----- */
RPTR firstkey(tree)
int tree;
{
    trx = tree;
    if (bheader[trx].leftmost == 0)
        return (RPTR) 0;
    read_node(bheader[trx].leftmost, &tnode);
    currnode [trx] = bheader[trx].leftmost;
    currkno [trx] = 1;
    return *((RPTR *) (tnode.keyspace + KLEN));
}

/* ----- last key ----- */
RPTR lastkey(tree)
int tree;
{
    trx = tree;
    if (bheader[trx].rightmost == 0)
        return (RPTR) 0;
    read_node(bheader[trx].rightmost, &tnode);
    currnode [trx] = bheader[trx].rightmost;
    currkno [trx] = tnode.keyct;
    return *((RPTR *)
        (tnode.keyspace + (tnode.keyct * ENTLN) - ADR));
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* ----- scan to the next sequential key ----- */
static RPTR scannext(p, a)
RPTR *p;
char **a;
{
    RPTR cn;
    if (tnode.nonleaf) {
        *p = *((RPTR *) (*a + KLEN));
        read_node(*p, &tnode);
        while (tnode.nonleaf) {
            *p = tnode.key0;
            read_node(*p, &tnode);
        }
        *a = tnode.keyspace;
        return *((RPTR *) (*a + KLEN));
    }
    *a += ENTLN;
    while (-1) {
        if ((tnode.keyspace + (tnode.keyct
            * ENTLN) != *a)
            return fileaddr(*p, *a);
        if (tnode.prtmode == 0 || tnode.rtsib == 0)
            break;
        cn = *p;
        *p = tnode.prtmode;
        read_node(*p, &tnode);
        *a = tnode.keyspace;
        while (*((RPTR *) (*a - ADR)) != cn)
            *a += ENTLN;
    }
    return (RPTR) 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* ---- scan to the previous sequential key ---- */
static RPTR scanprev(p, a)
RPTR *p;
char **a;
{
    RPTR cn;
    if (tnode.nonleaf) {
        *p = *((RPTR *) (*a - ADR));
        read_node(*p, &tnode);
        while (tnode.nonleaf) {
            *p = *((RPTR *)
                (tnode.keyspace+(tnode.keyct)*ENTLN-ADR));
            read_node(*p, &tnode);
        }
        *a = tnode.keyspace + (tnode.keyct - 1) * ENTLN;
        return *((RPTR *) (*a + KLEN));
    }
    while (-1) {
        if (tnode.keyspace != *a) {
            *a -= ENTLN;
            return fileaddr(*p, *a);
        }
        if (tnode.prntnode == 0 || tnode.ifsib == 0)
            break;
        cn = *p;
        *p = tnode.prntnode;
        read_node(*p, &tnode);
        *a = tnode.keyspace;
        while (*((RPTR *) (*a - ADR)) != cn)
            *a += ENTLN;
    }
    return (RPTR) 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

/* ----- locate pointer to child ---- */
static char *childptr(left, parent, btp)
RPTR left;
RPTR parent;
BTREE *btp;
{
    char *c;
    read_node(parent, btp);
    c = btp->keyspace;
    while (*(RPTR *) (c - ADR)) != left)
        c += ENTLN;
    return c;
}

/* ----- current key value ----- */
void keyval(tree, ky)
int tree;
char *ky;
{
    RPTR b, p;
    char *k;
    int i;
    trx = tree;
    b = currnode [trx];
    if (b) {
        read_node(b, &tnode);
        i = currkno [trx];
        k = tnode.keyspace + ((i - 1) * ENTLN);
        while (i == 0) {
            p = b;
            b = tnode.prtnode;
            read_node(b, &tnode);
            for (; i <= tnode.keyst; i++) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        k = tnode.keyspace + ((i - 1) * ENTLN);
        if (*(RPTR *) (k + KLEN) == p)
            break;
    }
}
mov_mem(k, ky, KLEN);
}
}

/* ----- current key ----- */
RPTR curkey(tree)
int tree;
{
    RPTR f = 0;
    trx = tree;
    if (currnode [trx]) {
        read_node(currnode [trx], &tnode);
        f = *(RPTR *)
            (tnode.keyspace+(currkno[trx]*ENTLN)-ADR);
    }
    return f;
}

/* ----- read a btree node ----- */
static void read_node(nd, bf)
RPTR nd;
BTREE *bf;
{
    bseek(nd);
    read(handle [trx], (char *) bf, NODE);
}

/* ----- write a btree node ----- */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

static void write_node(nd, bf)
RPTR nd;
BTREE *bf;
{
    bseek(nd);
    write(handle [trx], (char *) bf, NODE);
}

/* ----- seek to the b-tree node ----- */
static void bseek(nd)
RPTR nd;
{
    if (lseek(handle [trx],
              (long) (NODE+((nd-1)*NODE)),0) == ERROR) {
        errno = D_IOERR;
        dberror();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* -----Check CapsLock ON ----- */
int ChkCapsLock( )
{
    union REGS r;

    r.h.ah = 0x02;
    int86(0x16, &r, &r);
    return( ((r.h.al & 0x40)==0x40) ? 1:0);
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* ----- cdatadic.i ----- */

#define AZTEC      1      // Manx Aztec C86
#define CI_C86     2      // Computer Innovations C86
#define DATALIGHT  3      // Datalight C
#define ECOC       4      // Eco-C88
#define LATTICE    5      // Lattice C
#define LETSC      6      // Mark Williams Let's C
#define MICROSOFT  7      // Microsoft C
#define TURBOC     8      // Turbo C
#define WIZARD     9      // Wizard C

#define ERROR -1
#define OK 0

#ifndef TRUE
#define TRUE 1
#define FALSE 0
#endif

#if COMPILER != LATTICE
#if COMPILER != LETSC
#if COMPILER != CI_C86
#include <errno.h>
#endif
#endif
#endif

#if COMPILER != LETSC
#if COMPILER != DATALIGHT
#if COMPILER != CI_C86
#include <fcntl.h>
#endif
#endif

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#endif

#endif

#include <ctype.h>

extern int errno;

/*--TURBOC--*/
#include <sys\stat.h>
#define OPENMODE O_RDWR+O_BINARY
#define CMODE S_IWRITE

long lseek( );

#define NODE 512 /* length of a B-tree node */
#define RPTR long /* B-tree node and file address */

#define MXFILS 20 /* maximum files in a data base */
#define MXELE 100 /* maximum data elements in a file */
#define MXINDEX 5 /* maximum indexes per file */

/* ----- dbms error codes for errno return ----- */
#define D_NF 1 /* record not found */
#define D_PRIOR 2 /* no prior record for this request */
#define D_EOF 3 /* end of file */
#define D_BOF 4 /* beginning of file */
#define D_DUPL 5 /* primary key already exists */
#define D_OM 6 /* out of memory */
#define D_INDXC 7 /* index corrupted */
#define D_IOERR 8 /* i/o error */

#define MXKEYLEN 80 /* maximum key length for indexes */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#ifndef SCHEMA
/* ----- schema as built for the application ----- */
extern char *dbfiles [ ]; /* file names */
extern char *denames [ ]; /* data element names */
extern char *elmask [ ]; /* data element masks */
extern char eltype [ ]; /* data element types */
extern int ellen [ ]; /* data element lengths */
extern int *file_elo [ ]; /* file data elements */
extern int **index_elo [ ]; /* index data elements */

void mov_mem( ), set_mem( ), fatal( );
void cls_file( );
void build_b( );
void put_char( ), clear_screen( ), cursor( );
void error_message( ), clear_notice( ), post_notice( );

/* ----- data base function definitions ----- */
void db_open( ),db_cls( ),dberror( ),init_rcd( ),
      clr_rcd( ),rcd_fill( ),build_index( );

/* ----- screen driver function definitions ----- */
void init_screen( ),protect( ),edit( ),
      display_template( ),tally( ),put_field( );

#endif

/* ----- file header ----- */
typedef struct fhdr {
    RPTR first_record;
    RPTR next_record;
    int record_length;
} FHEADER;

```

```

/* ----- databas2.i ----- */
void build_b( );          /* Builds a new B-tree          */
RPTR locate( );          /* Searches a B-tree for a key  */
RPTR nextkey( );         /* Gets address of the next key */
RPTR prevkey( );         /* Gets address of the previous key */
RPTR firstkey( );        /* Gets address of first key    */
RPTR lastkey( );         /* Gets address of last key     */
RPTR currkey( );         /* Gets address of current key  */
void file_close( );      /* Closes a data file           */
RPTR new_record( );      /* Adds a record to a file      */
char *malloc( );
void init_index( ), cls_index( ), del_indexes( );
int db_opened = FALSE; /* data base opened indicator */
int curr_fd [MXFILS]; /* current file descriptor */
RPTR curr_a [MXFILS]; /* current record file address */
char *bfs [MXFILS]; /* file i/o buffers */
int bfd [MXFILS] [MXINDEX];
char dbpath [64];
int notice_posted = 0;
int prev_col = 0, prev_row = 0;

/* ----- open the data base ----- */
void db_open(path, fl)
char *path;
int *fl;
{
    char fnm [64];
    int i;
    if (!db_opened) {
        for (i = 0; i < MXFILS; i++)
            curr_fd [i] = -1;
        db_opened = TRUE;
    }
}

```



```

strcpy(dbpath, path);
while (*fl != -1) {
    sprintf(fnm, "%s%.8s.dat", path, dbfiles [*fl]);
    curr_fd [*fl] = file_open(fnm);
    init_index(path, *fl);
    if ((bfs [*fl] = malloc(strlen(*fl))) == NULL) {
        errno = D_OM;
        perror( );
    }
    fl++;
}
}

/* ----- add a record to a file ----- */
int add_red(f, bf)
int f;
char *bf;
{
    RPTR ad;
    extern RPTR new_record( );
    int rtn;
    if ((rtn = relate_red(f, bf)) != ERROR) {
        ad = new_record(curr_fd [f], bf);
        if ((rtn = add_indexes(f, bf, ad)) == ERROR) {
            errno = D_DUPL;
            delete_record(curr_fd [f], ad);
        }
    }
}

return rtn;
}

/* ----- find a record in a file ----- */
int find_red(f,k,key,bf)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int f;           /* file number      */
int k;           /* key number      */
char *key;       /* key value       */
char *bf;        /* buffer for record */
{
    RPTR ad;
    if ((ad = locate(treeno(f,k), key)) == 0) {
        errno = D_NF;
        return ERROR;
    }
    return getrcd(f, ad, bf);
}

/* ----- verify that a record is in a file ----- */
int verify_rcd(f,k,key)
int f;           /* file number      */
int k;           /* key number      */
char *key;       /* key value       */
{
    if (locate(treeno(f,k), key) == (RPTR) 0) {
        errno = D_NF;
        return ERROR;
    }
    return OK;
}

/* ----- find the first indexed record in a file ----- */
int first_rcd(f,k,bf)
int f;           /* file number      */
int k;           /* key number      */
char *bf;        /* buffer for record */
{
    RPTR ad;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if ((ad = firstkey(treeno(f,k))) == 0) {
    errno = D_EOF;
    return ERROR;
}
return getred(f, ad, bf);
}

```

```

/* ----- find the last indexed record in a file ----- */

```

```

int last_red(f,k,bf)

```

```

int f;                /* file number          */

```

```

int k;                /* key number          */

```

```

char *bf;             /* buffer for record   */

```

```

{

```

```

    RPTR ad;

```

```

    if ((ad = lastkey(treeno(f,k))) == 0) {

```

```

        errno = D_BOF;

```

```

        return ERROR;

```

```

    }

```

```

    return getred(f, ad, bf);

```

```

}

```

```

/* ----- find the next record in a file ----- */

```

```

int next_red(f,k,bf)

```

```

int f;                /* file number          */

```

```

int k;                /* key number          */

```

```

char *bf;             /* buffer for recor    */

```

```

{

```

```

    RPTR ad;

```

```

    if ((ad = nextkey(treeno(f, k))) == 0) {

```

```

        errno = D_EOF;

```

```

        return ERROR;

```

```

    }

```

```

        return getrcd(f, ad, bf);
    }

/* ----- find the previous record in a file ----- */
int prev_rcd(f,k,bf)
int f;          /* file number */
int k;          /* key number */
char *bf;      /* buffer for record */
{
    RPTR ad;

    if ((ad = prevkey(trceno(f,k))) == 0) {
        errno = D_BOF;
        return ERROR;
    }
    return getrcd(f, ad, bf);
}

/* ----- return the current record to the data base ----- */
int rtn_rcd(f, bf)
int f;
char *bf;
{
    int rtn;

    if (curr_a [f] == 0) {
        errno = D_PRIOR;
        return ERROR;
    }

    if ((rtn = relate_rcd(f, bf)) != ERROR) {
        del_indexes(f, curr_a [f]);
        if ((rtn = add_indexes(f, bf, curr_a [f])) == OK)
            put_record(curr_fd [f], curr_a [f], bf);
    }
}

```

```

        else

                errno = D_DUPL;

    }

    return rtn;

}

```

```

/* ----- delete the current record from the file ----- */

```

```

int del_rcd(f)

```

```

int f;

```

```

{

```

```

    if (curr_a [f]) {
        del_indexes(f, curr_a [f]);
        delete_record(curr_fd [f], curr_a [f]);
        curr_a [f] = 0;
        return OK;
    }

```

```

    errno = D_PRIOR;

```

```

    return ERROR;

```

```

}

```

```

/* ----- find the current record in a file ----- */

```

```

int curr_rcd(f,k,bf)

```

```

int f;                /* file number */

```

```

int k;                /* key number */

```

```

char *bf;            /* buffer for record */

```

```

{

```

```

    RPTR ad;

```

```

    extern RPTR currkey( );

```

```

    if ((ad = currkey(treeo(f,k))) == 0) {

```

```

        errno = D_NF;

```

```

        return ERROR;

```

```

    }

```

```

    getrcd(f, ad, bf);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    return OK;
}

/* ----- got the next physical sequential record from the file ----- */
int seqrecd(f, bf)
int f;
RPTR *bf;
{
    RPTR ad;
    int rtn;
    do {
        ad = ++curr_a [f];
        if ((rtn = (rel_rec(f,ad,bf)))==ERROR && errno!=D_NF)
            break;
    } while (errno == D_NF);
    return rtn;
}

/* ----- close the data base ----- */
void db_cls( )
{
    int f;
    for (f = 0; f < MXFILS; f++)
        if (curr_fd [f] != -1) {
            file_close(curr_fd [f]);
            cls_index(f);
            free(bfs[f]);
            curr_fd [f] = -1;
        }
    db_opened = FALSE;
}

```

```

/* ----- data base error routine ----- */
void dberror( )
{
    static char *ers [ ] = {
        "Record not found",
        "No prior record",
        "End of file",
        "Beginning of file",
        "Record already exists",
        "Not enough memory",
        "Index corrupted",
        "Disk i/o error"
    };
    static int fat [ ] = {0,1,0,0,0,1,1,1};
    error_message(ers [errno-1]);
    if (fat [errno-1])
        exit(1);
}

/* ----- compute file record length ----- */
int rlen(f)
int f;
{
    return epos(0, file_cle [f]);
}

/* ----- initialize a file record buffer ----- */
void init_rcd(f, bf)
int f;           /* file number */
char *bf;       /* buffer */
{
    circd(bf, file_cle[f]);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

/* ----- set a generic record buffer to blanks ----- */
void clrrod(bf, els)
char *bf;          /* buffer */
int *els;          /* data element list */
{
    int ln, i = 0, el;
    char *rb;
    while (*(els + i)) {
        el = *(els + i);
        rb = bf + epos(el, els);
        ln = ellen [el - 1];
        while (ln--)
            *rb++ = ' ';
        *rb = '\0';
        i++;
    }
}

/* ----- move data from one record to another ----- */
void rod_fill(s, d, slist, dlist)
char *s;          /* source record buffer */
char *d;          /* destination record buffer */
int *slist;       /* source data element list */
int *dlist;       /* destination data element list */
{
    int *s1, *d1;
    s1 = slist;
    while (*s1) {
        d1 = dlist;
        while (*d1) {
            if (*s1 == *d1)

```



```

        strcpy(d+epos(*d1,dlist),
              s+epos(*s1,slist));

        d1++;
    }
    s1++;
}

}

/* ----- compute relative position of a data element within a record -----*/
int opos(ei, list)
int ei;          /* element number */
int *list;      /* record element list */
{
    int len = 0;
    while (ei != *list)
        len += ellen [(*list++)-1] + 1;
    return len;
}

/* ----- index management functions ----- */
/* ---- initialize the indices for a file ---- */
static void init_index(path, f)
char *path;     /* where the data base is */
int f;         /* file number */
{
    char xname [64];
    int x = 0;
    while (*(index_clo [f] + x)){
        sprintf(xname, "%s%.8s.x%02d",path,dbfiles[f],x+1);
        if ((bfd [f] [x++] = btree_init(xname)) == ERROR) {
            printf("\n%s failure, Please run \"Rework\"
                \" option.\", xname);
            errno = D_INDXC;
        }
    }
}

```

```

        dberror( );
    }
}

/* ---- build the indices for a file ---- */
void build_index(path, f)
char *path;
int f;          /* file number */
{
    char xname [64];
    int x = 0, x1;
    int len;
    while (*(index_els [f] + x)){
        sprintf(xname, "%s%.8s.x%02d", path, dbfiles[f], x+1);
        len = 0;
        x1 = 0;
        while (*(index_els [f] + x) + x1))
            len += ellon [*(index_els [f] + x) + (x1++)-1];
        build_b(xname, len);
        x++;
    }
}

/* ----- close the indices for a file ----- */
static void cls_index(f)
int f;
{
    int x = 0;
    while (*(index_els [f] + x)){
        if (bfd [f] [x] != ERROR)
            btree_close(bfd [f] [x]);
        x++;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

