

โปรแกรมروتินการจัดการจอภาพ

(SCREEN MANAGEMENT LIBRARY ROUTINE)

โดย

นายกิตติพันธ์	เจียรพงษ์	28.1025
นายเกรียงไกร	นิศย์นต์	28.1031
นายฉกาจ	อังก์สุธาสาวิทย์	28.1053

อาจารย์ที่ปรึกษา

อ.दनันท์ กิตติพันธ์

วิทยานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

ปีการศึกษา 2531

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ปริญญาโท ประจำปีการศึกษา 2531

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยี พระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมรoutinesในการจัดการจอภาพ

ผู้จัดทำ

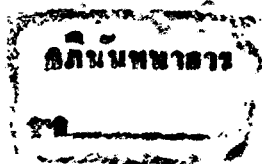
1. นายกิตติพันธ์ เจียรพงษ์ 28.1025

2. นายเกรียงไกร นิคัยนต์ 28.1031

3. นายณภาจ อังค์สุธาสาวิทย์ 28.1053

.....อาจารย์ที่ปรึกษา

(อ.ดเนี้ยววัฒน์ กิตตินันท์)





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

023214

สารบัญ

เรื่อง		หน้า
บทคัดย่อ		
บทที่ 1	บทนำ	1
บทที่ 2	วัตถุประสงค์และขอบเขตของโครงการ	3
	2.1 วัตถุประสงค์ของโครงการ	3
	2.2 ขอบเขตของโครงการ	3
บทที่ 3	การเชื่อมโยงภาษา Assembly กับ ภาษา C	6
	3.1 ลักษณะข้อมูลของภาษา C	6
	3.2 ลักษณะโมเดลของภาษา C	9
	3.3 การกำหนดการเขียนโปรแกรมภาษา Assembly	11
	3.4 การส่งผ่านตัวแปรของภาษา C	13
	3.5 การตั้งชื่อฟังก์ชันของภาษา C	14
	3.6 การกลับไปยังโปรแกรมที่เรียกมา	14
บทที่ 4	ลักษณะการทำงานของโปรแกรม	15
	4.1 หลักการทำงานของโปรแกรม FULLSCREEN MENU แบบ MANUAL	15
	4.2 หลักการทำงานของโปรแกรม FULLSCREEN MENU แบบ AUTO	17
	4.3 หลักการทำงานของโปรแกรม PULLDOWN MENU	19
	4.4 หลักการทำงานของโปรแกรม HELP SCREEN	22
บทที่ 5	ประสิทธิภาพในการนำไปใช้งาน	24
	5.1 ประสิทธิภาพของโปรแกรม	24
	5.2 ตัวอย่างการใช้งาน	25
บทที่ 6	สรุปการทำงานและวิจารณ์	30
ถึติกรรมประกาศ		
เอกสารอ้างอิง		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

		หน้า
รูปที่ 3.1	ลักษณะข้อมูลแบบ string	8
รูปที่ 3.2	ลักษณะโมเดลต่าง ๆ ของภาษา C	10-1 , 10-2
รูปที่ 4.1	PSEUDO CODE ของโปรแกรม FULLSCREEN MENU แบบ MANUAL	15
รูปที่ 4.2	FLOW CHART ของโปรแกรม FULLSCREEN MENU แบบ MANUAL	16
รูปที่ 4.3	PSEUDO CODE ของโปรแกรม FULLSCREEN MENU แบบ AUTO	17
รูปที่ 4.4	FLOW CHART ของโปรแกรม FULLSCREEN MENU แบบ AUTO	18
รูปที่ 4.5	PSEUDO CODE ของโปรแกรม PULLDOWN MENU	19
รูปที่ 4.6	FLOW CHART ของโปรแกรม PULLDOWN MENU	21
รูปที่ 4.7	PSEUDO CODE ของโปรแกรม HELP SCREEN	22
รูปที่ 4.8	FLOW CHART ของโปรแกรม HELP SCREEN	23

สารบัญตาราง

		หน้า
ตารางที่ 3.1	ชนิดของข้อมูลต่าง ๆ ในภาษา C	6-7
ตารางที่ 3.2	การกำหนดรูปแบบต่าง ๆ ในภาษา Assembly	12



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมรฐิการจัการจภาพ

กิตติพัทธ์ เจียรพงษ์

เกรียงไกร นิคยันต์

ฉกาจ อังค์สุธาสาวิทย์

ณัษวัฒน์ กิตตินันท์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2531

บทคัดย่อ

โครงการนี้เป็นการจัดทำ โปรแกรมรฐิการจัการจภาพ ซึ่งจะเน้นให้ผู้เขียนโปรแกรมลดความยุ่งยากและปริมาณในการเขียนโปรแกรมการแสดงผลทางจภาพ โดยจะมีโปรแกรมรฐิการที่มีรูปแบบที่ใช้งานง่ายและสะดวก คอยจัดการการแสดงผลทางจภาพให้ ซึ่งโปรแกรมที่จัดทำขึ้นนี้มีความสามารถใช้งานร่วมกับภาษาระดับสูง และสามารถใช้งานได้กับจภาพประเภทต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SCREEN MANAGEMENT LIBRARY ROUTINE