/* ---- add index values from a record to the indices ---- */
int add_indexes(f, bf, ad)
int f;
char *bf;
RPTR ad;
{
    int x = 0;
    int i;
    char key [MXKEYLEN];
    while (*(index_ele [f] + x)){
        *key = '\0';
        i = 0;
        while (*(index_ele [f] + x) + i)
            strcat(key,
                bf +
                epos(*(index_ele[f]+x)+(i++)),file_ele [f]);
        if (insertkey(bfd [f] [x], key, ad, !x) == ERROR)
            return ERROR;
        x++;
    }
    return OK;
}

```

```

/* --- delete index values in a record from the indices --- */
static void del_indexes(f, ad)
int f;
RPTR ad;
{
    char *bf;
    int x = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int i;
char key [MXKEYLEN];
if ((bf = malloc(rlen(f))) == NULL) {
    errno = D_OM;
    dberror();
}
get_record(curr_fd [f], ad, bf);
while (*(index_ele [f] + x)){
    *key = '\0';
    i = 0;
    while (*(*(index_ele [f] + x) + i))
        strcat(key,
            bf +
            epos(*(*(index_ele[f]+x)+(i++)), file_ele [f]));
    deletekey(bfd [f] [x++], key, ad);
}
free(bf);
}

/* ---- compute tree number from file and key number ---- */
static int treeno(f, k)
int f, k;
{
    return bfd [f] [k - 1];
}

/* ---- validate the contents of a record where its file is
related to another file in the data base ----- */
static int relate_red(f, bf)
int f;
char *bf;
{
    int fx = 0, mx, *fp;
    static int ff[] = {0, -1};

```

```

char *cp;
while (dbfiles [fx])      {
    if (fx != f && (*(index_elo [fx] + 1) == 0) {
        mx = *(index_elo [fx]);
        fp = file_elo [f];
        while (*fp)      {
            if (*fp == mx)  {
                cp = bf + epos(mx, file_elo [f]);
                if (data_in(cp))  {
                    if (curr_fd[fx] == -1)  {
                        *ff = fx;
                        db_open(dbpath, ff);
                    }
                    if(verify_rcd(fx,1,cp)==ERROR)
                        return ERROR;
                }
                break;
            }
            fp++;
        }
        fx++;
    }
}
return OK;
}

/* ----- test a string for data. return TRUE if any ---- */
static int data_in(c)
char *c;
{
    while (*c == ' ')
        c++;
    return (*c != '\0');
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* ----- get a record from a file ----- */
static int getrcd(f, ad, bf)
int f;
RPTR ad;
char *bf;
{
    get_record(curr_fd [f], ad, bf);
    curr_a [f] = ad;
    return OK;
}

extern FHEADER fh [ ];
/* ---- find a record by relative record number ----- */
static int rel_rcd(f, ad, bf)
int f;
/* file number */
RPTR ad;
int *bf;
{
    errno = 0;
    if (ad >= fh [curr_fd [f]].next_record) {
        errno = D_EOF;
        return ERROR;
    }
    getrcd(f, ad, bf);
    if (*bf == -1) {
        errno = D_NF;
        return ERROR;
    }
    return OK;
}

/* ----- error message ----- */
void error_message(s)

```

```

char *s;
{
    printf("%a");
    post_notice(s);
}

/* ----- clear notice line ----- */
void clear_notice( )
{
    int i;
    if (notice_posted) {
        bar_del(5,maxy-65,maxx-5,maxy-38,bkcolor);
        notice_posted = FALSE;
        cursor(prev_col, prev_row);
    }
}

/* ----- post a notice ----- */
void post_notice(s)
char *s;
{
    clear_notice( );
    setcolor(14); //YELLOW
    setwriteMode(0); //COPY_PUT
    ThaiStr(maxx/2-(8*QtyMidStr(s)/2),maxy-60,s);
    cursor(prev_col, prev_row);
    notice_posted = 1; //Yes
}

/* ----- Move a block ----- */
void mov_mem(s, d, l)
char *s, *d;
int l;

```

```

{
    if (d > s)
        while (l--)
            *(d + l) = *(s + l);
    else
        while (l--)
            *d++ = *s++;
}

/* ----- Set a block to a character value ----- */
void set_mem(s, l, n)
char *s, n;
int l;
{
    while (l--)
        *s++ = n;
}

/*
 * Convert a file name into its file token.
 * Return the token,
 * or ERROR if the file name is not in the schema.
 */
int filename(fn)
char *fn;
{
    char fname[32];
    int f;
    void name_cvt( );
    name_cvt(fname, fn);
    for (f = 0; dbfiles [f]; f++)
        if (strcmp(fname, dbfiles [f]) == 0)
            break;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

if (dbfiles [f] == 0)    {
    fprintf(stderr, "\nNo such file as %s", fname);
    return ERROR;
}
return f;
}

/* ----- convert a name to upper case ----- */
void name_cvt(c2, c1)
char *c1, *c2;
{
    while (*c1)    {
        *c2 = toupper(*c1);
        c1++;
        c2++;
    }
    *c2 = '\0';
}

```

```

/*-----*/
/*          DIC_COOR.I          */
/* FUNCTIONT : used for initialize short coordinate-variable */
/*-----*/

#define mx2 maxx/2
#define mx4 maxx/4
#define mx5 maxx/5
#define mx6 maxx/6
#define mx8 maxx/8
#define mx_4 maxx-maxx/4
#define mx_5 maxx-maxx/5
#define my2 maxy/2
#define my4 maxy/4
#define my5 maxy/5
#define my8 maxy/8
#define my_5 maxy-maxy/5
#define my_8 maxy-maxy/8

```

```

/*-----*/
/*          DICSNOW.I          */
/*          FUNCTION : creat snow upon window */
/*-----*/

#define RAND_MAX 0x7FFFU

#ifdef cplusplus
inline int _Cdecl random(int __num)
    { return(int)(((long)rand()*__num)/(RAND_MAX+1));}
#else
#define random(num)(int)(((long)rand()*(num))/(RAND_MAX+1))
#endif

extern void ToView(), bar_color(), Pansc();
void WinSnow(int gl, int gt, int gr, int gb, int vport,
int bk, int fg, int mode) {
    int seed = 1958;
    int seed2= 2345;
    register int i,j,x, y, h, w,bk2;
    unsigned long mxdot;
    struct viewporttype vp0,vpt;

    ToView(vport);
    getviewsettings(&vp0);
    setviewport(vp0.left+gl,vp0.top+gt,vp0.left+gr,vp0.top+gb,1);
    getviewsettings(&vpt);
    h = vpt.bottom - vpt.top;
    w = vpt.right  - vpt.left;
    setwriteMode(COPY_PUT);
    mxdot=(unsigned long)w*h/90;
    for(j=0; j<4; j++)    {
        srand( j%2==0?seed:seed2 );    // Restart random # function
        for( i=0; i<mxdot; ++i){    // Put 5000 pixels on screen
            x = 1 + random( w - 1 );    // Generate a random location
            y = 1 + random( h - 1 );

```

```

bk2=j<=2?BLACK:bk;
if(j<1)
    bar_color(x-2,y-2,x+2,y+2,bk2,0,0);
else
    bar_color(x-5,y-5,x+5,y+5,bk2,0,0);
}
}
bar_color(0,0,w,h,bk,fg,mode);
ToView(vport);
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
/*          KEYS.I          */
/*    FUNCTION : define scancode    */
/*-----*/

#define HT 9
#define RUBOUT 8
#define BELL 7
#define ESC 27

#define F1 187
#define F2 188
#define F3 189
#define F4 190
#define F5 191
#define F6 192
#define F7 193
#define F8 194
#define F9 195
#define F10 196

#define HOME 199
#define UP 200
#define PGUP 201
#define BS 203 /*left*/
#define FWD 205 /*right*/
#define END 207
#define DN 208
#define PGDN 209
#define INS 210
#define DEL 211

```

```

/*-----*/
/*          PH_CRT.I          */
/*  FUNCTION : all screen-manipulation function */
/*-----*/

int insert_mode = FALSE, ModeTemp2=0,HorFast=4,VerFast=8;
char MAIN_WORD[14]="a"; //usable any character
extern int prev_col, prev_row; /* current cursor location */
extern int screen_displayed; /* template displayed flag */

#define FIELDCHAR '_' /* field filler character */
#define CH_COLOR WHITE
#define CH_INVER LIGHTGRAY
#define CH_TEMPLATE BLUE
#define Xmsk 14
#define Ymsk 5
#define TitleBlank 4
#define List_Limit 6
#define DEPART 6 //enable any numbers

#define GET_PH -1
#define GET_VOC 8 //choose : find, add, write & game
#define GET_WORD 6
#define EDIT_PH 2 //PH=Phase
#define WR_PH 10
#define ADD_PH 3
#define FULL_PH 4
#define NO_ADD 7
#define NEW_PH 999
#define ADD_LIST 5
extern m_result *rodent;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

struct {
int prot;
int (*edits)( );
} sb [MXELE];
int *elist;
char *bf;
char *tname;
extern int sh_cursor( ), ToSys( );
void right_justify( ), ShowView2(void), ClearViewPort( ),
right_justify_zero_fill( ), left_justify_zero_fill( ),
disp_element( ), insert_status( ), put_fchar( ), data_coord( ),
tone( ), saymode( ), say_command( ), // ThaiStrJustify( ),
WindowExpand( ), WindowHide( ), ToView( ), Pause( ),
WinSmall( ), BarAttrib( ), View2CPY( ), ClsStatusKey( ),
MainTest( ), Sh_TestScr( );
extern void bar_color( ), ds_main( );
int ChkMainWord( ), IsNullStr( );
void put_field_xy( );
char *ToFieldNo( );
int MskLen( );
int ChkYN( );

/* ----- initialize the screen process ----- */
void init_screen(name, els, bfr)
char *name;
int *els;
char *bfr;
{
tname = name;
elist = els;
bf = bfr;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* ----- set the protect flag for a screen field ----- */
void protect(el, tf)
int el, tf;
{
sb[elp(el)].prot = tf;
}

/* ---- set the field edit function for a screen field --- */
void edit(el, func)
int el, (*func)( );
{
sb[elp(el)].edits = func;
}

/* ---- compute the relative position
of an element on a screen ----- */
static int elp(el)
int el;
{
int i;

for (i = 0; *(elist + i); i++)
    if (el == *(elist + i))
        break;

return i;
}

/* ----- Display the crt template ----- */
/* ----- Show field_name and mask of record ----- */
void display_template( )
{
int /*el, ct, *f;

```



```

char dotag[16], *cp1, *cp2;
/*-----
setcolor(CH_TEMPLATE);
if(wr_screen==1) {
    screen_displayed = TRUE;
    ct = no_flds( );
    ThaiStr(Xmak, Ymak+my5, tname);
    cursor(1, TitleBlank);
    for (i = 0; i < ct; i++){
    el = *(elist + i) - 1;
    cp1 = donames[el];
    cp2 = dotag;
    while (*cp1 && cp2 < dotag + sizeof dotag - 1){
    *cp2++ = *cp1 == '_'?' ': *cp1;
    cp1++;
    }
    *cp2 = '\0';
    moveto(Xmak, gety( )+30);
    ThaiStr(Xmak, gety( ), dotag);
    }
    }
-----*/
char *PhaseTemplate[ ] = {
    "ลำดับที่",
    "คำวลี",
    "ความหมาย",
    "ตัวอย่าง",
    "ตัวอย่าง(ต่อ)",
    "คำแปล"
};
setcolor(CH_TEMPLATE);
if(wr_screen==1) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

screen_displayed = TRUE;
ThaiStr(Xmask,Ymak+my5,tname);
cursor(1,TitleBlank);
for(i=0; i<6; i++)
    ThaiStr(Xmask, gety( )+(30*(i+1)), PhaseTemplate[i]);
}
insert_status( );
m_show( );
}

/* ----- Process data entry for a screen template. ----- */
int data_entry( )
{
int (*validfunct)( );
int field_ctr, exicode, el, i;
int field_ptr =0, done = FALSE, isvalid;
char *bfptr, *bfptr2, phase_list;
char *entry_msg[ ]={
    "ฝึกติกาการพิมพ์คำศัพท์..."
};

if(ModeTemp==GET_WORD)
    wr_screen=0;
display_template( );
if(ModeTemp==GET_WORD) {
    /// get mainword
    bar_color(mx5+4,mxy/3+11,mx5+111,mxy/3+37,CH_INVER,WHITE,3);
    say_command("การพิมพ์คำศัพท์จะต้องเป็นคำเดียว และเ"
"คยทั่วไปมักจะเป็นคำกริยา");
}
else if(ModeTemp==ADD_PH || ModeTemp==EDIT_PH)
    say_command("เมนูนี้ไม่มีคำสั่งออกโดยตรง ท่านต")

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"ของออกโดยวิธีบันทึกข้อมูลเสียก่อนเท่านั้น");

```

m_hide( );
tally( );
sh_image_voc((ModeTemp==EDIT_PH || ModeTemp==ADD_PH)?mx_4+28:mx_4+3,
              (ModeTemp==EDIT_PH || ModeTemp==ADD_PH)?maxy/5+5:maxy-maxy/6-79);
m_show( );
field_ctr = no_fds( );
/* ---- collect data from keyboard into screen ---- */
    NumPress=NumChoose=-1;
while (done == FALSE){
if(ModeTemp==GET_WORD) {
    m_hide( );
    bar_color(mx5+4,maxy/3+11,mx5+111,maxy/3+37,CH_INVER,WHITE,3);
    m_show( );
}
bfptr = bf + epos(elist[field_ptr], elist);
el = *(elist + field_ptr) - 1;
validfunct = sb[field_ptr].edits;
data_coord(el + 1);

if (sb[field_ptr].prot == FALSE){
    exitcode =
        read_element(eltype[el], elmask[el], bfptr);

    isvalid = (exitcode != ESC && validfunct) ?
        (*validfunct)(bfptr,exitcode) : OK;
}

else{
    exitcode = FWD;
isvalid = OK;
}

if (isvalid == OK)

```

```

{
switch (exitcode){// passed edit
    case -14:    //garbage
    case 444:    //garbage
        NumChoose=3;
        ButtonTakeOver( );
        View2CPY( );
    case DN:// cursor down key
        if(ModeTemp!=EDIT_PH && ModeTemp!=ADD_PH){
            clear_record( );
            break;
        }
    case FWD:    // ->
        if(field_ptr+1==field_ctr)
            field_ptr=0;
        else
            field_ptr++;
        break;
//-----case -3 ALWAYS next to '\n'-----//
    case '\n':    // enter/return
        if(NumChoose < 0 ) ///&& ModeTemp!=EDIT_PH)
            break;
        else
            exitcode=(NumChoose+1)* -1;
        if(exitcode!=-3) {
            done=FALSE;
            break;
        }
    case F4:
        if(ModeTemp!=GET_WORD)
            break;
        exitcode=-3;

```

case -3:

```

NumChoose=2;
ButtonTakeOver( );
View2CPY( );
if(field_ptr+1==field_ctr)
    field_ptr=0;
else if(ModeTemp!=GET_WORD)
    field_ptr++;
if(ModeTemp==GET_WORD)
{
    if(!isspace(*bfptr)) {
        i=0;
        while(*(bfptr+i) && *(bfptr+i)!=' ')
            i++;
        while(*(bfptr+i) && *(bfptr+i)!='\n')
            i++;
        if(*(bfptr+i) && *(bfptr+i)!=' '){
            post_notice(entry_msg[0]);
            clear_record( );
            break;
        }
        else {
            strcpy(MAIN_WORD,bfptr);
            for(i=0; i<strlen(MAIN_WORD); i++)
                *(MAIN_WORD+i) = tolower(*(MAIN_WORD+i));
        }
    }
}
else {
    post_notice(entry_msg[0]);
    clear_record( );
    break;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Set_Graphic_Cursor(PAUSED);
if(ChkMainWord(MAIN_WORD)==Yes)
done = endstroke(exitcode=DEPART);
if(ModeTemp==GET_WORD) {
    field_ptr=0;
    rdout( );
    wr_screen=0;
}
if(ModeTemp==FULL_PH ||
ModeTemp==NO_ADD) {
    done = endstroke(exitcode=DEPART);
    FILE_TEM = No;
}
if(ModeTemp==ADD_PH) {
    FILE_TEM = No;
    done = endstroke(exitcode=ADD_PH);
}
if(ModeTemp==NEW_PH) {
    FILE_TEM = No;
    done = endstroke(exitcode=ADD_PH);
}
if(ModeTemp==ADD_LIST) {
    ModeTemp=ADD_PH;
    done=endstroke(exitcode=DEPART);
    FILE_TEM=No;
}
if(ModeTemp==EDIT_PH) {
    m_show( );
    done = TRUE;
}
if(ModeTemp==DIC_PH) {
    //FILE_TEM not result in next status

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        exitcode = DIC_PH;
        done=TRUE;
        break;
    }
}

break;
case UP:
    // cursor up key
    if(ModeTemp==GET_WORD) {
        clear_record( );
        break;
    }
case -25:
case BS:/* back space */
    NumChoose=1; //EDIT_PH ==> UP-function
    ButtonTakeOver( );
    View2CPY( );
    if (field_ptr == 0)
        field_ptr = field_ctr - 1;
    else
        field_ptr--;
    break;
default:
    done = endstroke(exitcode);

    break;
}
}
}
return (exitcode);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* ----- Compute the number of fields on a template ----- */
static int no_flds( )
{
int ct = 0;

while (*(olist + ct))
ct++;

return ct;
}

/* ----- compute data element field coordinates ----- */
static void data_coord(el)
int el;
{
prev_col = Xmsk;
prev_row = elp(el)+Ymsk;
}

/* ----- read data element from keyboard ----- */
static int read_element(type, msk, bfr)
char type, *mak, *bfr;
{
char *mask=mak, *buff=bfr;
int done = FALSE, column=prev_col,maxcol,mincol,c,c2;
register int i,j,k,m,p;
char *read_msg[ ]= {
    "โปรดตรวจสอบคำที่พิมพ์อีกครั้ง..."
};

while (*mask != FIELDCHAR) {
    prev_col++;
    mask++;
}
}