Kittipat Chearapong

Kriengkai Nissayan

Chakart Angsutasavit

Danaiwat Kittikun Advisor

1988

Abstract

This project is on creating screen management library routine. It emphasizes on reducing complication and volume of program. By the easy and convenience using routine for screen management that is able to use with a high level language and widely using with so many kind of monitor.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในการใช้งานคอมพิวเตอร์ ส่วนแรกที่จะเป็นการเลือกรูปแบบให้การทำงานระหว่างผู้ใช้กับคอมพิวเตอร์ คือ การแสดงทางจอภาพ ซึ่งผู้ใช้จะสามารถใช้งานคอมพิวเตอร์ได้อย่างมีประสิทธิภาพภาพนั้น ก็ขึ้นอยู่กับว่ารูปแบบการแสดงผลออกทางจอภาพของโปรแกรมนั้น สามารถทำให้ผู้ใช้เข้าใจได้มากเพียงไร ดังนั้นในการเขียนโปรแกรมของผู้พัฒนาโปรแกรม ส่วนสำคัญส่วนแรกที่จะต้องคำนึงถึงคือ รูปแบบของหน้าจอของคอมพิวเตอร์ที่จะทำให้ผู้ใช้สามารถทำความเข้าใจ และใช้โปรแกรมได้ง่าย

จากความสำคัญของส่วนที่กล่าวมานี้ ทางผู้จัดทำจึงได้จัดทำโครงงานนี้ขึ้น เพื่อลดความยุ่งยากและประหยัดเวลาในการเขียนโปรแกรมในส่วนการขึ้นจอภาพ โดยโครงงานนี้ได้ร่วมรูปแบบของการแสดงทางจอภาพตามความต้องการของผู้ใช้และผู้เขียนโปรแกรม

ปฏิญานี้ฉบับนี้ แบ่งเนื้อหาออกเป็น 4 ส่วนใหญ่ ๆ คือ

ส่วนที่ 1 ประกอบด้วย บทที่ 1 , 2

กล่าวถึง บทนำ วัตถุประสงค์ และขอบเขตของโครงงาน

ส่วนที่ 2 ประกอบด้วย บทที่ 3

กล่าวถึง การเชื่อมโยงภาษา Assembly กับ ภาษา C ตั้งแต่ลักษณะข้อมูลของภาษา C ลักษณะโมเดลของภาษา C การกำหนดการเขียนโปรแกรมภาษา Assembly การส่งผ่านตัวแปรของภาษา C การตั้งชื่อฟังก์ชันของภาษา C จนถึงการกลับไปยังโปรแกรมที่เรียกมา

ส่วนที่ 3 ประกอบด้วย บทที่ 4 , 5

กล่าวถึง ลักษณะการทำงานของโปรแกรม ตั้งแต่ โปรแกรมของ FULLSCREEN MENU ทั้งแบบ AUTO และ แบบ MANUAL โปรแกรมของ PULLDOWN MENU โปรแกรมของ HELPSCREEN จนถึง ประสิทธิภาพในการนำไปใช้งาน และตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างการใช้งาน
ส่วนที่ 4 ประกอบด้วย บทที่ 6
กล่าวถึง สรุปรการทำงานและวิจารณ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

วัตถุประสงค์และขอบเขตของโครงการ

2.1 วัตถุประสงค์ของโครงการ

โปรแกรมรฐทินจัดการจอภาพที่จัดทำขึ้นนี้ เพื่อวัตถุประสงค์หลักดังนี้ คือ

1. เพื่อช่วยให้ผู้เขียนโปรแกรมในส่วนที่เกี่ยวข้องกับจอภาพ ลดขั้นตอนความยุ่งยากในการเขียนโปรแกรม และช่วยลดปริมาณโปรแกรมในการเขียนโปรแกรมการทำงานอย่างเดียวกัน
2. เพื่อช่วยให้ผู้เขียนโปรแกรมประหยัดเวลาในการเขียนโปรแกรมจัดการการแสดงผลทางจอภาพ สามารถนำเวลาที่เหลือไปพัฒนาส่วนอื่น ๆ ต่อไปได้
3. เพื่อช่วยให้ผู้เขียนโปรแกรมมีความสะดวกในการเขียนโปรแกรมจัดการการแสดงผลทางจอภาพ
4. เพื่อสนับสนุนการแสดงผลทางจอภาพของภาษาระดับสูง โดยโปรแกรมที่จัดทำขึ้นนี้จะ เป็น โปรแกรมที่ใช้ร่วมกับภาษาระดับสูง จึงเป็นการสนับสนุนการทำงานด้านการแสดงผลทางจอภาพของภาษาระดับสูงด้วย

2.2 ขอบเขตของโครงการ

ในการแสดงผลทางจอภาพ จะมีรูปแบบต่าง ๆ มากมาย ซึ่งในโครงการนี้เป็นโครงการที่มีขอบเขตการทำงานภายใน 1 ภาคการศึกษา โดยเป็นการจัดทำโปรแกรมรฐทินจัดการการแสดงผลทางจอภาพ สามารถแบ่งเป็นหัวข้อต่าง ๆ ดังนี้

1. เมนู (MENU)

เป็นโปรแกรมรoutinesจัดการการแสดงผลบนจอภาพ โดยจัดเป็นเมนูที่สามารถเลือกการทำงานต่าง ๆ ในโครงงานนี้เป็นการจัดทำเมนูรูปแบบต่าง ๆ ซึ่งแบ่งได้ คือ

1.1 FULL SCREEN MENU

เป็นการแสดงเมนูการทำงานที่มีรูปแบบการขึ้นจอภาพแบบหน้าจอเดียว โดยหน้าที่การทำงานทั้งหมดจะปรากฏในหน้าจอเดียว การแสดงผลแบบ FULL SCREEN MENU นี้ทางผู้จัดทำได้จัดทำโปรแกรมออกเป็น 2 แบบ คือ

- แบบ AUTO ซึ่งโปรแกรมรoutinesจัดการจอภาพจะทำการกำหนดตำแหน่งการแสดงผลบนจอภาพให้
- แบบ MANUAL โปรแกรมรoutinesนี้จะให้ผู้เขียนโปรแกรมทำการกำหนดตำแหน่งการแสดงผลบนจอภาพเอง

1.2 PULL DOWN MENU

เป็นการแสดงเมนูการทำงานซึ่งมีรูปแบบการขึ้นจอภาพ เป็นการแสดงหน้าที่การทำงานปรากฏบนหลาย ๆ หน้าจอ โดยเลื่อนไปกลับระหว่างหน้าจอได้

2. HELP SCREEN

เป็นการแสดงผลทางจอภาพ โดยโปรแกรมรoutinesที่จัดทำขึ้นนี้ จะนำข้อความที่เก็บในเท็กซ์ไฟล์ (Text file) มาทำการแสดงผลบนจอภาพที่มีการกำหนดขนาดของจอภาพในการแสดงข้อความได้

3. หน้าที่มีเศษอื่น ๆ ที่เกี่ยวข้องกับจอภาพ

เป็นส่วนที่โปรแกรมรู้จักในการจัดการจอภาพสามารถจัดการด้านการแสดงผลต่างๆ ซึ่งจัดแบ่งตามหน้าที่ได้ ดังนี้

3.1 การแสดงผล (DISPLAY)

แบ่งเป็นการแสดงผลรูปแบบต่าง ๆ คือ

- การแสดงผลข้อความทั้งแบบกำหนดความยาวของข้อความได้ และแบบที่โปรแกรมกำหนดความยาวให้

- การแสดงผลที่เป็นตัวอักษร

3.2 การลบหน้าจอ (CLEAR)

เป็นการจัดการทางด้านการลบหน้าจอภาพ โดยสามารถกำหนดขอบเขตของหน้าจอที่จะทำการลบออกได้

3.3 การเก็บบันทึก (SAVE)

เป็นการจัดการทางด้านการเก็บบันทึก โดยสามารถเลือกเก็บบันทึกหน้าจอแต่ละหน้าจอได้

3.4 การดึงข้อความกลับ (PICK)

เป็นการจัดการทางด้านการดึงข้อความของแต่ละหน้าจอที่ได้เก็บบันทึกไว้แล้ว มาทำการแสดงบนหน้าจออีก ซึ่งสามารถกำหนดขอบเขตของจอภาพที่จะแสดงผลได้

3.5 การแสดงหน้าต่าง (WINDOW)

เป็นการจัดการหน้าจอการแสดงผล โดยจัดทำเป็นหน้าต่างการแสดงผล ซึ่งสามารถกำหนดขอบเขตของหน้าต่าง, กำหนดชนิดของหน้าต่างได้

บทที่ 3

การเชื่อมโยงภาษา Assembly กับภาษา C

ลักษณะการเชื่อมโยงโปรแกรมที่เขียนด้วยภาษา ASSEMBLY เข้ากับภาษา C สามารถทำได้ไม่ยาก แต่ในการที่จะติดต่อกับภาษา C ต้องทราบถึงลักษณะข้อมูล , โครงสร้าง และ การส่งค่าตัวแปรของภาษา C ก่อน จึงสามารถทำการติดต่อกันได้

3.1 ลักษณะข้อมูลของภาษา C

ภาษา C มีลักษณะข้อมูลอยู่ 4 แบบ คือ

- ข้อมูลแบบ Character
- ข้อมูลแบบ Integer
- ข้อมูลแบบ Float
- ข้อมูลแบบ Pointer

ซึ่งในแต่ละแบบ ยังสามารถแบ่งย่อยได้อีก ซึ่งรายละเอียดของข้อมูลดูได้จาก ตารางที่ 3.1

ตารางที่ 3.1

ชนิดของข้อมูล	ความยาว (บิต)	ขอบเขต
unsigned char	8	0 - 255
char	8	-128 - 127
enum	16	-32768 - 32767
unsigned short	16	0 - 65535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดของข้อมูล	ความยาว (บิต)	ขอบเขต
short	16	-32768 - 32767
unsigned int	16	0 - 65535
int	16	-32768 - 32767
unsigned long	32	0 - 4294967295
long	32	-2147483648 - 2147483647
float	32	3.4E-38 - 3.4E+38
double	64	1.7E-308 - 1.7E+308
long double	64	1.7E-308 - 1.7E+308
pointer	16	(near, _cs, _ds, _es, _ss pointers)
pointer	32	(far, huge pointers)

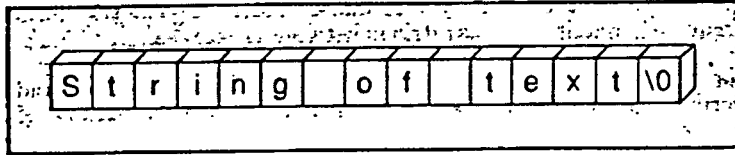
ตารางที่ 3.1 แสดงลักษณะตัวแปรของภาษา C

นอกจากนี้ ยังมีลักษณะข้อมูลแบบอื่น ๆ อีก คือ

ข้อมูลแบบ String

ข้อมูลแบบ string เป็นข้อมูลแบบของ Character ที่เรียงต่อกันแล้วปิดท้ายด้วย Character Null (chr 0) ซึ่งมีลักษณะ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แสดงลักษณะข้อมูลแบบ string

ข้อมูลแบบ Array

ข้อมูลแบบ Array ของ ภาษา C ก็มีลักษณะแบบเดียวกับภาษาอื่น ๆ แต่การเริ่มต้นของ Array จะเริ่มที่ 0 จนกระทั่งถึงตำแหน่งตัวสุดท้ายลบหนึ่ง (0, 1, 2, ..., n-1) เช่น arr[3] จะมีลักษณะ ดังนี้ คือ arr[0], arr[1], arr[2]

การกำหนด Array หลายมิติ มีการเก็บแบบที่เรียกว่า row-column order เช่น arr[3][2] มีการเก็บ คือ