```



```

i=0; while(*(mask+ i++));
maxcol=prev_col+i-1;
mincol=prev_col;

saymode(type!='A'?ENG:THAI);
    while (TRUE)    {
        cursor(prev_col, prev_row);    //-- CURSOR ---
        Set_Graphic_Cursor(GLOVE);
        m_show( );
        c=GetKeyMou(getx( ),gety( ));
        if(notice_posted==Yes) {
clear_notice( );
        notice_posted=No;
    }
    switch (c)    {
        case '\b':    //BackSpace
        case BS:    //LeftArrow
            if (buff == bfr) {
                c2= c == BS;
                done = FALSE;
                break;
            }
            if(c=='\b')    //backspace
            {
                // Del Mid Char
                if(DelVowel(*(buff-1))==0)
                {
                    mov_mem(buff, buff-1, strlen(buff));
                    *(buff+(strlen(buff)-1)) = ' ';
                    if(*(mask-1)!=FIELDCHAR)
                        {--prev_col; --mask;}
                    cursor(--prev_col, prev_row);
                }
            }
        }
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

disp_element(--buff,--mask);
}
else
{
mov_mem(buff, buff-1, strlen(buff));
*(buff+(strlen(buff)-1)) = ' ';
--buff;
}
}

if(c==BS) /*Left-Arrow*/
{
if(prev_col > mincol)
{
while(IsMidChar(*(buff-1))!=1)
buff--;
if(IsMidChar(*(buff-1))==1)
{
prev_col--; buff--; mask--;
}
while(*mask && *mask!=FIELDCHAR &&
prev_col > mincol)
{
if(IsMidChar(*mask)==1) prev_col--;
mask--;
}
}
}
c2=c;
break;
case DEL:
/// first Mid char

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov_mem(buff+1, buff, strlen(buff+1));
*(buff+(strlen(buff)-1)) = '\0';
//// UpLow less than 3
while(IsMidChar(*buff)!=1)
{
    mov_mem(buff+1, buff, strlen(buff+1));
    *(buff+(strlen(buff)-1)) = '\0';
}
cursor(prev_col, prev_row);
disp_element(buff,mask);
break;
case FWD: // Right arrow
//protect spacebar and end of line
if(ModeTemp==GET_WORD && IsNullStr(*(buff+1),Yes)==Yes)
    break;
if(IsNullStr(*buff,Yes)==Yes)
    break;
if(prev_col < maxcol)
{
    while(IsMidChar(*(buff+1))!=1) {
        if(IsNullStr(*(buff+1),Yes)==Yes)
            break;
        buff++;
    }
    if(IsMidChar(*(buff+1))!=1) {
        if(*(buff+1)!='\0') {
            prev_col++;
            buff++;
            mask++;
        }
    }
    else { //full buffer
        while(IsMidChar(*buff)!=1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        buff--;
        //full_toll at all condition of mask
        if(IsNullStr(*(mask+1),(*(mask+1)=='\0'?Yes:No))==Yes)
            break;
        prev_col++;
        mask++;
    }
}

while(*mask && *mask!=FIELDCHAR &&
      prev_col<maxcol) {
    if(IsMidChar(*mask)==1) prev_col++;
    mask++;
}
//end if(prev_col< maxcol)
break;
case F1:
case -15: //mouse
case -1:
case -2:
    switch(c) {
        case F1:
        case -1:
            NumChoose=0;
            break;
        case -2:
            NumChoose=1;
            break;
        case -15:
            NumChoose=4;
            break;
        default:
            break;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
H_A:
ButtonTakeOver( );
View2CPY( );
HELPABOUT=1;
HelpAbout(c);
HELPABOUT=!1;
break;

```

```

//THAI:ENG
case -44:
case F10:
    if(ModeTemp!=GET_WORD && VocMode==WR_PH) {
        NumChoose=3;
        ButtonTakeOver( );
        View2CPY( );
    }
    if(type!='A' && say_mode==ENG) {
        error_message("ผิดคำเฉพาะโหมด English..");
        break;
    }
    saymode((say_mode==THAI?ENG:THAI));
    break;

```

```

//INSERT:OVERWRITE
case -55:
case INS:
    INS_WR:
    if(ModeTemp!=GET_WORD && VocMode==WR_PH) {
        NumChoose=4;
        ButtonTakeOver( );
        View2CPY( );
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    insert_mode ^= TRUE;
    insert_status( );
    break;

case '\': // horizontal tab
    ButtonTab(6);
    ToView(2);
    setwritemode(COPY_PUT);
    c2=c=(NumChoose+1)* -1;
    break;

case 'v':
    switch(NumChoose) {
        case 2if(NumChoose==2 && (ModeTemp==EDIT_PH ||
            ModeTemp==ADD_PH)) {
            c=DN;
            ButtonTakeOver( );
            View2CPY( );
            break;
        }
        else
            goto CHAR;
    }
case 0:
case 1:
case 4:
    if(NumChoose==0) {
        c=-1;
        goto H_A;
    }
    else
        if(NumChoose==1) {
            c=-2;
            if(ModeTemp==GET_WORD)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        goto H_A;
    else {
        c=UP;
        goto CHAR;
    }
}

else {
    c=-15;
    if(ModeTemp==GET_WORD)
        goto H_A;
    else //EDIT_PH
        goto INS_WR;
}

case 3:
case 5:
case 6:
    c=NumChoose%3!=0?-6:(NumChoose==3?-14:-7);
    goto CHAR; //-14:cancel -6dos -7exit
case -1: //Not yet press <Tab> but
        //press ENTER
        //c=-3; //enter
        if(ModeTemp==EDIT_PH || ModeTemp==ADD_PH) {
            c=DN;
            break;
        }

        if(ModeTemp!=GET_STU)
            post_notice("โปรดใช้คีย์คอน"OKAY" หรือกด F4");
        break;
    default:
        goto CHAR;
}

case F2:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case F5:
case F8:
case F9:
case HOME:
case END:
case UP:
case DN:
case PGUP:
case PGDN:
    if(ModeTemp==GET_WORD)
        break;
default:
    CHAR:
    c2=c;
    c=asc_scancode(c,say_mode); //47
    if (endstroke(c2)) {
        done = TRUE;
        break;
    }
    else if(IsNullStr(*buff,Yes)==Yes) {
        tone( );
        break;
    }
    if(type == 'B' && !isdigit(c) && !isspace(c))
    {
        error_message("ตัวเลข หรือ space_bar เท่านั้น");
        break;
    }
    if(type != 'A' && type != 'E' && !isdigit(c) &&
    type != 'B') {
        error_message("ตัวเลขเท่านั้น");
        break;
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

    }
    if(type == 'E' && isdigit(c)){
        post_notice("ท่านพิมพ์ตัวเลข...");
        tone( );
    }
}

// UpDownChar k=key_in j=buff_char
if((k=IsMidChar(c))!=1)
{
    j=(unsigned char)*(buff-1);
    if(j == ' ' || j == '\0' || prov_col<=Xmak ||
        (j>223 && j<229) ) // 226=SLA_O..
    { tone( );
      break;
    }
    if(k==5) //MaTo + Han-ar-kard
    { m=0;j=1; //3
      do { //2
          j=IsMidChar(*(buff-i)); i++;
          if(j==3 || j==4) m++;
      } while(j!=2 && j!=1); //2
      switch(m) { //1
          case 0: // prev_char=normal
              mov_mem(buff, buff+2, strlen(buff)-2);
              *buff=(int)209; *(buff+1)=(int)233;
              break;
          case 1: //prev_char=up-vowel x 1
              mov_mem(buff, buff+1, strlen(buff)-1);
              DelVowel(*(buff-1));
              *(buff-1)=(int)209; *buff=(int)233;
              break;
          case 2: //prev_char=up-vowel x 2
              DelVowel(*(buff-2));

```

```

DelVowel(*(buff-1));
*(buff-2)=(int)209; *(buff-1)=(int)233;
break;
} //1
disp_element(buff-m,mask-m+1);
buff+=2-m;
if (!*mask)
c2 = c = FWD;
break;
} // End chech k=5..? //-3

for(i=1; i<4; i++) // write_protect 4 level simultaneous
{
if(i==3) { // type sla-oo.. after sla-co.. and mai-to..
disp_element(buff-i+1,mask); //del vowels before write its at later
tone( );
post_notice(read_msg[0]);
if(!*mask)
c2 = c = FWD;
break;
}
if( k>=(j=IsMidChar(*(buff-i))) || j==1 ) {
mov_mem(buff-i+1, buff-i+2, strlen(buff-i+1)-1 );
*(buff-i+1) = c;
disp_element(buff-i+1,mask-i+1);
buff++;
if(!*mask)
c2 = c = FWD;
if(IsMidChar(*(buff-i-1))==2 && k==3) {
post_notice(read_msg[0]);
tone( );
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

break;
} //4
if(k==j)
{

DelVowel(*(buff-i));
*(buff-i)=c;

disp_element(buff-i,mask-i+1);

if(!*mask) c2 = c = FWD;
break;
}
// if k<j ;NoFound MidChar or sla_oo....-> repeat_Loop
if(IsMidChar(*(buff-i))==3 && k==2) { //check level 2 & 3
simultanous
for(p=1; p<3; p++) {
if(IsMidChar(*(buff-i+p))>2) //write after from del prev-loop
DelVowel(*(buff-i+p));
}
if(IsMidChar(*(buff-i-1))!=2) { //No sla_oo or sla_u
mov_mem(buff-i, buff-i+1, strlen(buff-i)-1 );
*(buff-i)=c;
DelVowel(*(buff-i)); //write c
}
else {
DelVowel(*(buff-i-1)); //del old location
*(buff-i-1)=c;
DelVowel(*(buff-i-1)); //write c
}
post_notice(read_msg[0]);
tone( );
//Pause( );
while(IsMidChar(*buff)!=1)

```

```

        buff++;
    break;
}
//key_in lower than buffer, to chk continue..
DelVowel(*(buff-i)); //del mai-to... before write it by disp_element( )
}
break;
}

```

```

//if MidChar
if(ModeTemp==GET_WORD)
    if(IsNullStr(*(buff+1),Yes)==Yes)
        break;
    if (insert_mode) { //0
        if(type != 'A' && type != 'E') // type == N
            { //1
                j=BuffLen(buff,mask);
                mov_mem(buff, buff+1, j-1);
            } //1
        else
            { //2
                mov_mem(buff,buff+1,strlen(buff)-1);
                i=0; j=0;
                do
                { //3
                    if(*(buff+j)=='\0')
                        break;
                    if(IsMidChar(*(buff+j))==1)
                        { i++; j++; }
                    while(IsMidChar(*(buff+j))!=1)
                        j++;
                }while(*(mask+i)!='\0'); //3
            }
        }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(IsMidChar(*(buff+j))!=1 ) { //ignore vowel of
    //the last column in mask
    if(*(buff+j)=='\0')
        break;
    j++;
}
while(*(buff+j]!='\0') {
    *(buff+j)=' ';
    j++;
}
disp_element(buff,mask);
} //2
} //0...End insert_mode
if( type=='A' && ( IsNullStr(*(buff+1),Yes)==Yes ||
    IsNullStr(*(buff),Yes)==Yes )) { // Do not<3: used entire buff-field
    tone( );
    disp_element(buff,mask);
    c2 = c = FWD;
    break;
}
/// Overwrite Mode
if(*buff !='\0')
    *buff = c;
while(*(buff+1) && IsMidChar(*(buff+1))!=1) {
    mov_mem(buff+2, buff+1, strlen(buff+2));
    if(*(buff+strlen(buff)-1) !='\0')
        *(buff+strlen(buff)-1) = ' ';
}
disp_element(buff,mask);
if(*(buff+1) && *(buff)) {
    buff++;
    mask++;
}

```

```

        prev_col++;
    }
    else {
        if(IsNullStr(*(mask+1),No)==Yes) {
            tone( );
            break;
        }
        c2 = c = FWD;
        prev_col++;
        mask++;
        break;
    }

    while(*mask && *mask!=FIELDCHAR) {
        if(IsMidChar(*mask)==1)
            prev_col++;
        mask++;
    }

    if (!*mask || prev_col>=maxcol) {
        c2 = c = FWD;
        break;
    }
} // which(c)

if (!*mask || prev_col>=maxcol) {
    tone( );
    done = TRUE;
}

if (done){
    if (type == 'D' &&
        c != ESC /*&& validate_date(bfr) != OK */)
        return -1;
    break;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
} // while (TRUE)
if (c != ESC && (type != 'A' && type != 'E')) {
    if (type == 'C') {
        if (*mask++ && *buff == ' ') {
            *buff++ = '0';
            if (*mask++ && *buff == ' ')
                *buff++ = '0';
        }
    }
    if (type == 'Z' || type == 'D')
        right_justify_zero_fill(bfr, mak);
    else if (type != 'B')
        right_justify(bfr, mak);
    cursor(column, prev_row);
    disp_element(bfr, mak);
}
return (c2);
}

int IsNullStr(char buffer, int condit) {
int true;

if(condit==Yes) {
    true=buffer=='\0'?Yes:No;
}
else
    true=buffer!='\0'?Yes:No;
if(true==1) {
    post_notice("หน่วยความจำรองรับไฟล์นี้หมดแล้วครับ...");
    tone( );
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return true;
}

/* ----- test c for an ending keystroke ----- */
endstroke(int c2)
{
    switch (c2)    {
        case '\':
        case -1:
        case -2:
        case -3:
        case -4:
        case -14:
        case -5:
        case -15:
        case -25:
        case -6:
        case -7:
        case -44:
        case 444:
        case -55:
        case '\n':
        case '\':
        case ESC:
        case F1:
        case F2:
        case F3:
        case F4:
        case F5:
        case F6:
        case F7:
        case DEPART: //replace F7 by DEPART

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

case F8:
case F9:
case F10:
case PGUP:
case PGDN:
case HOME:
case END:
case UP:
case DN:
case ADD_PH:
case EDIT_PH:
case FULL_PH:
    return TRUE;
default:
    return FALSE;
}
}
;
/* ----- right justify, space fill ----- */
static void right_justify(char *buff, char *mask) {
int len;
len = BuffLen(buff,mask);
while ((*buff == ' ' || *buff == '0') && len) {
len--;
*buff++ = ' ';
}
if (len)
while (*(buff+(len-1)) == ' ') {
mov_mem(buff, buff+1, len-1);
*buff = ' ';
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

void right_justify2(char *buff, char *mask) {
int len;

len = BuffLen(buff,mask);
while (*buff == ' ' || *buff == '0' && len) {
len--;
*buff++ = ' ';
}
if (len)
while (*(buff+len) == ' ') {
mov_mem(buff, buff+1, len-1);
*buff = ' ';
}
}

/* ----- right justify, zero fill ----- */
static void right_justify_zero_fill(char *buff, char *mask)
{
int len;
if (spaces(buff))
return;
len = BuffLen(buff, mask);
while (*(buff + len - 1) == ' '){
mov_mem(buff, buff + 1, len-1);
*buff = '0';
}
}

/* ----- test for spaces ----- */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int spaces(c)
char *c;
{
    while (*c == ' ')
        c++;
    return !*c;
}

/* ---- display all the fields on a screen ---- */
void tally( )
{
    int *els = elist;
    if(wr_screen==1)
        while (*els)
            put_field(*els++);
}

/* ----- write a data element on the screen ----- */
void put_field(el)
int el;
{
    data_coord(el);
    cursor(prev_col, prev_row);
    disp_element(bf + epos(el, elist), elmask[el - 1]);
}

/* ----- write a data element on the screen_location ----- */
void PutFieldXYTx(x, y, PtrField0, field_no)
int x, y, field_no;
char *PtrField0;
{

```

```

ThaiStr(x*10, y*10, ToFieldNo(PtrField0, field_no));

// while(--field_no > 0)
// while(*PtrField0++); //stop at first byte of the next field
// ThaiStr(x*10, y*10, PtrField0);
}

```

```
char *ToFieldNo(PtrField0, field_no)
```

```
int field_no;
```

```
char *PtrField0;
```

```

{
    while(--field_no > 0)
        while(*PtrField0++); //stop at first byte of the next field
    return(PtrField0);
}

```

```
/* ----- display a data element ----- */
```

```
static void disp_element(b, msk)
```

```
char *b, *msk;
```

```

{
    register int i,j,k=1;
    i=getx( ); j=gety( );

```

```
/*--- setwritemode(XOR_PUT); -----*/
```

```

while (*msk) {
    if(!*b)
        k=0;
    moveto(i,j);
    if(IsMidChar>(*b)==1)
        setwritemode(COPY_PUT);
    bar_del(i,j-2,i+7,j+21,CH_INVER);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        setwriteMode(XOR_PUT);
        setColor(CH_INVER);
        i=rdcw_attr(i,j,0, *mak != FIELDCHAR?*mak : (k==1?*b++:' '));
        if(IsMidChar(*b)==1)
            mak++;
    }
    setwriteMode(COPY_PUT);
    cursor(prev_col,prev_row);
}

/* ----- display insert mode status ----- */
static void insert_status( )
{
    char *say[2]={"เขียนทับ", "เขียนแทรก"};
    m_hide( );
    setwriteMode(COPY_PUT);
    bar_color(maxx/4+maxx/8+20, maxy/10, maxx/2+maxx/22+10,
maxy/6-2,CYAN,0,((insert_mode)?3:4));
    ThaiStrShade(maxx/2-maxx/30+10, maxy/10+2,((insert_mode)?say[1]:say[0]),
    EGA_LIGHTCYAN,BLUE,2,Yes);
    cursor(prev_col,prev_row);
    m_show( );
}

/* ----- write a field character ----- */
static void put_fchar(c)
int c;
{
    //dummy function
    c+=0;
}
int IsMidChar(unsigned char c)