- arr[0][0]
- arr[0][1]
- arr[1][0]
- arr[1][1]
- arr[2][0]
- arr[2][1]



3.2. ลักษณะโมเดลของภาษา C

การกำหนดโมเดลภาษา C มีการกำหนดโมเดลออกเป็น 6 แบบ ทำให้มีผลต่อการเขียนโปรแกรมด้านรูปแบบการเขียน และ การส่งค่าตัวแปรซึ่งลักษณะของโมเดลทั้ง 6 แบบ มีดังนี้ คือ

3.2.1 โมเดลแบบ Tiny

โมเดลของ Tiny Model มีการกำหนดเซ็กเมนต์ทั้งหมด (CS,DS,SS,ES) อยู่ที่ตำแหน่งเดียวกัน ทำให้ขนาดของโปรแกรมนั้นมีขนาดไม่เกิน 64 กิโลไบต์ การกำหนดตัวแปรแบบ pointer เป็นแบบ near การกำหนดโมเดลแบบนี้ทำให้เปลี่ยนโปรแกรมที่ทำการ compile แล้วเป็นในรูป .COM format ได้

3.2.2 โมเดลแบบ Small

การกำหนดโมเดลแบบ Small Model มีการกำหนดเซ็กเมนต์ของ Code และ Data แยกกันและไม่ซ้อนทับกัน ทำให้ขนาดของ Code และ static Data มีขนาดอย่างละ 64 กิโลไบต์ ส่วน Extra และ Stack Segment อยู่ตำแหน่งเดียวกับ Data Segment ส่วนการกำหนดตัวแปรแบบ pointer เป็นแบบ near เช่นเดียวกับแบบ Tiny การใช้โมเดลแบบนี้นิยมใช้กันเป็นส่วนมาก

3.2.3 โมเดลแบบ Medium

การกำหนดโมเดลแบบนี้ ขนาดของ Code มีขนาด 1 เมกะไบต์ ส่วนการกำหนดขนาดของ Data มีขนาด 64 กิโลไบต์ ทำให้มีการใช้ Pointer ของ Code เป็นแบบ Far และ Pointer ของ Data เป็นแบบ Near โมเดลแบบนี้ใช้สำหรับขนาดของโปรแกรมใหญ่แต่มีการใช้ข้อมูลน้อย

3.2.4 โมเดลแบบ Compact

การกำหนดโมเดลแบบนี้ ตรงข้ามกับโมเดลแบบ Medium ซึ่งขนาดของ Data มีขนาด 1 เมกะไบต์ ส่วนการกำหนดขนาดของ Code มีขนาด 64 กิโลไบต์ ทำให้มีการใช้ Pointer ของ Data เป็นแบบ Far และ Pointer ของ Code เป็นแบบ Near โมเดลแบบนี้ใช้สำหรับขนาดของโปรแกรมเล็กแต่มีการใช้ข้อมูลมาก

3.2.5 โมเดลแบบ Large

การกำหนดชนิดตัวแปร Pointer ของโมเดลนี้เป็นแบบ Far ทั้ง Code และ Data แต่ขนาดของ static data ยังกำหนดขนาดแค่ 64 กิโลไบต์