```

```

{ /* Not Upper/Down Char */

if(c==219) { c=5; goto EXIT; } /*Hanarkard + Maito*/
if(c>230 && c<237) { c=4; goto EXIT; }/* 1level top vowel */
if(c>211 && c<216 /*sla-oo..*/ || (c>236 && c<240)
    || c==209/*sla-arm*/) { c=3; goto EXIT; }/* 2level top vowel */
if(c>215 && c<219) { c=2; goto EXIT; }/* low vowel */
if(c==' ') c=1;
else c=1; /* is middle char */
EXIT: return((int)c);
}

int DelVowel(unsigned char c)
{
switch(IsMidChar(c)) {
case 1: return 0;
default: setwriteMode(XOR_PUT);
        setColor(CH_INVER);
        rdw_attrib(getx( ),0, c);
        setwriteMode(COPY_PUT);
return 1;
}
}

int BuffLen(char *buff, char *mask)
{
int i=0,j=0,k=0;
while(*(mask+i)!=FIELDCHAR) {
if(IsMidChar(*(buff+j))==1) { //Mid-character
i++;
k++;
}
}
}

```

```

else
    k++;

    j++;
}
return(k);
}

int MskLen(char *mask)
{
int i=0;
    while(*(mask+i++));
return(i);
}

void tone( )
{
    sound(494);
    Delay(0.01);
    nosound( );
    sound(330);
    Delay(0.007);
    nosound( );
    sound(262);
    Delay(0.001);
    nosound( );
}

void saymode(int mode)
{
    say_mode=mode;
    bar_color(maxx/4, maxy/10, maxx/4+maxx/8,maxy/6-2,CYAN,0,(mode==ENG?4:3));
    ThaiStrShade(maxx/4+maxx/16,maxy/10+2,(mode==ENG?"ENG":"ไทย"),

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    EGA_LIGHTCYAN,BLUE,2,Yes);
}

void say_command(char *command) {
    setlinestyle(0, 1, 1);
    bar_color(6,maxy-36,maxx-6,maxy-7,CYAN,0,3);
    setcolor(BLACK);
    ThaiStrShade(maxx/2, maxy-maxy/17-6, command,YELLOW,BLACK,2,Yes);
}

/* ---- display all phasal_verbs depending on such main_word ----- */
char phaso_no[MAX_PH_LIST][6];

/* ----- Check head word for existing ---- */
int ChkMainWord(char *head_name)
{
    int (*validfunc)( );
    int isvalid;
    int field_ctr, field_ptr=0, exitcode, al, i, j,maxphase;
    char *bfptr, *bfptr2;

    TakeOverDbFile( TellFileNo(head_name,0) ); //0==dicwords group
    bfptr = bf + epos(elist[field_ptr], elist);
    validfunc = sb[field_ptr].edits;
    for(i=0; i<strlen(head_name); i++)
        *(bfptr+i)=*(head_name+i);

    wr_screen=0; //ignore display_template
    (*validfunc)(bfptr);
    setviewport(view2.left,view2.top,view2.right,view2.bottom,1);
    wr_screen=1;
    bfptr2 = ToFieldNo(bfptr,2); // to the start of field#2
    while(*bfptr2!=\0) //goto the end of field#2

```



```

    bfptr2++;
bfptr2--;
for(i=0; (*bfptr2 && *bfptr2--==' '); i++);
if(VocMode==WR_PH && i<List_Limit) {
    Set_Graphic_Cursor(GLOVE);
    tone( );
    ModeTemp=FULL_PH;
    if(ChkYN("<เต็ม> , ท่านจะแก้ไขคำวลีเดิมหรือไม่ <Y/N> ? :",Yes)==No) {
        nlock_kb=Yes;
        wr_screen=0;
        return ModeTemp=GET_WORD;
    }
}

if(!spaces(bfptr=ToFieldNo(bfptr,2))) //data no_empty ?
{
    setviewport(view2.left,view2.top,view2.right,view2.bottom,1);
    while(*bfptr!='\0') //goto the end of field
        bfptr++;
    bfptr--;
    while(*bfptr && *bfptr==' ') //found space, walk-back..
        bfptr--;
j=0;
while(*bfptr) {
    for(i=0; i<5; i++)
        phase_no[j][i]='\0';
    i=5;
    while(*bfptr && *bfptr!=' ') {
        phase_no[j][5] = '\0'; //Compiler didn't fill '\0'
        phase_no[j][--i] = *bfptr;
        bfptr--;
    }
    while(*bfptr && *bfptr==' ')

```

```

bfptr--;
    if(j>0) {
        head_name = &phase_no[j][0];
    }
    maxphase=j++;
}
LookForPhase(maxphase,0); //0=dic //1=study
// SH_CAPS=1;
nolock_kb=No;
return ModeTemp;
}
else { //--Phasal list was emptied
if(VocMode==DIC_PH) {
    m_hide( );
    WindowExpand(maxx/5,maxy/2-28,maxx-maxx/5,maxy-maxy/3-13,
    EGA_LIGHTRED,YELLOW,4);
    ThaiStrShade(maxx/2+30,maxy/2+7,"ไม่พบในแฟ้มข้อมูล...",
    YELLOW,RED,2,Yes); //"This word not found"
    sh_draw1(maxx/5+30,maxy/2-22,"flag.drw");
    Set_Graphic_Cursor(GLOVE);
    m_show( );
    Pause( );
    m_hide( );
    WindowHide(maxx/5,maxy/2-28,maxx-maxx/5,maxy-maxy/3-13,
    CYAN,YELLOW,1);
    clear_record( );
    nolock_kb=Yes;
    ModeTemp=GET_WORD;
    return No;
}

Set_Graphic_Cursor(GLOVE);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if((i=ChkYN("จะเพิ่มคำนี้หรือไม่ (Y)es/(N)o/(C)ancel ? :
",Yes))!=Yes) {
    clear_record( );
    nlock_kb=Yes;
    ModeTemp=GET_WORD;
    return No;
}
LookForPhase(NEW_PH,0);
nlock_kb=No;
ModeTemp=NEW_PH;
return NEW_PH;
}
}

void WindowExpand(l,t,r,b,bkshade,boxcolor,boxmode)
int l,t,r,b,bkshade,boxcolor,boxmode;
{
    register int i,j;

    setwriteMode(COPY_PUT);
    setLineStyle(0,0,1);
    j=(r+l)/2;
    i=(b+t)/2;

    for(;;) {
        j>1+(maxx/7)?j--=(r-1)/8:; // j--=maxx/7:;
        i>t+(maxy/25)?i--=(b-t)/8:;
        bar_color(j,i,((l+r)/2)+(((l+r)/2)-j),(b+t)/2+(((b+t)/2)-i),
            bkshade,boxcolor,boxmode);
        if(j==1 && i==t)
            break;
        Delay(0.001);
    }
}

```

```

}
}

void WindowHide(l,t,r,b,bkshade,boxcolor,boxmode)
int l,t,r,b,bkshade,boxcolor,boxmode;
{
register int i=t,j=l;
setwritemode(COPY_PUT);
setlinestyle(0,0,1);
for(;;) {
j=j<((r+1)/2)-((r-1)/8) ? j+==(r-1)/8:(r+1)/2;
i=i<((b+t)/2)-((b-t)/8) ? i+==(b-t)/8:(b+t)/2;
bar_color(l,t,r,i,bkshade,0,0);
bar_color(l,t,j,b,bkshade,0,0);
bar_color(r-(j-1),t,r,b,bkshade,0,0);
bar_color(l,b-(i-t),r,b,bkshade,0,0);
if(boxmode!=0 && i<(b+t)/2) {
setcolor(boxcolor);
rectangle(j+1,i+1,r-(j-1)-1,b-(i-t)-1);
}
if(j===(r+1)/2 && i===(b+t)/2)
break;
Delay(0.0001);
}.
}

```

```

void Shadow(l,t,r,b,bb,rr)
register int t,l,r,b,bb,rr;
{
setcolor(BLACK);
setlinestyle(4,21845,1);
for( t+=<b; ) {

```

```

if(t<=bb+1) {
    if(t%2==0)
line(rr,t,r-1,t);
    else
        line(rr+1,t,r,t);
}
else {
    if(t%2==0)
        line(l,t,r-1,t);
    else
        line(l+1,t,r,t);
}
}
}

extern void *image2;
extern unsigned numbytes;
int ChkYN(ask,winhide)
char *ask;
int winhide;
{
int c,y,x,BkTemp;
register int i,j,old_mode_temp=ModeTemp;
ModeTemp=-5; //any number that !=GET_WORD and !=EDIT_PH
ToView(2);
x=maxx/2-(8*QtyMidStr(ask)/2)-35;
y=maxy/3-20;
if(m_events->yaxis>=view2.top+y-25 &&
m_events->yaxis<=view2.top+y+(maxy/5)+(maxy/22) &&
m_events->xaxis>=view2.left+x-25 &&
m_events->xaxis<= view2.left+maxx-x+(maxx/25))
    m_hide( );
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(winhide!=0) {
    BkTemp=getpixel(x,y);
    sv_image1(x,y, maxx-x+(maxx/25),y+(maxy/5));
    numbytes = image_size(x+(maxx/25), (y+maxy/5), maxx-x+(maxx/25),
        y+(maxy/5)+(maxy/22));
    image2 = (void *)malloc(numbytes);
    get_image(x+(maxx/25), y+(maxy/5), maxx-x+(maxx/25),
        y+(maxy/5)+(maxy/22), image2);
}
else
    sv_image1(x,y, maxx-x+(maxx/25),y+(maxy/5)+(maxy/22));
WindowExpand(x,y,maxx-x,y+maxy/5,LIGHTMAGENTA,YELLOW,1);
Shadow(x+(maxx/25),y+(maxy/22),(maxx-x)+(maxx/25),
    y+(maxy/5)+(maxy/22),y+(maxy/5-2),maxx-x);
for(i=0; i<3; i++)
    bar_color(maxx/4+4+(i*maxx/6), y+maxy/8-7,
        maxx/4-3+((1+i)*maxx/6),y+maxy/5-12,
        LIGHTRED,0,4); //if change color, must change color in bar_del( )
    //which under ans-respond module
setcolor(WHITE);
setlinestyle(0,0,3);
for(i=0;i<3;i++) {
    if(i<2)
        line(maxx/4+(i==0?15:12)+(i*maxx/6), y+maxy/8,maxx/4+(i==0?45:
            32)+(i*maxx/6), y+maxy/8+12);
        line(maxx/4+(i!=1?15:12)+(i*maxx/6), y+maxy/8+12,maxx/4+(i!=1?45:
            32)+(i*maxx/6), y+maxy/8);
    if(i>1)
        line(maxx/4+12+(i*maxx/6), y+maxy/8+9,maxx/4+15+(i*maxx/6),
            y+maxy/8+12);
}
setcolor(BLACK);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ThaiStrShade(x+10, y+(maxy/16)-12, ask,WHITE,BLACK,2,No);
ThaiStrShade(maxx/4+47, y+maxy/8-2, "ไม่",WHITE,BLACK,2,No); //No
ThaiStrShade(maxx/4+38+(maxx/6), y+maxy/8-2, "ยกเลิก",
WHITE,BLACK,2,No); //cancel
ThaiStrShade(maxx/4+52+(2*maxx/6), y+maxy/8-2, "ใช่",
WHITE,BLACK,2,No); //yes
do {
    say_mode=ENG;
    if(say_mode==ENG) say_mode=THAI;
    m_show( );
    c=(c=GetKeyMou(maxx-x-55,y+(maxy/16)-9))>=0?tolower(c):c;
    settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
    settextjustify(LEFT_TEXT, CENTER_TEXT);
    setcolor(BLACK);
    m_hide( );
    switch(c) {
    case -99:
    case 'n':
        outtextxy(maxx-x-55,y+maxy/16,"NO");
        VocBarAtt(8);
        ToView(2);
        c=No;
        break;
    case ESC:
    case -98:
    case 'c':
        outtextxy(maxx-x-55,y+maxy/16,"CANCEL");
        VocBarAtt(9);
        ToView(2);
        c=Cancel;
        break;
    case -97:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 'y':
    outtextxy(maxx-x-55,y+maxy/16,"YES");
    VocBarAtt(10);
    ToView(2);
    c=Yes;
    break;
default :
    outtextxy(maxx-x-55,y+maxy/16,". . .");
    c=3;
    break;
}
xold=m_events->xaxis;
yold=m_events->yaxis;
Delay(0.09);
m_hide( );
bar_del(maxx-x-55,y+5, maxx-x-4,y+40,LIGHTMAGENTA);
}while(c==3);
if(m_events->yaxis>=view2.top+y-25 &&
m_events->yaxis<=view2.top+y+(maxy/5)+(maxy/22) &&
m_events->xaxis>=view2.left+x-25 &&
m_events->xaxis<= view2.left+maxx-x+(maxx/25))
    m_hide( );
if(winhide!=0) {
    putimage(x+(maxx/25), (y+maxy/5), image2, COPY_PUT);
    free(image2);
    WindowHide(x,y,maxx-x+(maxx/25),y+(maxy/5), BkTemp,YELLOW,1);
    m_hide( );
    sh_image1(x,y);
}
else
    sh_image1(x,y);
//check near got_word_box

```



```

if((m_events->xaxis>box_width+mx5-2 && m_events->xaxis < box_width+mx5+86 ) &&
    ( m_events->yaxis>bar_height+mxy/3+4 && m_events->yaxis < bar_height+mxy/3+49))
    m_moveto(box_width+maxx/4,bar_height+mxy/3+52);
m_moveto(xold,yold);
m_show( );
ModeTemp=old_mode_temp;
return c;
}

void ClearViewPort( ) {
bar_del(0,0,maxx,mxy,bkcolor);
}

void ShowView2(void) {
ToView(2);
sh_draw1(0,1,"laitail.drw");
sh_draw1(maxx-61,1,"laitair.drw");
bar_color(mx4,mxy/50,mx2+mx4, mxy/12+2,EGA_LIGHTBLUE,RED,2);
ThaiStrShade(mx2+5, mxy/21-6, "ตำนานกริยาวลี และตะแสง",EGA_LIGHTCYAN,BLACK,2,Yes);
}

void ClsStatusKey(void) {
bar_del(mx5,mxy/12+5,maxx-maxx/4+5, mxy/5-1,LIGHTGRAY);
}

void ToView(int Viewport) {
switch(Viewport) {
case 1:
setviewport(view1.left,view1.top,view1.right,view1.bottom,1);
maxx=view1.right-view1.left;
mxy=view1.bottom-view1.top;

```

```

        break;
    case 2:
        setviewport(view2.left,view2.top,view2.right,view2.bottom,1);
        maxx=view2.right-view2.left;
        maxy=view2.bottom-view2.top;
        break;
    case 3:
        setviewport(view3.left,view3.top,view3.right-23,view3.bottom-22,1);
        maxx=view3.right-23-view3.left;
        maxy=view3.bottom-22-view3.top;
        break;
    case 4:
        setviewport(view4.left, view4.top, view4.right, view4.bottom,1);
        maxx=view4.right-view4.left;
        maxy=view4.bottom-view4.top;
        break;
}
}

void Pause(void) {
    m_events->button=2;
    while(kbhit( )) getch( );
    while(m_events->button!=1 && !kbhit( ));
    while(kbhit( )) getch( );
}

void HelpAbout(int AH_Mode) {
    int mx,my,i,j,k,c,mainabout;
    char *about[ ] = {
        "ปริญญานิพนธ์", //0

```

"จำนวนกริยาวิเศษณ์และสระแดง",
 "Incomplete-Version๑.๐",
 "๗ พฤศจิกายน ๒๕๑๗",
 "คณะครุศาสตร์อุตสาหกรรม",
 "สจล.",
 "ที่ปรึกษา",
 "ผู้ประพันธ์",
 "", // "ออก",
 "อ.วราวิทย์ สมหา", //9
 "อ.ถันติ ดันตระกุล",
 "อ.สุชิน อาษาหาญ",
 "สาขา",
 "อิเล็กทรอนิกส์และคอมพิวเตอร์",
 "GRAPHICS PRESENTATION", //14
 "เชิงชาย ไชยชิน",
 "DATABASE SYSTEM DESIGN",
 "อนุชชรุ อ่อนคง",
 "THAI DRIVER ALGORITHM",
 "วัชร รัตนานนท์",
 "TOPDOWN DESIGN & FONTS",
 "ชุกิจ ศรีใหม่"
 };

```

char *HMsg[ ] = {
    "HELP-เมนู",//0    ///main menu
    "<Tab>เลือกเมนู",
    "<ด้าน>แสดงที่มาโปรแกรม",
    "<ศัพท์>เมนูศัพท์",
    "<เพิ่ม>ข้อมูลเพิ่มเติมที่เสียบ",
    "<ไม้>ปรับความเร็ว",
    "<เพื่อ>DOS ชั่วคราว",
    "<ESC:ประตู>จบโปรแกรม",
  
```

"HELP-เมนูศัพท์", //8 ///vocab menu

"<F1:HELP>เมนูช่วยเหลือ",

"<ถัด>แสดงที่มาโปรแกรม",

"<ค้นหา>เปิดคำศัพท์",

"<เรียนรู้>วิธีใช้คำศัพท์",

"<เพิ่ม>คำศัพท์ด้วยตนเอง",

"<ทดสอบ>คุณคนเก่ง",

"<ESC:ประตู>ออกเมนู", //ESC:out

"HELP-คำศัพท์หลัก", //16 ///GET_WORD-mode DIC and mode WR_PH

"<F1:HELP>เมนูช่วยเหลือ",

"<ถัด>แสดงที่มาโปรแกรม",

"<F4:OK.>งานคำศัพท์",

"<ถัด>ทังไป ทิมใหม่",

"<มีถัด>ความเร็วคุณขอม",

"<เพื่อ>DOS ชั่วครว",

"<ESC:ประตู>ออกเมนู", //ESC:out

"HELP-คูศัพท์วลี", //24 ///Phase list mode DIC

"<F1:HELP>เมนูช่วยเหลือ",

"<ถัด>แสดงที่มาโปรแกรม",

"<OK.>ตกลงใจ",

"< >ถัดขึ้นเลือก",

"< >ถัดลงเลือก",

"<เพื่อ>DOS ชั่วครว",

"<ESC:มี>ออกเมนู", //esc

"HELP-แก้ศัพท์วลี", //32 ///Phase list mode WR_PH

"<F1:HELP>เมนูช่วยเหลือ",

"<ถัด>แสดงที่มาโปรแกรม",

"<OK.>ตกลงใจ",

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"> >เลื่อนขึ้นเลือก",
 "< >เลื่อนลงเลือก",
 "<เพื่อ>DOS ชั่วคราว",
 "<ESC:มือ>ออกเมนู", //esc