3.2.6 โมเดลแบบ Huge

โมเดลแบบนี้เหมือนกับแบบ Large แต่ขนาดของ static data มีขนาดใหญ่กว่า 64 กิโลไบต์

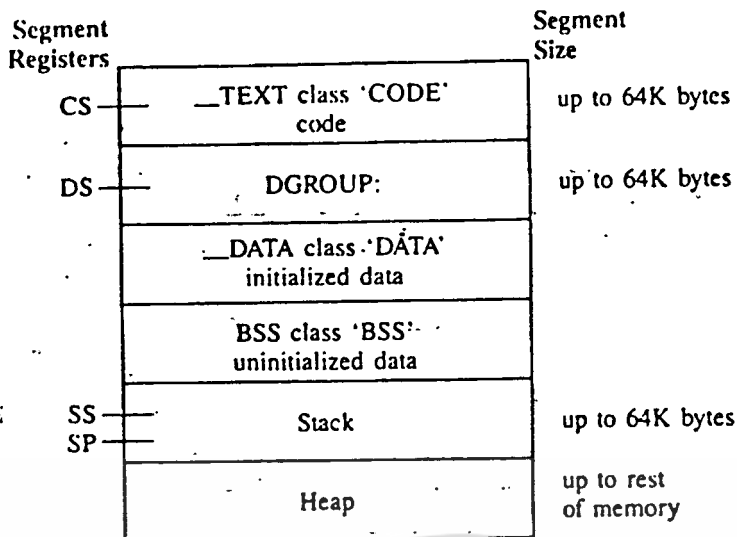


Figure 9.5: Compact Model Memory Segmentation

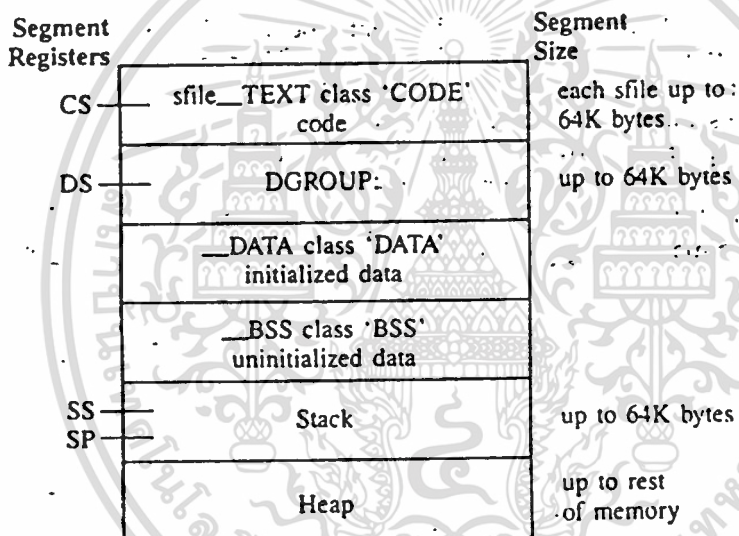


Figure 9.6: Large Model Memory Segmentation

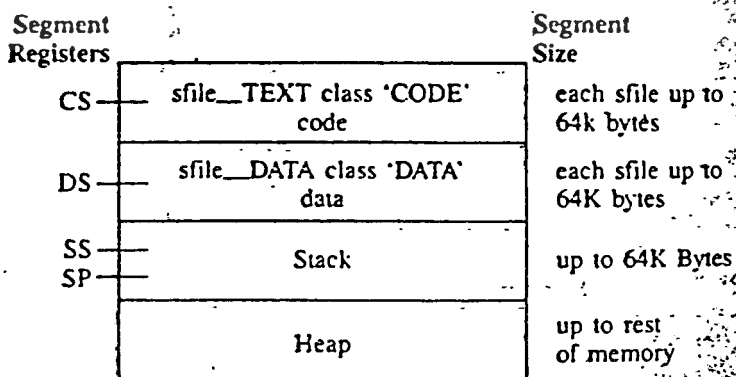


Figure 9.7: Huge Model Memory Segmentation

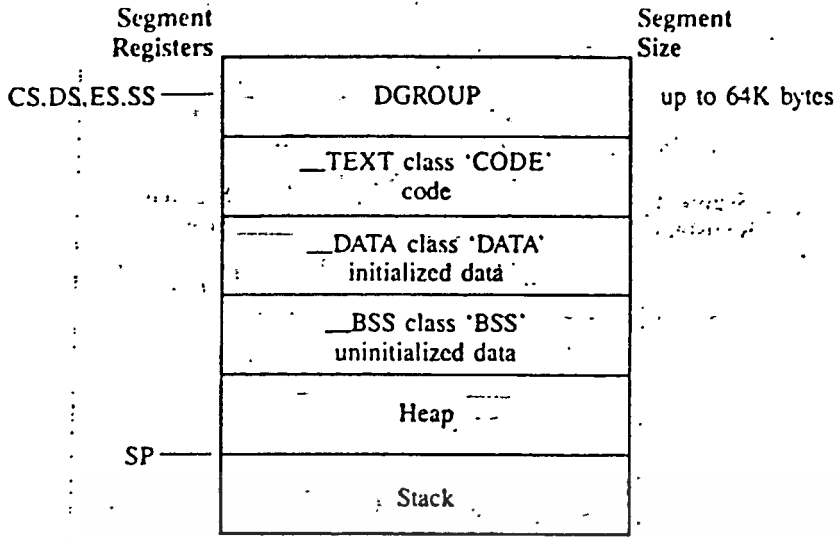


Figure 9.2: Tiny Model Memory Segmentation

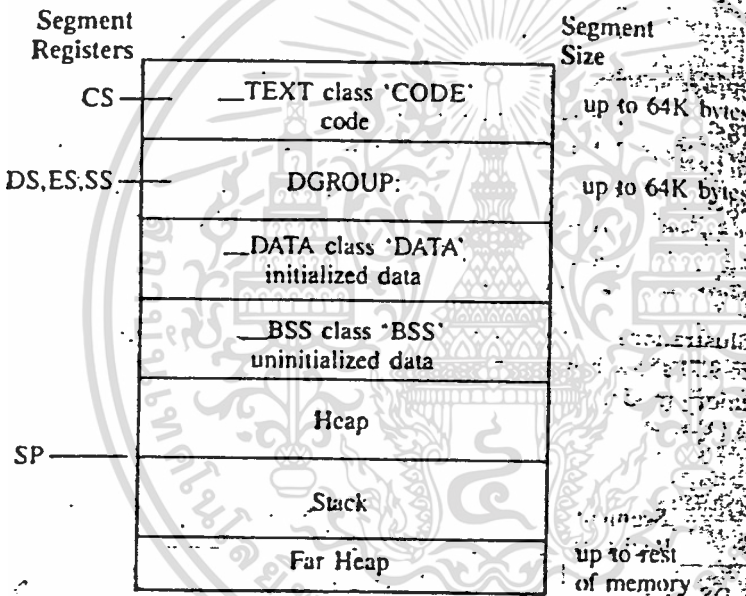


Figure 9.3: Small Model Memory Segmentation

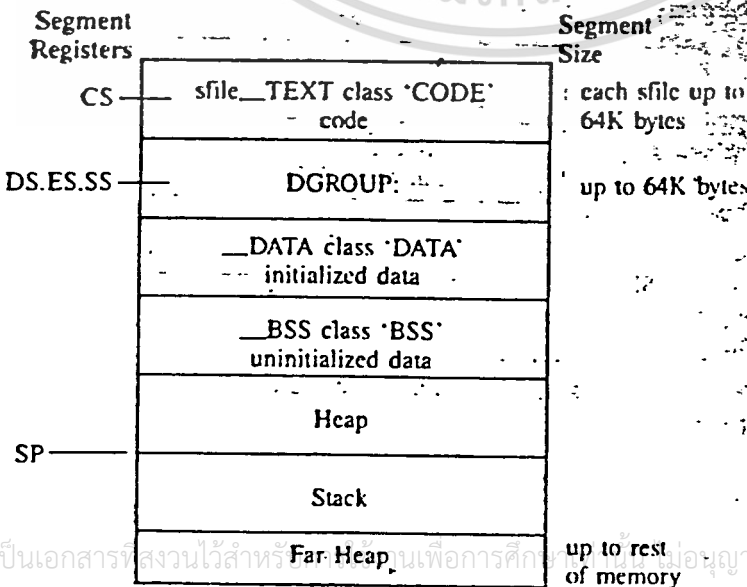


Figure 9.4: Medium Model Memory Segmentation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ Far Heap การศึกษาเพื่อการใช้งานที่มีอยู่ภายใต้เงื่อนไขการใช้งานไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ที่สืบ หรือที่พบในข้อต่อของเอกสาร และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การกำหนดการเขียนโปรแกรมภาษา Assembly

การกำหนดลักษณะของโปรแกรมภาษา ASSEMBLY ที่เรียกมาโดยภาษา C มีลักษณะดังนี้ คือ

```
< text >      BYTE      PUBLIC  'CODE'
               ASSUME    CS:< text >,DS:< dseg >
               <.....Code Segment.....>
< text >      ENDS
< dseg >      GROUP    _DATA,_BSS
< data >      SEGMENT  WORD      PUBLIC  'DATA'
               <.....initielized data segment.....>
< data >      ENDS
_BSS          SEGMENT  WORD      PUBLIC  'BSS'
               <.....uninitialized data segment.....>
_BSS          ENDS
               END
```

คำรอง code,data,dseg ดูได้จาก ตาราง 3.2

Model	ข้อกำหนดที่ใส่แทนที่	Code and Data Pointers
Tiny,	<code> = _TEXT	Code: DW _TEXT:xxx
Small	<data> = _DATA <dseg> = DGROUP	Data: DW DGROUP:xxx
Compact	<code> = _TEXT <data> = _DATA <dseg> = DGROUP	Code: DW _TEXT:xxx Data: DD DGROUP:xxx
Medium	<code> = filename_TEXT <data> = _DATA <dseg> = DGROUP	Code: DD _TEXT:xxx Data: DW DGROUP:xxx
Large	<code> = filename_TEXT <data> = _DATA <dseg> = DGROUP	Code: DD _TEXT:xxx - Data: DD DGROUP:xxx
Huge	<code> = filename_TEXT <data> = filename_DATA <dseg> = filename_DATA	Code: DD xxx Data: DD xxx

ตารางที่ 3.2 แสดงการกำหนดต่างๆในภาษา assembly

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การส่งผ่านตัวแปรของภาษา C

การส่งผ่านตัวแปรของภาษา C ใช้การส่งผ่านค่าตัวแปรทาง Stack โดยค่าที่เก็บอยู่ใน Stack จะถูกเก็บลงจากขวาไปซ้าย คือ ค่าที่อยู่บนสุดของ Stack เป็นตัวทางซ้ายสุดแล้วโปรแกรมภาษา C จะเก็บเอา Return Address เก็บตามลงมา ค่า Return Address นั้น จะมีขนาดเท่าไรนั้นขึ้นอยู่กับชนิดของโมเดลที่กำหนดว่า Code Pointer เป็นแบบ Far หรือ Near ตัวอย่างเช่น

Function(v1,v2)

ลักษณะ Stack ของ Near Pointer

ลักษณะ Stack ของ Far Pointer

SP : Ret Address

SP : Ret Address

SP + 2 : v1

SP + 2 : Ret Segment

SP + 4 : v2

SP + 4 : v1

SP + 6 : v2

การส่งค่ากลับของภาษา C

การส่งค่ากลับของภาษา C มีลักษณะดังนี้ คือ

- การส่งค่ากลับแบบ 16 บิต ในการกำหนดค่าตัวแปรขนาด 2 ไบต์ ต่าง ๆ ที่ส่งกลับ (Char, int, short, enum, near pointer) จะต้องส่งค่ากลับผ่านรีจิสเตอร์ AX
- การส่งค่ากลับแบบ 32 บิต ในการกำหนดค่าตัวแปรขนาด 4 ไบต์ ต่าง ๆ ที่ส่งกลับ รวมทั้ง Far และ Huge Pointer จะต้องส่งค่ากลับผ่านรีจิสเตอร์ DX และ AX โดยรีจิสเตอร์ DX เก็บเลขนัยสำคัญสูง และ AX เก็บเลขนัยสำคัญต่ำ

- การส่งค่ากลับของตัวแปรแบบ Float ใช้การส่งค่าผ่าน 8087 top-of-stack (TOS) register หรือที่เรียกว่า ST(0) ถ้าใช้โปรแกรมจำลอง 8087 การส่งค่าจะส่งในรีจิสเตอร์ที่จำลอง TOS Register

3.5 การตั้งชื่อฟังก์ชันของภาษา C

การกำหนดชื่อฟังก์ชันในภาษา Assembly ที่เรียกมาจากภาษา C โดยภาษา C มีการกำหนดชื่อฟังก์ชันเป็นตัวเล็กและจะมีการเติมตัว underbar (_) ที่ข้างหน้าฟังก์ชันดังนั้น การกำหนดชื่อฟังก์ชันในภาษา Assembly ต้องมีการกำหนดเป็นตัวเล็กและให้โปรแกรม แอสเซมเบอร์ มีการป้องกันการแปลงชื่อฟังก์ชันเป็นตัวใหญ่ และต้องกำหนดชื่อฟังก์ชันในภาษา Assembly ที่ขึ้นต้นด้วยสัญลักษณ์ underbar โดยที่ฟังก์ชันนั้นในภาษา C ไม่ต้องขึ้นต้นด้วย underbar เนื่องจากคอมไพเลอร์มีการจัดการให้เสร็จ