"HELP-พจนานุกรมวาลี", //40 ///Show DIC
 "<F1:HELP>เมนูช่วยเหลือ",
 "<ฉับ>แสดงที่มาโปรแกรม",
 "<อักษร>เปลี่ยนแบบอักษร",
 "< >ขึ้นคู่มือข้อมูล",
 "< < >ลงคู่มือข้อมูล",
 "<ผู้เอกสาร>บันทึกคำศัพท์",
 "<ESC:มือ>ออกเมนู", //esc

"HELP-แก้,เพิ่มวาลี", //48 ///EDIT_PH VocMode=WR_PH
 "<F1:HELP>เมนูช่วยเหลือ",
 "< >เลื่อนขึ้น",
 "<OK>เมื่อท่านพอใจ",
 "<F10:ธง>ไทย/ENGLISH",
 "<INS:หนังสือ>พิมพ์ทับ/แทรก",
 "<เพื่อ>DOS ชั่วคราว",
 "<F2:คิตส์>บันทึกข้อมูล",

"HELP-เมาส์", //56 ///mouse
 "ปรับความเร็วเมาส์",
 " เพิ่มความเร็ว",
 " ลดความเร็ว ",
 "=>",

"HELP-ฉับ",//61 ///my project
 "กดปุ่ม",
 "F2:ที่ปรึกษา",

"F3 ผู้ประพันธ์"

};

```
char *mousefast[] = {
    "เมาส์-ถมกรค",
    "ปรับความเร็ว",
    "เพิ่ม",
    "ลด"
};
```

```
char *datfile[] = {
    "",
    "",
    "",
    "=>",
    ""
};
```

```
HELPABOUT=1; //define area-pointer of mouse in GetKeyMou( )
```

```
getviewsettings(&vp);
```

```
mx=maxx;
```

```
my=maxy;
```

```
ToView(4);
```

```
m_hide( );
```

```
sv_image1(0,0,maxx,maxy);
```

```
setcolor(WHITE);
```

```
if(AH_Mode==1 || AH_Mode==F1) {
```

```
    switch(ModeTemp) {
```

```
        case GET_MAIN:
```

```
            i=0;
```

```
            j=7;
```

```
            break;
```

```

    case GET_VOC:
i=8;
j=15;
break;

    case GET_WORD:
i=16;
j=23;
break;

    case GET_PH:
if(VocMode==WR_PH) {
i=32;
j=39;
}
else {
i=24;
j=31;
}
break;
case DIC_PH:
i=40;
j=47;
break;
case EDIT_PH:
i=48; //VocMode=WR_PH
j=55;
break;
    'default:
i=0;
j=7;
break;
}
WinButton(HMsg[i],"*");

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Shadow(20,20,maxx,maxy,maxy-20-2,maxx-20-1);
setcolor(WHITE);
for(k=0; (++i<=j);) {
    switch(i) {
        case 28:
        case 36:
        case 44:
        case 50:
        case 58:
            showchar(33, 32+(i==50?1:3)*22,(unsigned char)253);
            break;
        case 29:
        case 37:
        case 45:
        case 59:
            showchar(33, 32+4*22,(unsigned char)252);
            default:
            break;
        }
        ThaiStr(20,32+(k++*22), HMsg[i]);
    }
    m_show( );
    while((c=GetKeyMou(-50,-50))!=-93 && c!=ESC);
    'VocBarAtt(11);
}

else if(AH_Mode===-2) { ////////// ABOUT
WinButton(about[0],"***");
Shadow(20,20,maxx,maxy,maxy-20-2,maxx-20-1);
for(i=1; i<9; i++) {
    setcolor(i<6?YELLOW:RED);
    ThaiStrShade(maxx/2-10,(i<6?(i*20)+15:(i*27)-23),about[i],
    EGA_LIGHTCYAN,BLUE,2,Yes);
}
}

```



```

}
m_show( );
do {
mainabout=Yes;
switch(k=GetKeyMou(-50,-50)) {
case -1:
case F1:
NumChoose=!NumChoose;
VocBarAtt(0);
WinButton(HMsg[61],"*");
for(i=0; i<3; i++) { //9-13
setcolor(WHITE);
// ThaiStrShade(maxx/2-10,(i*25)+maxy/5 ,HMsg[i+62],Yes);
ThaiStrShade(maxx/2-10,(i*25)+maxy/5 ,HMsg[i+62],
EGA_LIGHTCYAN,BLUE,2,Yes);
}
break;
case F2:
case -95:
VocBarAtt(13);
WinButton(about[6],"*");
for(i=0; i<5; i++) { //9-13
setcolor(WHITE);
ThaiStrShade(maxx/2-10,(i*25)+maxy/5 ,about[i+9],EGA_LIGHTCYAN,
BLUE,2,Yes);
}
break;
case F3:
case -94:
VocBarAtt(12);
WinButton(about[7],"*");
for(i=0,j=0; i<8; i++,j=(i%2==0)?0:1) { //12-19

```

```

        ThaiStrShade(maxx/2-10,(i*18)+35 ,about[i+14],
        j==0?YELLOW:WHITE,BLUE,2,Yes);
    }
    break;
    case ESC:
    case -93:
        VocBarAtt(11);
        break;
    default:
        k=0;
        mainabout=No;
        break;
}
if(k!=-93 && k!=ESC && k!=0) {
    m_show( );
    while((c=GetKeyMou(-50,-50))!=-93 && c!=ESC );
    VocBarAtt(11);
    k=0;
    m_hide( );
    WinButton(about[0],"***");
    for(i=1; i<9; i++) {
        ThaiStrShade(maxx/2-10,(i<6?(i*20)+15:(i*27)-23),about[i],
        EGA_LIGHTCYAN,BLUE,2,Yes);
    }
    m_show( );
    mainabout=No;
}
}while(k!=-93 && k!=ESC && k==0 && mainabout==No);
    VocBarAtt(11);
}
else if(AH_Mode==F6 || AH_Mode==15) { //MOUSE-VELOCITY
    WinButton(mousefast[0],"** *");

```

```

Shadow(20,20,maxx,maxy,maxy-20-2,maxx-20-1);
for(i=2; i<5; i++) {
    setcolor(i<3?YELLOW:RED);
    ThaiStrShade(maxx/2-10,(i<3?(i*20)+15;((i+2)*27)-23),mousefast[i-1],
    EGA_LIGHTCYAN,BLUE,2,Yes);
}

setcolor(YELLOW);
ThaiStrShade(maxx/8,5*27-25, HMsg[60], EGA_LIGHTCYAN,
BLUE,2,No); //==>'
for(i=0; i<=HorFast; i++) {
    bar_del(maxx-maxx/4+3,maxy/3,maxx-maxx/6,maxy-(maxy/9),
    EGA_LIGHTBLUE);
    bar_color(maxx-maxx/4,maxy-(maxy/9)-(120-(3*i))-4,
    maxx-maxx/6,maxy-(maxy/9),WHITE,RED,1);
}
m_show();
do {
    switch(k=GetKeyMou(-50,-50)) {
case -1:
    case F1:
        NumChoose=!NumChoose;
        VocBarAtt(0);
        WinButton(HMsg[56],"*");
        setcolor(WHITE);
        for(i=0; i<3; i++) {
            ThaiStrShade(maxx/2-10,(i*25)+maxy/5 ,HMsg[i+57],
            EGA_LIGHTCYAN,BLUE,2,Yes);
            if(i>0)
                showchar(maxx/4, (i*25)+maxy/5,(unsigned char)254-i);
        }
        m_show();
    while((c=GetKeyMou(-50,-50))!= -93 && c!=ESC );

```

```

VocBarAtt(11);
m_hide( );
WinButton(mousefast[0], "*** *");
for(i=2; i<5; i++) {
    setcolor(i<3?YELLOW:RED);
    ThaiStrShade(maxx/2-10,(i<3?(i*20)+15:(i+2)*27)-23,mousefast[i-1],
        EGA_LIGHTCYAN,BLUE,2,Yes);
}
setcolor(YELLOW);
ThaiStrShade(maxx/8,5*27-25, datfile[3],
    EGA_LIGHTCYAN,BLUE,2,No);
for(i=0; i<=HorFast; i++) {
    bar_del(maxx-maxx/4+3,maxy/3,maxx-maxx/6,maxy-(maxy/9),EGA_LIGHTBLUE);
    bar_color(maxx-maxx/4,maxy-(maxy/9)-(120-(3*i))-4,maxx-maxx/6,maxy-(maxy/9),WHITE,RED,1);
}
m_show( );
break;
case -92:
case -95:
case UP:
case DN:
    ToView(4);
    bar_del(10,5*27-23,maxx/3-18,maxy-maxy/3,EGA_LIGHTBLUE);
    setcolor(YELLOW);
    if(k== -92 || k==UP) {
        VocBarAtt(14);
        HorFast=--HorFast<1?1:HorFast;
    }
else {
    VocBarAtt(13);
    HorFast=++HorFast>40?40:HorFast;
}

```

```

VerFast=HorFast*2;

setwritemode(COPY_PUT);
setcolor(YELLOW);
ThaiStrShade(maxx/8,(k==92||k==UP?5*27-25:5*27), datfile[3],
             EGA_LIGHTCYAN,BLUE,2,No); //"==>"
m_move_ratio(HorFast,VerFast);
m_hide( );
bar_del(maxx-maxx/4+3,maxy/3,maxx-maxx/6,maxy-(maxy/9),EGA_LIGHTBLUE);
bar_color(maxx-maxx/4,maxy-(maxy/9)-(120-(3*HorFast))-4,maxx-maxx/6,
maxy-(maxy/9),WHITE,RED,1);
m_show( );
break;
default:
break;
}
} while(k!=93 && k!=ESC);
VocBarAtt(11);
}
ToView(4);
m_hide( );
WinSnow(0,0,maxx-20,maxy-20,4,EGA_LIGHTBLUE,WHITE,1);
sh_image1(0,0);
m_show( );
setviewport(vp.left,vp.top,vp.right,vp.bottom,1);
maxx=mx;
maxy=my;
HELPABOUT=0;
}

void WinButton(char *title, char *show_button) { //! window & some buttons
int ij;

```

```

WinSmall(title);
for(j=0,i=8-strlen(show_button); (++i<8);) {
    if(*(show_button+j)!=' ')
        bar_color(maxx/3-14,(i*27)-28,maxx-maxx/3-9,(i*27),
        CYAN,0,4); //creat button
    j++;
}
}

```

```

void WinSmall(char *title) { //1 window & 1 button
int i;

```

```

    ToView(4);
    m_hide( );
    WinSnow(0,0,maxx-20-1,maxy-20,4,EGA_LIGHTBLUE,WHITE,1);
    bar_color(maxx/5-10,6,maxx-maxx/5-10, 31,CYAN,0,2); //top-button
    bar_color(maxx/3-14,188,maxx-maxx/3-9,216,CYAN,0,4); //bottom-button
    for(i=0; i<2; i++)
        ThaiStrShade(maxx/2-10,(i<1?10:193), (i<1?title:"OKAY"),
        WHITE,BLACK,2,Yes);
}

```

```

void BarAttrib(int l, register int t, int r,
    register int b, int color, int WRmode) {

```

```

int l2,t2,r2,b2; //TEMP
    m_hide( );
    l+=3;
    t+=2;
    r-=2;
    b-=2;
    l2=l,t2=t,r2=r,b2=b;

```

```

/*-----*/
/*          PH_DAMAG.I          */
/*          FUNCTION : Check file damage          */
/*-----*/

void Status(int code) {
    int c;
    if(code!=0) {
        printf("\nERROR in executable file..");
        printf("\n\nDo you want to continue <Y/N> ? : ");
        do {
            switch(c=toupper(getch( ))) {
                case 'N':
                    cleardevice( );
                    closegraph( );
                    exit(-1);
                default: break;
            }
        }while(c!='Y');
        printf("\nHit a key to continue...");
    }
}

void IsDamage(void) {
    char pathnam[145], curdir[130]; //variable number
    int i;
    char *args[ ] = {
        "ph_vrs2.pwa",
        "cd-ong15",
        , NULL
    };
    getcwd(curdir,130);
    i=0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

do {
    if(curdir[i]==(char)\' && curdir[i+1]==\'0')
        curdir[i]=\'0\';
    i++;
} while(curdir[i]!=\'0\');
sprintf(pathnam,"%s\\%s",curdir,args[0]);
Status(spawnl(P_WAIT,pathnam,args[0],args[1],NULL));
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

/*-----*/
/*      datafile.c      */
/*      FUNCTION : open/close database file      */
/*-----*/
#define flocat(r,l) ((long)(sizeof(FHEADER)+(((r)-1)*(l))))

static int handle [MXFILS];
FHEADER fh [MXFILS];

/* ----- create a file ----- */
void file_create(name, len)
char *name;
int len;
{
    int fp;
    FHEADER hd;
#if COMPILER == MICROSOFT
    extern int _iomode;
    _iomode = 0x8000;
#endif
#if COMPILER == LATTICE
    extern int _iomode;
    _iomode = 0x8000;
#endif

    unlink(name);
    fp = creat(name,CMODE); /*S_IWRITE*/
    close(fp);
    fp = open(name, OPENMODE); /*O_RDWR+O_BINARY); */
    hd.next_record = 1;
    hd.first_record = 0;
    hd.record_length = len;
    write(fp, (char *) &hd, sizeof hd);
}

```

```

        close(fp);
    }

/* ----- open a file ----- */
int file_open(name)
char *name;
{
    int fp;
#ifdef COMPILER == MICROSOFT
        extern int _iomode;
        _iomode = 0x8000;
#endif
#ifdef COMPILER == LATTICE
        extern int _iomode;
        _iomode = 0x8000;
#endif
    for (fp = 0; fp < MXFILS; fp++)
        if (handle [fp] == 0)
            break;
    if (fp == MXFILS)
        return ERROR;
    if ((handle [fp] = open(name, OPENMODE)) == ERROR)
        return ERROR;
    lseek(handle [fp], 0L, 0);
    read(handle [fp], (char *) &fh [fp], sizeof(FHEADER));
    return fp;
}

/* ----- close a file ----- */
void file_close(fp)
int fp;

```

```

{
    lseek(handle [fp], 0L, 0);
    write(handle [fp], (char *) &fh [fp], sizeof(FHEADER));
    close(handle [fp]);
    handle [fp] = 0;
}

/* ----- create a new record ----- */
RPTR new_record(fp, bf)
int fp;
char *bf;
{
    RPTR rcdno;
    extern char *malloc( );
    FHEADER *c;
    extern int free( );
    if (fh [fp].first_record) {
        rcdno = fh [fp].first_record;
        if ((c = (FHEADER *)
            malloc(fh [fp].record_length)) == NULL) {
            errno = D_OM;
            dberror( );
        }
        get_record(fp, rcdno, c);
        fh [fp].first_record = c->next_record;
        free(c);
    }
    else
        rcdno = fh [fp].next_record++;
    put_record(fp, rcdno, bf);
    return rcdno;
}

```

```

/* ----- retrieve a record ----- */
int get_record(fp, rcdno, bf)
int fp;
RPTR rcdno;
char *bf;
{
    if (rcdno >= fh [fp].next_record)
        return ERROR;

    lseek(handle [fp],
           flocate(rcdno, fh [fp].record_length), 0);
    read(handle [fp], bf, fh [fp].record_length);
    return OK;
}

/* ----- rewrite a record ----- */
int put_record(fp, rcdno, bf)
int fp;
RPTR rcdno;
char *bf;
{
    if (rcdno > fh [fp].next_record)
        return ERROR;

    lseek(handle [fp],
           flocate(rcdno, fh [fp].record_length), 0);
    write(handle [fp], bf, fh [fp].record_length);
    return OK;
}

/* ----- delete a record ----- */
int delete_record(fp, rcdno)
int fp;
RPTR rcdno;
{

```

```

FHEADER *bf;
extern char *malloc( );

if (redno > fh [fp].next_record)
    return ERROR;
if ((bf = (FHEADER *)
        malloc(fh [fp].record_length)) == NULL) {
    errno = D_OM;
    dberror( );
}
set_mem(bf, fh [fp].record_length, '\0');
bf->next_record = fh [fp].first_record;
bf->first_record = -1;
fh [fp].first_record = redno;
put_record(fp, redno, bf);
free(bf);
return OK;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้.

```

/*-----*/
/*      PH_DELAY.I      */
/*  FUNCTION : Check time delay beyond delay( ); */
/*      Hear...millisecond      */
/*-----*/

```

```
void DelayChkKey(float TimSet)
```

```

{
    float start, second;
    int Exit=0;
    start = clock( );
    do {
        second=(clock( )-start) / CLK_TCK;
        if( second > TimSet )
            Exit=1;
    }while(kbhit( )==0 && Exit==0); /*NO*/
}

```

```
void Delay(float TimSet)
```

```

{
    float start, second;
    start = clock( );
    while(((clock( )-start) / CLK_TCK) < TimSet);
}

```

```

/*-----*/
/*      PH_DRW.I      */
/* FUNCTION : show or hide any icon */
/*      */
/*-----*/

void *file1;

void *image;

void *rd_fileimage(char *filename);

void wr_fileimage(void *image, unsigned size, char *filenam);

void sh_draw1(int l, int t, char *filenam);

void sv_draw1(int l, int t, int r, int b, char *filenam);

void sv_pic1(int l, int t, int r, int b, char *filenam);

void sv_pic1(int l, int t, int r, int b, char *filenam)
{
    unsigned size1 = 0;
    unsigned *size2;

    *size2=size1;
    *size2 = imagesize( l, t, r, b );
    image = (void *)malloc( *size2 );
    getimage( l, t, r, b, image );
    putimage( l, t, image, COPY_PUT );
    wr_fileimage( image, size1, filenam );
    file1 = rd_fileimage( filenam );
    putimage( l,t, file1, COPY_PUT );
    free(file1);
}

void sv_draw1(int l, int t, int r, int b, char *filenam)
{
    unsigned size1 = 0;
    unsigned *size2;

```

```

size2=&size1;
*size2 = imagesize( l, t, r, b );
image = (void *)malloc( *size2 );
getimage( l, t, r, b, image );
putimage( l, t, image, COPY_PUT );
wr_fileimage( image, size1, filename );
file1 = rd_fileimage( filename );
putimage( l,t, file1, COPY_PUT );
free(file1);
}

```

```

void sh_draw1(int l, int t, char *filename)
{
file1 = rd_fileimage( filename );
putimage( l,t, file1, COPY_PUT );
free(file1);
}

```

```

void wr_fileimage(void *image, unsigned size, char *filename)
{
FILE *f1;
if( (f1= fopen( filename, "wb" )) ==NULL)
{
outtextxy(10,50,"Cannot open wb_file");
getch( );
}
fwrite( image, size, 1, f1 );
fflush( f1 );
fclose( f1 );
}

```

```

void *rd_fileimage(char *filename)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

unsigned i;
unsigned xsize, ysize, size;
void *tempimage;
FILE *f1;
if( (f1= fopen( filename, "rb" )) ==NULL)
{
gotoxy(1,1);
printf("Cannot open rb_file");
getch( );
}
xsize = fgetc( f1 ) | ( fgetc( f1 ) << 8 );
ysize = fgetc( f1 ) | ( fgetc( f1 ) << 8 );
size = imagesize( 0, 0, xsize, ysize );
tempimage = (void *)malloc( size );
rewind( f1 );
fread( tempimage, size, 1, f1 );
fclose( f1 );
return( tempimage );
}

```



```

/*-----*/
/*          PH_IMAGE.I          */
/*  FUNCTION : save or restore image-area  */
/*-----*/

```

```
void *image1, *image2, *image_voc;
```

```
unsigned numbytes;
```

```
void sv_image1(int l, int t, int r, int b)
```

```

{
numbytes = imagesize(l, t, r, b);
if((image1 = (void *)malloc(numbytes)) == (void *)NULL)
{
printf("Not enough memory \n");
getch();
}
else getimage(l, t, r, b, image1);
}

```

```
void sh_image1(int l, int t)
```

```

{
putimage(l, t, image1, COPY_PUT);
free(image1);
}

```

```
void sv_image_voc(int l, int t, int r, int b)
```

```

{
numbytes = imagesize(l, t, r, b);
if((image_voc = (void *)malloc(numbytes)) == (void *)NULL)
{
printf("Not enough memory \n");
getch();
}
}

```

```
    }  
    else getimage(l, t, r, b, image_voc);  
}  
  
void sh_image_voc(int l, int t)  
{  
    putimage(l, t, image_voc, COPY_PUT);  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
/*          PH_INCL.I          */
/*  FUNCTION : pre-include files for phase-program  */
/*-----*/

int TxCol, say_mode=0,wr_screen=0;

```

```

#include <graphics.h>
#include <stdio.h>
#include <ctype.h>
#include <dos.h>
#include <time.h>
#include <string.h>
#include <process.h>
#include <dir.h>

#include <io.h>
#include <dic_coor.i>
#include <ph_thai.i>
#include <ph_drw.i>
#include <ph_image.i>
#include <dicmouse.i>
#include <ph_delay.i>
#include <dicsnow.i>
#include <ph_damag.i>
#include <keys.i>
#include <capslock.i>
#include <cdatadic.i>
#include <tkovfile.i>
#include <ph_study.i>
#include <ph_pvd.i1>
#include <ph_pvd.i2>
#include <ph_pvd.i3>
#include <databas2.i>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <ph_datfi.i>  
#include <btrees.i>  
#include <ph_crt.i>  
#include <ph_sys.i>
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
/*
/*          PH_STUDY.I          */
/*          FUNCTION : STUDY ANY USER      */
/*
/*-----*/

```

```

#define F5 191
#define MAX_STU 30
int old_c, kb_press;
extern void ClsStatusKey();
extern void say_command();
extern void BarLine();
extern void Make_Screen();

int GetKeyMou2(void)
{
    register int x,y,step,mx,my;
    int c=999,i,j;
    for(;;){
        while(kbhit() {
            c=(c=getch())==0 ? getch():c;
            kb_press=1;
        }
        break;
    }
    if(m_events->flag && m_events->button==1 ) {
        if(m_events->xaxis<box_width) {
            if(m_events->yaxis<VStep)
                c=UP;
            else if(m_events->yaxis<2*VStep)
                c=DN;
            else if(m_events->yaxis<3*VStep)
                c=FWD;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if(m_events->yaxis<4*VStep)
c=BS;

else if(m_events->yaxis<5*VStep)
c='\r';

else if(m_events->yaxis<6*VStep)
c=INS; //new student
if(m_events->yaxis>6*VStep)
c=ESC; //cancel

kb_press=1;
}
else { //student-number
c=999;
}

if(c!=999)
break;
}
if(c!=999)
break;
}
KeyMouRelease();
return c;
}

void MakeCrtStudy(void) {
Make_Screen(12,0);
setwriteMode(COPY_PUT);
m_hide();
bar_del(maxx/18,maxy/5, maxx-maxx/18, maxy-maxy/6,LIGHTGRAY);
bar_color(maxx/3+28,maxy/7+15,maxx-maxx/3-30,maxy/7+40,
LIGHTBLUE,WHITE,2);
ThaiStrShade(maxx/2,maxy/7+19,"ลงทะเบียนนักศึกษา",WHITE,RED,2,Yes);
say_command("ครั้งแรกเลือก*ลงทะเบียนใหม่*");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

void box_choose(int col, int row, int color) {
    register int x, y;
    ToView(2);
    col--;
    row--;
    m_hide();
    x=3+(col*(maxx/4));
    y=(maxy/5)+(27*row)+19;
    setwriteMode(XOR_PUT);
    setColor(color);
    BarLine(x,y,x+maxx/4-5,y+25);
    m_show();
}

char *File2Message(char *Filename) {
    FILE *fp;
    unsigned char c, *msg="#####";
    char *ExitCode="##";
    int i;

    if((fp=fopen(Filename,"r"))==NULL) {
        error_message("file not found");
        strcpy(ExitCode,"-1");
        return ExitCode;
    }
    i=0;
    c=(unsigned char)getc(fp);
    while(!feof(fp)) {
        *(msg+i++)=(char)c;
        c=(char)getc(fp);
    }
}

```



```

fclose(fp);
*(msg+i)='\0';
return msg;
}

void Message2File(char *Filename, char *Username, char *Alpha,
char *Item) {
FILE *fp;
char *bftom="#####";

if((fp=fopen(Filename,"w"))==NULL) {
    post_notice("Cannot open to write file");
    exit(1);
}
else{
    sprintf(bftom,"%s%s%s\n",Username,Alpha,Item);
    *(bftom+40)='\0';
    fprintf(fp,"%s",bftom);
    fclose(fp);
}
}

int RegisterFor(int max_stu) {
int c,i, student=1, col=1, row=1, maxrow;

char *msg[]={
    "\n",
    "ท่านต้องการใช้ข้อมูลเดิมหรือไม่<Y/N> ?",
    "ท่านต้องการออกจากการลงทะเบียน<Y/N> ?"
};
maxrow=max_stu/4+1;
setwriteMode(COPY_PUT);
sh_draw1(maxx-maxx/18-95,maxy/5-59,"learn.drw");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

do{
box_choose(col,row,WHITE);
if((c=GetKeyMou2()) <= 0) {      /// if((c=GetKeyMou2()) <= 0) {
    box_choose(col,row,WHITE);
    row=student/4+1;
    col=student-(4*(row-1))+1;
}
else {
    box_choose(col,row,WHITE);
    switch(c) {
        case UP: row--row<1?1:row;
            break;
        case DN: if(++row==maxrow) {
            if(col>max_stu%4)
                --row;
            }
            if(row>maxrow)
                row=maxrow;
            break;
        case BS: col--col<1?1:col;
            break;
        case FWD:
            if(row<maxrow)
                col=++col>4?4:col;
            else if(col%4<max_stu%4)
                ++col;
            break;
        case '\r':
            if((c=ChkYN(msg[1],No))==Yes) {
                student=4*(row-1)+col; //rebuild user-list
                c=-3;
                break;
            }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if(c==No){ //new data unit
    m_hide();
    UserDefineUnit(1);
    student=c=-2; //to list_user
    break;
}
else //c==cancel
    c=0;
break;
case INS:
if(max_stu+1>MAX_STU) {
    error_message("เพิ่มไม่ได้แล้ว");
    //student=-2;
    c=0;
    break;
}
UserDefineUnit(1);
student=c=-2; //to list_user
break;
case ESC:
if((c=ChkYN(msg[2],No))==Yes)
    student=c=-1;
else
    c=0;
break;
default:
break;
}
}
}while( c!=-1 && c!=-2 && c!=-3 );
ToView(2);
ClsStatusKey();
bar_del(2,maxy/5-15, maxx-2, maxy-maxy/6,bkcolor);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return(student);
}

#define USER_FILE_MEM 2000

//-----EDIT ALL DATA ..BUT NOT SAVE IT-----
char *EditUnitUser(char *userbuff, char *username, char *alpha_unit,
    char *item_unit, char *Found) {

char *alpha, *item, *buffitem;

int i=0,j=0;

buffitem="#####"; //46
sprintf(buffitem,"%s%s%s",username,alpha_unit,item_unit); //kill bug

strcpy(alpha,alpha_unit);
strcpy(item,item_unit);

buffitem="#####";
sprintf(buffitem,"%s%s",alpha_unit,item_unit); //kill bug
while(i<USER_FILE_MEM-40) {
    if(strncmp(userbuff+i,username,30)==0) { // found vocab
while(*(userbuff+i++)!='\n'); //new-data olduser
        buffitem="#####";
        sprintf(buffitem,"%s %s\n",alpha_unit,item_unit); //get next data
        for(j=0; j<strlen(buffitem); j++) //insert new data
            *(userbuff+i+j)=*(buffitem+j);
        strcpy(Found,"1");
        return Found;
    }
    else //No found
        i++;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}          //WHILE
strcpy(Found,"0");
return Found;
}

//-----rd_mode==0:make screen but 1: make no screen
int ShowUserUnitName(int item_num, int rd_mode) {
FILE *fp,*fp2;
unsigned char c;
int i,j,k,student=0,y,x,lin,col,bf_len;
char *bfitem="#####", *bf_users;

if((fp=fopen("unituser.dat","r"))==NULL) {
    post_notice("file not found");
if((fp=fopen("unituser.dat","w"))==NULL) {
    error_message("Cannot open to write file");
    return 0;
}
else{
    fputs("\nฟ้าคะ          \na 00002\n"
"รุ่ง          \nb 00150\n",fp); //NAME-LONG=30
    fclose(fp);          //for chk later
}
}
else {
    fclose(fp);
}

if((bf_users=(char *)malloc(USER_FILE_MEM))==NULL) {
    printf("Not enough memory");
    return 0;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if((fp=fopen("unituser.dat","r"))==NULL) {
    post_notice("file not found");
    free(bf_users);
    return 0;
}
i=0;
c=(unsigned char)getc(fp);
while(!feof(fp)) {
    *(bf_users+i++)=(char)c;
    c=(char)getc(fp);
}
fclose(fp);
if(rd_mode==0) {
    MakeCriStudy();
}
col=0;
lin=0;
bf_len=strlen(bf_users);
for(i=0; i<bf_len; i++) {
    while(*(bf_users+i)=='\n' && i<bf_len)
        i++;
    j=i;
    k=0;
    while(++j<50+i && j<bf_len) { // long-name approximate
        if(*(bf_users+j)=='\n') {
            k++;
        }
    }
    if(k==0) // no data-name
        break; // exit main-loop
    j=0;
    while(*(bf_users+i)!='\n' && i<bf_len) {
        *(bf+m+j) = *(bf_users+i);
    }
}

```

```

        j++;
        i++;
    }
    *(bfitem+j)='\0';
    student++;

    x=3+(col*(maxx/4));
    y=(maxy/5)+(27*lin)+20;
    if(rd_mode==0) {
        bar_color(x,y,x+maxx/4-5,y+25,CYAN,WHITE,2);
        ThaiStrShade(x+7,y+4,bfitem,WHITE,BLACK,2,No);
    }
    else if(rd_mode==1) { //rd_mode==1
        if(student==item_num) {
            if((fp2=fopen("usertemp.dat","w"))==NULL) {
                error_message("Cannot open to write file");
                return 0;
            }
            else{
                strncpy(bfitem,bf_users+i-30,46);
                *(bfitem+40)='\0';
                free(bf_users);
                fprintf(fp2,"%s",bfitem);
                fclose(fp2); //for chk later
                return 1;
            }
        }
    }
}

if(col++>2) {
    col=0;
    lin++;
}

```

```

while(*(bf_users+i)=='\n' && i<bf_len) //between name and alpha
    i++;
while(*(bf_users+i)!='\n' && i<bf_len) //between alpha + item
    i++;
i--;
}
free(bf_users);
return student;
}

//-----SERACH NAME,IF FOUND SAVE DATA , IF NO, CREAT AND SAVE-----

int UnitUserFile(char *username, char *useralpha,
    char *useritem) {
//rd_wr==0 ==>no found
//rd_wr==1 ==>found

FILE *fp;
unsigned char c;
int i;
char *bf_users , *Found, *name, *alpha, *item, *bftom;

if((fp=fopen("unituser.dat","r"))==NULL) {
    post_notice("file not found");
if((fp=fopen("unituser.dat","w"))==NULL) {
    post_notice("Cannot open to write file");
return 0;
}
else{
fputs("\nฟ้าทะละ          \na 00002\n"
    "\nรุ่ง          \nb 00150\n",fp); //NAME-LONG=30
fclose(fp);          //for chk later
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

}
else {
    fclose(fp);
}

if((bf_users=(char *)malloc(USER_FILE_MEM))==NULL) {
    printf("Not enough memory");
    return 0;
}

if((fp=fopen("unituser.dat","r"))==NULL) {
    post_notice("file not found");
    return 0;
}
i=0;
c=(unsigned char)getc(fp);
while(!feof(fp)) {
    *(bf_users+i++)=(char)c;
    c=(char)getc(fp);
}
fclose(fp);

Found="#";
name="#####";
alpha="#";
item="####";
bfitem="#####"; //46
printf(bfitem,"%s%s%s\n",username,useralpha,useritem); //kill bug

strcpy(name,username);
strcpy(alpha,useralpha);
strcpy(item,useritem);
EditUnitUser(bf_users, name, alpha, item, Found);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

switch(i=atoi(Found)) {
    case 0: //user-name not found
        post_notice("No Found.");
        if((fp=fopen("unituser.dat","a+"))==NULL) {
            error_message("Cannot open to append file");
            return 0;
        }
        sprintf(bf_users,"%s\n%s %s\n",name,alpha,item);
        break;
    case 1: // user-name is found
        if((fp=fopen("unituser.dat","w+"))==NULL) {
            error_message("Cannot open to write file");
            return 0;
        }
        break;
    default: post_notice("ERROR");
        return 0;
}

i=strlen(bf_users);
while(!isascii(*(bf_users+i)) && *(bf_users+i)!='\n')
    *(bf_users+i)='\0';

fwrite(bf_users,sizeof(char),strlen(bf_users),fp);
fclose(fp);
free(bf_users);
return 1;
}

```

```

int GetOrgEndUnit(char *alpha, char *org, char *end) {
int i;
char *msgtem1="#####", *msgtem2="#####",
*StartEndItem[26][2]={

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

"00001","00149", //A
"00150","00249", //B
"00250","00487", //C
"00488","00667", //D
"00001","00117", //E
"00118","00250", //F
"00251","00349", //G
"00350","00441", //H
"00443","00553", //I
"00554","00580", //J
"00581","00606", //K
"00175","00240", //L
"00241","00295", //M
"00296","00312", //N
"00313","00368", //O
"00369","00578", //P
"00607","00622", //Q
"00623","00775", //R
"00604","00837", //S
"00001","00152", //T
"00153","00174", //U
"00579","00603", //V
"00668","00733", //W
" - "," - ", //X ///Missing phasal verb
"00734","00736", //Y
"00737","00739" //Z

```

```
};
```

```

if(!isalpha(*(alpha)))
    return -1;
else {
    i=(int)(tolower(*(alpha)));
    if(i==120) { // ==> 'x'==120 //Not yet datas in 'X'
        m_show();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return 0;
    }
}
strcpy(msgtom1,StartEndItem[i-97][0]); //ascii a=97 b=98 ,...
strcpy(msgtom2,StartEndItem[i-97][1]);
for(i=0; i<6; i++) {
    *(org+i) = (i<5)?*(msgtom1+i):'\0';
    *(end+i) = (i<5)?*(msgtom2+i):'\0';
}
return 1;
}
;
//-----mode define say message-----

int UserDefineUnit(int mode) {
char *username="#####", *alpha="##",
    *item, *msg, *exit_code="0";
int i,c;
ToView(2);
do {
    strcpy(username,""); //30
    bar_color(maxx/9,maxy/5,maxx-maxx/5,maxy-maxy/5,CYAN,WHITE,4);
    setwritemode(COPY_PUT);
    ThaiStrShade(mx5+14,maxy/3+50,"ไปรคบันทึกรชื่อ",WHITE,RED,2,No);
    GetString((unsigned char)'A',"-----",
    username,exit_code); //40 but rest 1byte array for'\0'
    if(strcmp(exit_code,"1")!=0) { //press ESC
        return 0; //exit function
    }
    if((i=NoCountSpaceEnd(username))<=0){
        error_message("ชื่อของท่านเป็นฉันทัด");
        i=-1;
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