3.6 การกลับไปยังโปรแกรมที่เรียกมา

การกลับไปยังโปรแกรมภาษา C ที่เรียกมานั้นสามารถใช้คำสั่ง RET ได้โดยตรง โดยไม่ต้องมีการลบล้างสแต็กที่ส่งค่ามาเหมือนภาษาอื่น ๆ เนื่องจากโปรแกรมที่เรียกมานั้นจะมีการล้างสแต็กให้ซึ่งถ้ามีการล้างสแต็กโดยตัวฟังก์ชันเป็นตัวจัดการจะทำให้โปรแกรมมีการทำงานที่ผิดพลาดได้

บทที่ 4

ลักษณะการทำงานของโปรแกรม

4.1 หลักการทำงานของ PROGRAM FULLSCREEN MENU แบบ MANUAL

START	
READ PARAMETER FROM PROGRAM CALL	; อ่านค่าพารามิเตอร์จากโปรแกรม
DISPLAY MENU	; ขึ้นเมนูตามตำแหน่งที่ผู้ใช้กำหนด
READ KEY PRESS	; อ่านค่าการกดคีย์จากผู้ใช้
RETURN VALUE KEY TO PROGRAM CALL	; ส่งค่ากลับไปยังโปรแกรมที่เรียกมา
STOP	

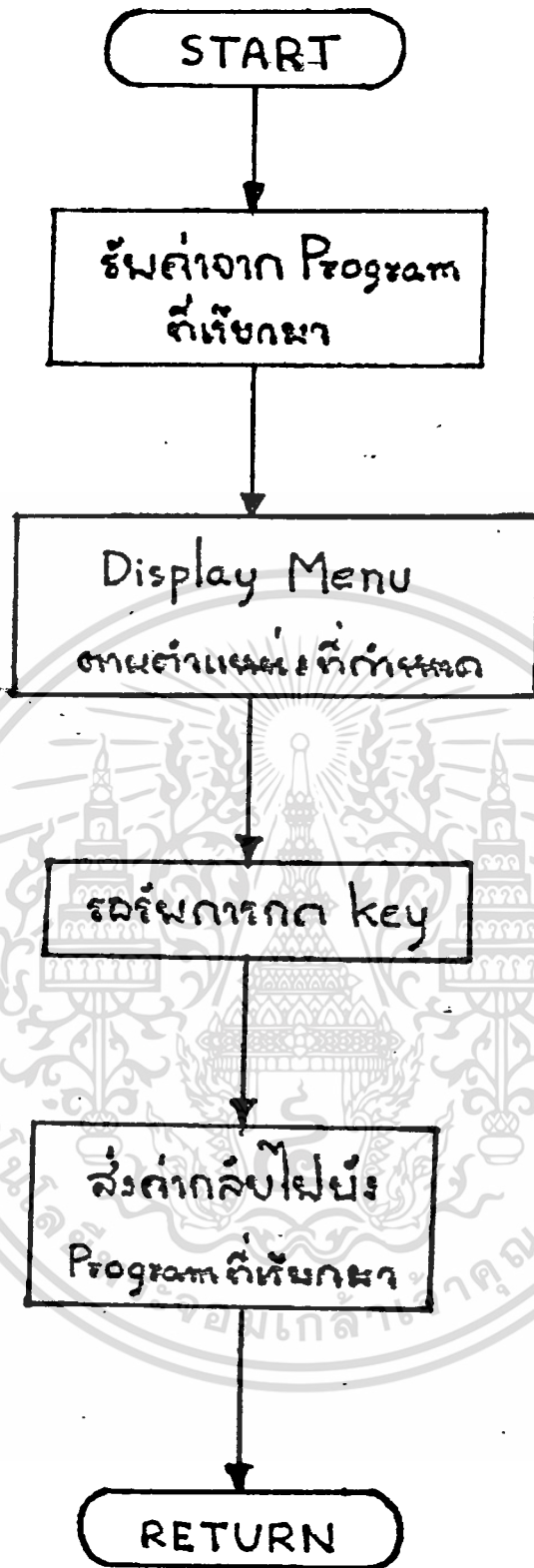
รูปที่ 4.1 แสดง Pseudo-code ของโปรแกรมรoutines FULLSCREEN MENU แบบ MANUAL

จากรูปที่ 4.1 และรูปที่ 4.2 เป็นการแสดง PSEUDO CODE และ FLOWCHART ของรoutinesการจัดการการขึ้นจอภาพแบบ FULL SCREEN MENU แบบ MANUAL

หลักการทำงานของรoutinesแบบนี้ก็คือ

1. เมื่อมีการเรียกใช้รoutinesของการขึ้นเมนูแบบเต็มจอ แบบที่ผู้ใช้กำหนดการขึ้นตำแหน่งของเมนูเองนี้แล้ว ก็จะมีการอ่านค่าพารามิเตอร์ต่างๆ จากโปรแกรมที่เรียกใช้มา
2. จากนั้นก็จะทำการขึ้นเมนูแบบเต็มจอ ตามตำแหน่งที่ผู้ใช้กำหนดให้ โดยค่าที่จะขึ้นในเมนู ก็จะเป็นค่าที่ถูกส่งมาจากโปรแกรมที่เรียกมาด้วย
3. ขั้นตอนต่อไปหลังจากขึ้นเมนูแล้ว ก็จะรอให้ผู้ใช้ทำการกดคีย์บอร์ด เพื่อรับค่าที่ผู้ใช้เลือกจากเมนู
4. โปรแกรมรoutinesจะทำการส่งค่าที่ผู้ใช้เลือก ผ่านกลับไปยังโปรแกรมที่เรียกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