} while(i<=0); //Name not found
do{
    c=0;
    strcpy(alpha, " ");
    bar_del(mx5+4,maxy/3+45,maxx-maxx/4,maxy/3+70,CYAN);
    setwriteMode(COPY_PUT);
    ThaiStrShade(mx5+14,maxy/3+50,(mode<=1)?"โหมคอักษร <A...Z>":
    "โหมคอักษรทดสอบ <A..Z>",
    WHITE,RED,2,No);
    GetString((unsigned char)'E','_',alpha,exit_code);
    if(strcmp(exit_code,"1")==0) //press ESC
    return 0; //exit function
    item="#####"; //initialize
    msg="#####";
    switch(GetOrgEndUnit( alpha, item, msg)) { //msg store end_item
case 0:
    post_notice("ไม่มีคำศัพท์");
    c=0;
    break;
case -1:
    error_message("โปรดพิมพ์ตัวอักษรตัวอักษร");
    c=0;
    break;
default:
    c=1;
    break;
    }
}while(c==0); //LOOP

*(alpha+1)='\0';
if(c!=0) {
    UnitUserFile(username, alpha, item);
    return 1; //true
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
else {
    error_message("Cannot write file");
    return 0; //false
}
}

int ChooseAlpha(int mode) {
char *alpha="##",
    *item="#####", *msg="#####", *exit_code;
int i,c;

FILE *fp;
char *bftom="#####";

do{
    c=0;
    strcpy(alpha, " ");
    bar_del(mx5+4,maxy/3+45,maxx-maxx/4,maxy/3+70,CYAN);
    setwriteMode(COPY_PUT);
    ThaiStrShade(mx5+14,maxy/3+50,(mode<=1)?"โหมดอักษร <A..Z>":"โหมดอักษรทดสอบ <A..Z>",
    WHITE,RED,2,No);
    GetString((unsigned char)'E',"_",alpha,exit_code);
    if(strcmp(exit_code,"1")==0) //press ESC
        return 0; //exit function
    item="#####"; //initialize
    msg="#####";

    switch(GetOrgEndUnit( alpha, item, msg)) { //msg store end_item
case 0:
    post_notice("ไม่มีคำศัพท์");
    c=0;
    break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case -1:
    error_message("โปรดพิมพ์ตัวอักษรตัวอักษร");
    c=0;
    break;
default:
    c=1;
    break;
}
}while(c==0); //LOOP

*(alpha+1)='\0';
if(c!=0){
    if((fp=fopen("usertemp.dat","w"))==NULL) {
        error_message("Cannot open to write file");
        exit(1);
    }
    else{
        bftem="ฟ้าทะ";
        *(bftem+31)='\na 00002\n';
        fprintf(fp,"%s",bftem);
        fclose(fp);
    }
    return 1; //true
}
else {
    error_message("Cannot write file");
    return 0; //false
}
}

void StudyUnit(void) {
//char *msg="#####";

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int i,max_stu;
ToView(2);
bar_del(maxx/18, maxy/5,maxx-maxx/18,maxy-maxy/6,bkcolor);
do {
max_stu = ShowUserUnitName(0 , 0); //0second=show user-list
i=RegisterFor(max_stu); // i return also student no.
if(i>=0){
ShowUserUnitName(i, 1); // 1= save current data-user, not on creen
LookForPhase(0,1); //0first=vocphase.. //1second=student
clear_record(); //0second=dic-add-phase
ClearMemData();
db_cls();
}
} while (i!= -1);

m_show();
}

int SaveDataUser(char *msg,int mode, int fi_chg) {
//mode0=DIC,mode1=study,mode2=game
//fi_chg=0: No change userfilename ,study-mode
//fi_chg=1: Change userfilename ,study-mode
int i,c,y,min_x,oldmode,temp=ModeTemp;
char *ext, *exit_code, /* *alphas,*/ *msgtem=" ";
FILE *fp,*fp2;
struct data {
char name[41];
char cap_mode[2];
char item_unit[6];
char org_unit[6];
char end_unit[6];
}data_user;
char *msg_save[] = {

```



```

"บันทึกศัพท์",
",
"ชื่อไฟล์",
"[ ][.dic]",
"เก็บในมาตรฐาน สมอ.",
", //5
"Help",
"ยกเลิก",
"HELP-บันทึกศัพท์",
" บันทึกไว้เพื่อพิมพ์หรือ",
"แสดงในเวิร์ดโปรเซส", //10
"เซอร์โคจ",
",
",
" //14
};

char *msg_regis[] = {
"ลงทะเบียน",
",
"ชื่อไฟล์",
"[ ][.stm]",
"(เก็บด้วยสกุล .stm)",
", //5
"Help",
"ยกเลิก",
"HELP-ลงทะเบียน",
" จดจำชื่อและ",
"สถานะเพื่อใช้งาน", //10
"โอกาสต่อไป",
",
",
" //14

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

};

```

char *StartEndItem[26][2]={
    "00001","00149", //A
    "00150","00249", //B
    "00250","00487", //C
    "00488","00667", //D
    "00001","00117", //E
    "00118","00250", //F
    "00251","00349", //G
    "00350","00441", //H
    "00443","00553", //I
    "00554","00580", //J
    "00581","00606", //K
    "00175","00240", //L
    "00241","00295", //M
    "00296","00312", //N
    "00313","00368", //O
    "00369","00578", //P
    "00607","00622", //Q
    "00623","00775", //R
    "00604","00837", //S
    "00001","00152", //T
    "00153","00174", //U
    "00579","00603", //V
    "00668","00733", //W
    " - "," - ", //X ///No phasal verb
    "00734","00736", //Y
    "00737","00739" //Z
};

```

```
strcpy(msgtem,fileuser);
```

```
ext=(mode==0)?" .dic":(mode==1)?" .stu": ".gmo";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

m_hide();
if(fi_chg==1) {
    *(fileuser+12)='\0'; //protect error long-name exceed 12 alphas
    ToView(4);
    HELPPABOUT=1; //define area-pointer of mouse in GetKeyMou()
    ModeTemp=SV_USER; // ON cursor;
    sv_image1(0,0,maxx,maxy);

    LOOP:
    m_hide();
    setcolor(WHITE);
    WinButton(mode==0?msg_save[0]:msg_regis[0],"***");
    Shadow(20,20,maxx,maxy,maxy-20-2,maxx-20-1);
    for(i=1; i<8; i++) {
        ThaiStrShade(maxx/2-10,(i<6?(i*20)+15:(i*27)-23),
            mode==0?msg_save[i]:msg_regis[i],
            i<5?LIGHTGREEN:YELLOW,BLACK,2,Yes);
    }
    min_x=c=maxx/4-13;
    y=maxy/2-46;
    setcolor(YELLOW);
    for(i=0; i<strlen(fileuser); i++) { //display old filename ,if even exiting..
if(*(fileuser+i)!='.') {
        c=rdcw_attrib(c,y,0,*(fileuser+i));
}
else{
        for(c=i; c<=i+4; c++) // delete extension name for creat later
            *(fileuser+c)=' ';
        *(fileuser+c)='\0';
        break;
}
}
moveto(min_x,y);

```

```

i=0;
say_mode=THAI; //sh_cursor THAI style
do { ToView(4);
    m_show();
    c=GetKeyMou(min_x+(i*8),y);
    ToView(4);
switch(c) {
case '*': ///////////////
case '+': //
case '=': // invalid ascii for naming the file
case '\\': //
case '!': ///////////////
case ':':
case '\':
    printf("\a"); //beep
    c=0;
    break;
case '\b': //backspace
goto LEFT;
case -95:
case F1:
    VocBarAtt(13);
    m_hide();
    WinButton(msg_regis[8],"*");
    for(i=0; i<6; i++) {
        setcolor(YELLOW);
        ThaiStrJustify(maxx/2-10,(i*25)+maxy/6,
            msg_regis[i+9],Yes);
    }
    m_show();
    while((i=GetKeyMou(-50,-50))!= -93 && i!=ESC);
    VocBarAtt(11);
    KeyMouRelease();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    . ToView(4);
      goto LOOP;
case -93:
case '\r':
    c='\r';
    VocBarAtt(11);
    if(*fileuser==' ') {
        printf("\a");
        c=0;
    }
    break;
case ESC: //Cancel button
case -94:
    c=ESC;
    VocBarAtt(12);
    break;
case FWD: //right
    i++;i>77?i;
    moveto(min_x+(i*8),y);
    c=0;
    break;
case BS: //left
    LEFT:
    i--;i<0?0;i;
    moveto(min_x+(i*8),y);
    c=0;
    break;
case INS:
case DEL:
    break;
default:
    if(endstroke(c))
        break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        *(fileuser+i)=(char)c;
        moveto(min_x+(i*8),y);
        bar_dol(min_x+(i*8),y,min_x+7+(i*8) ,y+20,LIGHTBLUE);
        setcolor(YELLOW);
        rdcw_attrib(min_x+(i*8),y,0,c);
        i++;i>77?i;
        moveto(min_x+(i*8),y);
        break;
    } //SWITCH
    }while(ct==' ' && ct==ESC );
    HELPABOUT=0;
    ModeTemp=oldmodotemp;
    m_hide();
    sh_image1(0,0);
    if(c==ESC)
c=-1; // goto ExitFunc;
} //(MODE==0) {
else //fi_chg=0
    c='\n'; //for line below

if(c=='\n') {
    if(mode>=1) {
        ToView(2);
        setwritemode(COPY_PUT);
        bar_color(maxx/18,maxy/5,maxx-maxx/18,maxy-maxy/6,CYAN,WHITE,4);
        bar_color(maxx/4,maxy/5+10,maxx-maxx/4,maxy/5+43,LIGHTBLUE,WHITE,4);
        ThaiStrShade(maxx/2,maxy/5+19,mode<=1?"ทำรายการใหม่":"Recording your background",
WHITE,RED,3,Yes);
        do {
            strcpy(data_user.name,"");
            bar_dol(mx5+4,maxy/3+45,maxx-maxx/4,maxy/3+70,CYAN);
            ThaiStrShade(mx5+14,maxy/3+50,"โปรดบันทึกชื่อ",WHITE,RED,2,No);
            GetString((unsigned char)'A',"_____");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

data_user.name,exit_code); //40 but rest 1byte array for'\0'

if(strcmp(exit_code,"1")==0) { //press ESC
    // c=-1;
    m_show();
    return -1;
}

    if((i=NoCountSpaceEnd(data_user.name))<=0)
error_message("ชื่อของท่านเป็นนินจา");
} 'while(i<=0); //Name not found

// printf("T1:.nam=(%s)",data_user.name);
UsersName=data_user.name;
// *(UsersName+40)=(char)\0';
// printf("T2:Un=(%s)",UsersName);
do{
    // alphas=" ";
    strcpy(data_user.cap_mode," ");
    bar_del(mx5+4,maxy/3+45,maxx-maxx/4,maxy/3+70,CYAN);
    ThaiStrShade(mx5+14,maxy/3+50,mode<=1?"โหมคอักษร <A...Z>":"โหมคอักษรทดสอบ <A...Z>",
    WHITE,RED,2,No);
//    strcpy(data_user.cap_mode,GetString((unsigned char)'E','_',alphas,exit_code));
//    if(strcmp(exit_code,"1")==0) { //press ESC
    GetString((unsigned char)'E','_',data_user.cap_mode,exit_code);
    if(strcmp(exit_code,"1")==0) { //press ESC

        // c=-1;
        m_show();
        return -1;
    }

//    printf("T3:.cap=(%s)",data_user.cap_mode);
//    if((i=isalpha((int) *alphas))==0) //not alphabet
    if((i=isalpha((int) *data_user.cap_mode))==0) //not alphabet

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

error_message("A..Z เท่านั้น");
}while(i==0); // not alphabet

// printf("T4:");
*(data_user.cap_mode+1)=(char)\0;
UsersAlpha=data_user.cap_mode;
*(UsersAlpha+1)=(char)\0;
// printf("T5:UA1=(%s)",UsersAlpha);

i=(int)tolower((int) *(UsersAlpha));
// printf("T6 i=%d "i);
//---- Alpha X : no existing the data ----//

if(i==120) { // ==> 'x'==120
    post_notice("ไม่มีคำศัพท์");
    m_show();
    return -1;
}

//for save: .org_unit is same .item_unit
strcpy(data_user.org_unit,StartEndItem[i-97][0]);
strcpy(data_user.item_unit,StartEndItem[i-97][0]);
UsersItemUnit=data_user.item_unit;
UsersOrgUnit=data_user.org_unit;
strcpy(data_user.end_unit,StartEndItem[i-97][1]);
UsersEndUnit=data_user.end_unit;
} //IF.MODE>=1
if(fi_chg==0)
    strcpy(fileuser,msgtom);
//naming the file
if(fi_chg==1){
    *(fileuser+NoCountSpaceEnd(fileuser))='\0';
    strcat(fileuser,ext);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

}
i=0;
while(*(fileuser+i++)!='.');
```

เปิดไฟล์ไม่ได้

```

*(fileuser+(i=(i>9)?9:i)+3)='\0';
if(mode==0) {
    if((fp=fopen(fileuser,"a"))==NULL) {
        error_message("เปิดไฟล์ไม่ได้");
        c=-1;
    }
    else {
        fwrite(msg,sizeof(char),strlen(msg),fp);
        fclose(fp);
        c=0;
    }
}
else { //mode!=0 or mode1 mode2
    TakeOverUserFile((mode==1)?(char *)fileuser:"fighter.gmo",1/*wr*/,data_user.name ,
    data_user.cap_mode,data_user.item_unit,
    data_user.item_unit/*data_user.org_unit*/,data_user.end_unit);
    c=0;
}
} //IF(C=='R'
m_show();
return c;
}

```

```

/*-----*/
/*          PH_SYS.C          */
/*    FUNCTION : cursor manager    */
/*-----*/

/* ----- get a keyboard character ----- */
int sh_cursor( );
int get_char( );
void put_char(int c);

int get_char( )
{
    int c;
    sh_cursor(getx( ), gety( )+14,say_mode);
    if((c=getch( ))==0)
        c=getch( )|128;
    return (c);
}

/* ----- write a character to the screen ----- */
void put_char(int c) /* c NO USE */
{
    switch (c)    {
        case FWD:    /*gotoxy(wherex( )+1,wherey( ));*/
                    break;
        case UP:     /*gotoxy(wherex( ),wherey( )-1);*/
                    break;
        default:     break;
    }
}

/* ----- set the cursor position ----- */
void cursor(x,y)

```

```

int x, y;
{
    moveto(x*8,y*30);
}

/* ----- clear the screen ----- */
int screen_displayed = 0;
void clear_screen( )
{
    clearviewport( );
}

int sh_cursor(x, y, m)
register int x,y,m; /* m==mode*/
{
    register int i,c,M=0;

do {
    if(ModeTemp==GET_PH)
        if(m_events->xaxis>view3.left && m_events->xaxis<view3.right &&
            m_events->yaxis>view3.top && m_events->yaxis<view3.bottom)
            m_hide( );
        else
            m_show( );
    if(ModeTemp==GET_WORD || ModeTemp==EDIT_PH || ModeTemp==ADD_PH ||
        ModeTemp==SV_USER || ModeTemp==GET_STU) {
        setlinestyle(0,1,m==THAI?3:1);
        setwriteMode(XOR_PUT);
        spcolor(YELLOW);
        for(i=0;i<2;i++)
            {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

switch(m) {
    case THAI:
        line(x,y,x+7,y);
        line(x,y-3,x+7,y-3);
        line(x,y-6,x+7,y-6);
        line(x,y-9,x+7,y-9);
        DelayChkKey(0.03);
        break;
    default: line(x,y+2,x+7,y+2);
        line(x,y+3,x+7,y+3);
        DelayChkKey(0.02);
        break;
}
}
}

if(m_events->flag==0x02) {
    M=CLIC;
    break;
}
}. while(!kbhit() && M==0);
setwriteMode(COPY_PUT);
return(M);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
/*      PH_THAI      */
/*      FUNCTION : my thai driver      */
/*-----*/

#define BYTECHAR  160

#define ON        1
#define BLANK_KEY 25
#define ESC_KEY   1
#define BACKSPACE 8
#define THAI      0
#define ENG       3

int color, maxx, maxy, maxcolor, bkcolor;
unsigned char font[5120],thaikay[BYTECHAR],ascii,funckey;
int asc_scancode(unsigned char asc, int say_mode);
extern void m_hide();

void readfont()
{
    int file;
    file = open ("phasetai.fon", 1);
    read(file, thaikay, BYTECHAR);
    read(file, font, 5120);
    close (file);
}

void opengraph()
{
    int grdriver, grmode,gr_error;
    detectgraph(&grdriver, &grmode);
    if(grdriver<0) {
        printf("\nNo graphics hardware detected !");
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    exit(1);
}
grdriver=9;
grmode=2;
initgraph(&grdriver,&grmode,"");
if((gr_error=graphresult())!=grOk) {
    printf("\nOpen Graphic ERROR: %s.",grapherrormsg(gr_error));
    exit(1);
}
readfont();
cleardevice();
maxx=getmaxx();
maxy=getmaxy();
box_width=maxx/7-1;
setbkcolor(BLACK);
setpalette(BLACK,BLACK);
maxcolor=getmaxcolor();
bkcolor=LIGHTGRAY;
}

void showchar(register int x, register int y, unsigned char ASCII)
{
    register int k;

    ASCII-=7;
    for (k=0; k<=19; k++) {
        if (font[(ASCII-1)*20+k] != 0) {
            if ((biosequip() & 48) == 48)
                setlinestyle(4,font[(ASCII-1)*20+k] << 7,1);
            else setlinestyle(4,font[(ASCII-1)*20+k],1);
            line(x+7,y+k,x,y+k);
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

int asc_scancode(unsigned char c, int say_mode)
{
int asc;

/* Translate from keyboard TO So-mo-or */
if((c==32)|(c==27)) asc = (c==32) ? BLANK_KEY : ESC_KEY;
else
{
if(say_mode==THAI)
{
switch(c) {
case 100:    asc=154; break; /* ก */
case 45:    asc=155; break; /* ข */
case 56:    asc=157; break; /* ค */
case 83:    asc=159; break; /* ก */
case 39:    asc=160; break; /* ก */
case 48:    asc=161; break; /* ก */
case 67:    asc=162; break; /* ง */
case 61:    asc=163; break; /* ก */
case 58:    asc=164; break; /* ก */
case 71:    asc=165; break; /* ช */
case 80:    asc=166; break; /* ช */
case 69:    asc=167; break; /* ฉ */
case 68:    asc=168; break; /* ฉ */
case 123:   asc=169; break; /* ฎ */
case 82:    asc=170; break; /* ฎ */
case 60:    asc=171; break; /* ฐ */
case 73:    asc=172; break; /* ฑ */
case 102:   asc=173; break; /* ฒ */
case 57:    asc=174; break; /* ณ */
case 53:    asc=175; break; /* ด */
case 109:   asc=176; break; /* ต */

```

```

case 84:      asc=177; break; /* ถ */
case 111:    asc=178; break; /* ท */
case 91:     asc=179; break; /* ฑ */
case 120:    asc=180; break; /* น */
case 122:    asc=181; break; /* ฅ */
case 47:     asc=182; break; /* ฎ */
case 114:    asc=183; break; /* ฌ */
case 97:     asc=184; break; /* ฬ */
case 52:     asc=185; break; /* ฬ */
case 44:     asc=186; break; /* ฬ */
case 112:    asc=187; break; /* ฎ */
case 105:    asc=188; break; /* ฌ */
case 65:     asc=189; break; /* ฐ */
case 93:     asc=190; break; /* ฎ_right */
case 49:     asc=190; break; /* ฎ_lefttop */
case 59:     asc=192; break; /* ฎ */
case 76:     asc=193; break; /* ฎ */
case 75:     asc=194; break; /* ฏ */
case 108:    asc=195; break; /* ฏ */
case 115:    asc=196; break; /* ฐ */
case 62:     asc=197; break; /* ฌ */
case 118:    asc=198; break; /* ฎ */
case 86:     asc=199; break; /* ฑ */
case 79:     asc=200; break; /* ฎ */
case 116:    asc=201; break; /* ฬ */
case 121:    asc=202; break; /* mai_han_argart */
case 107:    asc=203; break; /* ฐ */
case 101:    asc=204; break; /* ฎ */
case 98:     asc=205; break; /* sla_ii */
case 117:    asc=206; break; /* sla_ee */
case 55:     asc=207; break; /* ฎ */
case 110:    asc=208; break; /* ฎ */
case 54:     asc=209; break; /* ฎ */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

case 94: asc=210; break; /* ฤ */
 case 38: asc=212; break; /* ฮ อ */ /*Han-arkard & Maito*/
 case 103: asc=217; break; /* ๕ */
 case 99: asc=218; break; /* ๖ */
 case 70: asc=219; break; /* ๗ */
 case 46: asc=220; break; /* ๘ */
 case 119: asc=221; break; /* ๙ */
 case 33: asc=222; break; /* ชช(๖งศษพณ ฤศร ก) */
 case 113: asc=223; break; /* ๑ */
 case 72: asc=224; break; /* ๒ */
 case 106: asc=225; break; /* ๓ */
 case 104: asc=226; break; /* ๔ */
 case 85: asc=227; break; /* ๕ */
 case 74: asc=228; break; /* ๖ */
 case 78: asc=229; break; /* ๗ */
 case 50: asc=40; break; /* / */
 case 51: asc=38; break; /* . */
 case 87: asc=27; break; /* " */
 case 125: asc=32; break; /* ' */
 case 34: asc=39; break; /* , */
 case 90: asc=33; break; /* (*/
 case 88: asc=34; break; /*) */
 case 81: asc=233; break; /* 0 THAI*/
 case 64: asc=234; break; /* 1(thai) */
 case 35: asc=235; break; /* 2(thai) */
 case 36: asc=236; break; /* 3(thai) */
 case 37: asc=237; break; /* 4(thai) */
 case 42: asc=238; break; /* 5(thai) */
 case 40: asc=239; break; /* 6(thai) */
 case 41: asc=240; break; /* 7(thai) */
 case 95: asc=241; break; /* 8(thai) */
 case 43: asc=242; break; /* 9(thai) */
 case 66: asc=211; break; /* * */

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 77:      asc=231; break; /* */
case 63:      asc=191; break; /* */
case 89:      asc=230; break;
case 92:      asc=85;  break; /* (low)near backspace*/
case 124:     asc=51;  break; /* (top)near spacebar*/
case 126:     asc=30;  break; /* (top)near 1 : % */
case 96:      asc=189; break; /* (low)near 1*/

```

```

default:     break;

```

```

}
}
else if(c<127) asc =(int)c-7; /* sey_mode=ENG */
}
sey_mode += 0; /*-- solv from ERROR --*/
return((int)asc+7);
}

```

```

int QtyMidStr(char *s)

```

```

{
int i,j=0;
for(i=0; i<strlen(s); i++)
if(IsMidChar((unsigned char) *(s+i))==1)
j++;
return(j);
}

```

```

void ThaiStr(int g_col, int g_row, char *content)

```

```

{ /* Translate from so-mo-or to display */
register int i;
for(i=0; i<strlen(content); i++)
g_col = rdcw_attrib(g_col, g_row, 0/*3*/, *(content+i));
}

```

```

void ThaiStrJustify(g_x, g_y, message, is_center)
int g_x, g_y, is_center;
char *message;
{
ThaiStr(g_x==is_center==1 ? g_x-(8*QtyMidStr(message)/2):g_x,g_y, message);
}

```

```

void ThaiStrShade(int g_col, int g_row, char *content, int colorCh,
    int shade,int num,int just) { //Translate from so-mo-or to display
register int i, j;
int c=getcolor();

setcolor(shade);
for(j=0; j<num; j++)
    ThaiStrJustify(g_col+1-j, g_row+1-j, content, just);
setcolor(colorCh);
    ThaiStrJustify(g_col+1-j--, g_row+1-j--, content, just);
setcolor(c);
}

```

```

int rdew_attrib(int column, int row, int attrib, unsigned char content) {
int dec_col=!ON;
register int ypix, xpix;
unsigned char asc;
ypix=0; xpix=0; dec_col=!ON;
if (tolower(content) >160)
    asc=tolower(content);
else
    asc=toascii(content);
if ((asc>211 && asc<216) || asc==209/*MAI_HAN_ARGART*/) {
    xpix=-8;
    dec_col=ON;
    goto SH_CH;
}

```

```

}
if (asc>230 && asc<240) {
    xpix=-9;
    dec_col=ON;
    goto SH_CH;
} /*MAI_TO etc*/
if ( (asc>215) && (asc<219) ) /*SLA_OO etc*/
{
    xpix=-8;
    dec_col=ON;
    goto SH_CH;
}
if (asc==211) /*SLA_AM*/
{
    showchar(column-3,row,237);
    asc=210;
}
SH_CH:
    showchar(column+xpix,row+ypix,asc);
if(attrib==3) showchar(column+xpix,row+ypix+1,asc);
if (dec_col!=ON) column +=8;
    setlinestyle(0,255,1);
return(column);
}

int bar_del(int l, int t, int r, int b, int color) {
    setcolor(color);
    setfillstyle(1,color);
    setlinestyle(0,0,1);
    m_hide();
    bar(l,t,r,b);
    return(getcolor());
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
/*          TKOVFILE.I          */
/*    FUNCTION : user-file manager    */
/*-----*/

void TakeOverUserFile(char *filenam,int rd_wr_mode, char *msg1,
    char *msg2, char *msg3, char *msg4, char *msg5) { //0=rd,1=wr
char *msg;
FILE *fp;
struct data {
    char name[41];
    char cap_mode[2];
    char item_unit[6];
    char org_unit[6];
    char end_unit[6];
}data_user;

if((fp=fopen(filenam,rd_wr_mode==0?"r":"w")==NULL)
    error_message("ไม่สามารถเปิดไฟล์ได้");
else {
    if(rd_wr_mode==0) { //read
        fgets(msg,sizeof(struct data)/*55*/,fp);
        UsersName=msg;
        UsersAlpha=msg+41;
        UsersItemUnit=msg+43;
        UsersOrgUnit=msg+49;
        UsersEndUnit=msg+55;
    }
    else { //mode == 1 write
        strcpy(data_user.name,msg1);
        strcpy(data_user.cap_mode,msg2);
        strcpy(data_user.item_unit,msg3);
        strcpy(data_user.org_unit,msg4);
        strcpy(data_user.end_unit,msg5);
    }
}
}

```

```
fwrite(&data_user,sizeof(struct data),1,fp);  
}  
fclose(fp);  
}  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; ----- ph_pvd.voc Phasal verb dictionary
 ; ----- data element dictionary

#schema PH_PVD

#dictionary

MAIN_WORDS, E, 13, "-----"
 PHASE_LIST, B, 60, "-----"
 MAIN2_WORDS, E, 13, "-----"
 PHAS2_LIST, B, 60, "-----"
 MAIN3_WORDS, E, 13, "-----"
 PHAS3_LIST, B, 60, "-----"
 PHASE_NO, Z, 5, "-----"
 PHASE_NAME, E, 40, "-----"
 PHASE_MEANS, A, 45, "-----"
 EXAM_ENG1, E, 50, "-----"
 EXAM_ENG2, E, 20, "-----"
 EXAM_THAI, A, 70, "-----"
 PHAS2_NO, Z, 5, "-----"
 PHAS2_NAME, E, 40, "-----"
 PHAS2_MEANS, A, 45, "-----"
 EXAM2_ENG1, E, 50, "-----"
 EXAM2_ENG2, E, 20, "-----"
 EXAM2_THAI, A, 70, "-----"
 PHAS3_NO, Z, 5, "-----"
 PHAS3_NAME, E, 40, "-----"
 PHAS3_MEANS, A, 45, "-----"
 EXAM3_ENG1, E, 50, "-----"
 EXAM3_ENG2, E, 20, "-----"
 EXAM3_THAI, A, 70, "-----"
 STUDENTS_NAME, A, 40, "-----"
 MAINWORDS_NO, Z, 5, "-----"
 #end dictionary

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; ----- file specifications

#file DICWORDS

MAIN_WORDS

PHASE_LIST

#end file

#file DICWORD2

MAIN2_WORDS

PHAS2_LIST

#end file

#file DICWORD3

MAIN3_WORDS

PHAS3_LIST

#end file

#file VOCPHASE

PHASE_NO

PHASE_NAME

PHASE_MEANS

EXAM_ENG1

EXAM_ENG2

EXAM_THAI

#end file

#file VOCPHAS2

PHAS2_NO

PHAS2_NAME

PHAS2_MEANS

EXAM2_ENG1

EXAM2_ENG2

EXAM2_THAI

#end file



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

#file VOCPHAS3
PHAS3_NO
PHAS3_NAME
PHAS3_MEANS
EXAM3_ENG1
EXAM3_ENG2
EXAM3_THAI
#end file

```

```

#file STUDENTS
STUDENTS_NAME
MAINWORDS_NO
#end file

```

```

; ----- index specifications

```

```

#key DICWORDS MAIN_WORDS
#key DICWORD2 MAIN2_WORDS
#key DICWORD3 MAIN3_WORDS

```

```

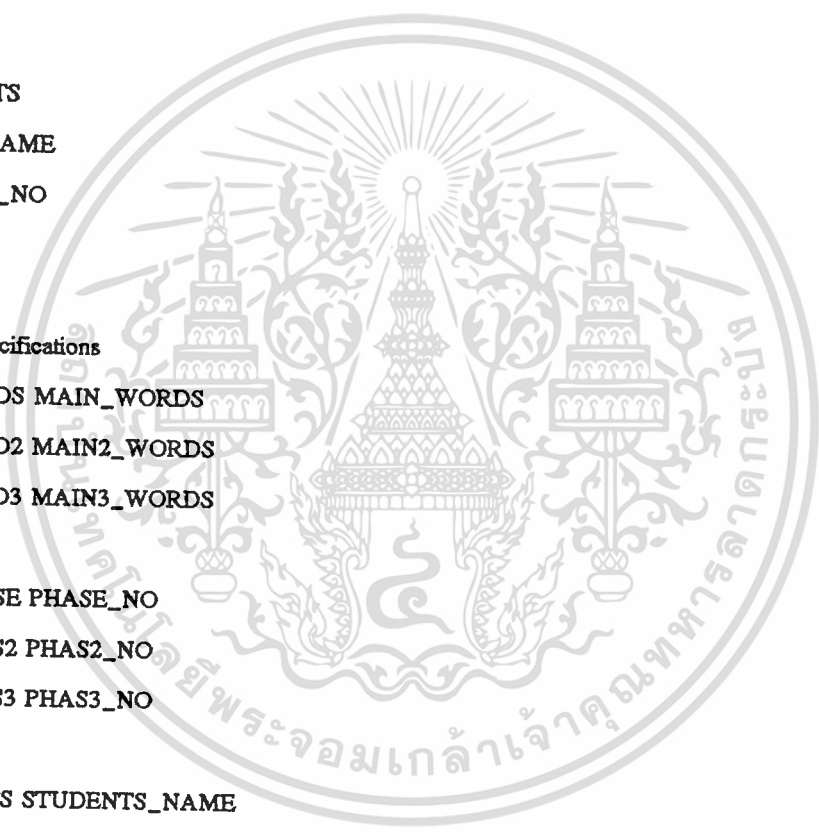
#key VOCPHASE PHASE_NO
#key VOCPHAS2 PHAS2_NO
#key VOCPHAS3 PHAS3_NO

```

```

#key STUDENTS STUDENTS_NAME
#end schema

```



```

/*-----builtcrc.bat-----*/
cd\project
del chk_vr.crc
copy c:\borlandc\myoutput\phases.exe
copy c:\borlandc\myoutput\ph_vrs2.pwa
cd\
copy c:\borlandc\myoutput\ph_vrs1.exe
ph_vrs1.exe
pause
cd\project
copy c:\chk_vr.crc
pause
cd\project
/*-----exe2com.bat-----*/
cd\
copy c:\borlandc\myoutput\ph_vrs2.exe c:\dos5
cd\dos5
exe2bin ph_vrs2.exe ph_vrs2.com
copy ph_vrs2.com c:\borlandc\myoutput
cd\borlandc\myoutput
copy ph_vrs2.com *.pwa
cd\
/*-----ph_schem.bat-----*/
copy c:\borlandc\myoutput\schemdic.exe
ph_schem < ph_pvd.voc > ph_pvd.i1 -1
ph_schem < ph_pvd.voc > ph_pvd.i2 -2
ph_schem < ph_pvd.voc > ph_pvd.i3 -3
pause
copy ph_pvd.voc c:\borlandc\myincl
copy ph_pvd.i* c:\borlandc\myincl

```

บรรณานุกรม

- คร.สุชาย ธนวเสถียร. "โครงสร้างข้อมูล", กรุงเทพมหานคร: หจก.เอช-เอน การพิมพ์, 2535.
- ทนุ เตียวรัตน์กุล. "611 ส่วนอังกฤษ", กรุงเทพมหานคร: มิตรสัมพันธ์กราฟฟิคอาร์ต, 2529.
- ผ.อ. นิจ ทองโสภิต. "ส่วนกริยาภาษาอังกฤษ และสะแลง", กรุงเทพมหานคร: สำนักพิมพ์แพรวพิทยา, 2530.
- มนตรี พจนารถาวณิชย์. "การเขียนโปรแกรมด้วยเทอร์โบซี", กรุงเทพมหานคร: หจก. เอช-เอน การพิมพ์, 2535.
- รองศาสตราจารย์ มัณฑนา ปราการสมุทร. "การเขียนชุดคำสั่งภาษา", กรุงเทพมหานคร: บริษัทควงกมลสมัย จำกัด, 2534.
- ลินดา เจน. "วลีและประโยคตัวอย่าง คำแปล", กรุงเทพมหานคร : โอ. เอส. พรินติ้งเฮาส์, 2533.
- อภิรักษ์ เชิดขวงศ์วณิชย์. "กลุ่มคำกริยาภาษาอังกฤษที่ใช้บ่อย", กรุงเทพมหานคร : สำนักพิมพ์นานมี บุคส์, 2536.
- อาจารย์ สมพงษ์ ภาวจิตร. "หลักการใช้กริยาวลี 555 คำ", กรุงเทพมหานคร: สำนักพิมพ์อักษรวัฒนา, 2536.
- Ben Ezzell. "Graphics programming in turbo C 2.0", 1988.
- Borland International. "Turbo C reference guide version 2.0", U.S.A., 1988.
- Jack Purdum. "C Programmer's toolkit", U.S.A., 1989.
- J. Terry godfrey. "Applied C", Prentice-Hall: 1990.
- Kent Porter. "Stretching Turbo C", New York, 1989.
- M. Tim Grady. "Turbo C Programming Principles and Practice", Mc Graw-Hill: 1989.
- Waite Group Inc. "Ansi C", 1989.