4.2 หลักการทำงานของ PROGRAM FULLSCREEN MENU แบบ AUTO

START

READ PARAMETER FROM PROGRAM CALL ; อ่านค่าจากโปรแกรมที่เรียก

CALCULATE POSITION OF MENU ; คำนวณตำแหน่งของการขึ้นเมนู

DISPLAY MENU ; ขึ้นเมนูตามตำแหน่งที่คำนวณได้

READ KEY PRESS ; อ่านค่าที่ผู้ใช้เลือก

RETURN VALUE KEY TO PROGRAM CALL ; ส่งค่ากลับไปยังโปรแกรมที่เรียก

STOP

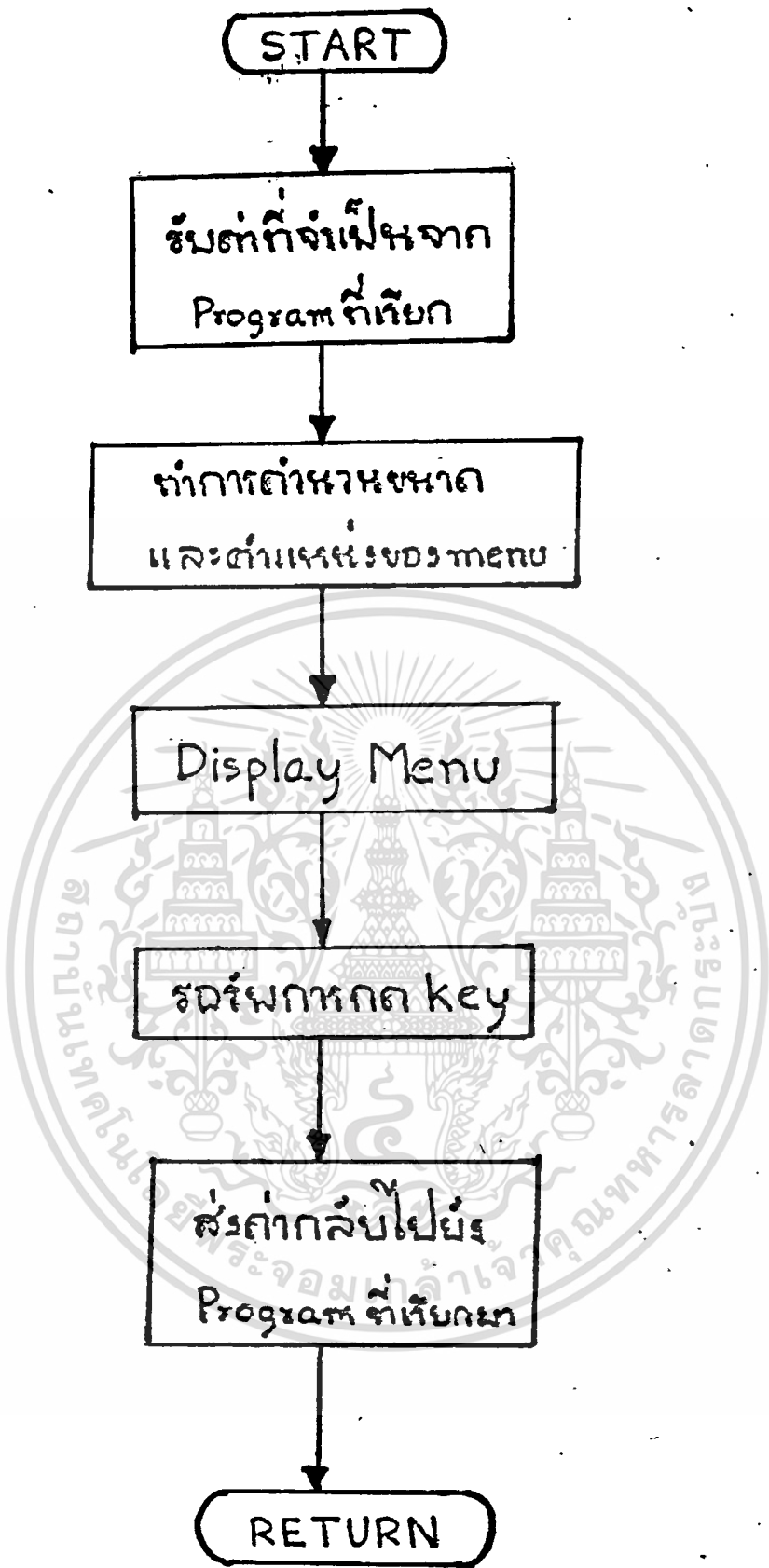
รูปที่ 4.3 แสดง Pseudo-code ของโปรแกรมรoutines FULLSCREEN MENU แบบ AUTO

จากรูปที่ 4.3 และรูปที่ 4.4 เป็นการแสดง PSEUDO CODE และ FLOWCHART ของรoutines การจัดการการขึ้นจอภาพแบบ FULL SCREEN MENU แบบ AUTO

หลักการทำงานของรoutines แบบนี้ก็คือ

1. เมื่อมีการเรียกใช้รoutines ของการขึ้นเมนูแบบเต็มจอ แบบ AUTO ซึ่งโปรแกรม จะทำการคำนวณตำแหน่งของการขึ้นเมนูให้ ก็จะมีการอ่านค่าพารามิเตอร์ต่างๆ จากโปรแกรมที่เรียกใช้มา
2. โปรแกรมจะทำการคำนวณตำแหน่งของการขึ้นเมนู โดยจะดูจาก parameter ที่ผู้ใช้กำหนดให้
3. จากนั้นก็จะทำการขึ้นเมนูแบบเต็มจอ ตามตำแหน่งที่คำนวณได้ โดยค่าที่จะขึ้นในเมนู ก็จะเป็นค่าที่ถูกส่งมาจากโปรแกรมที่เรียกมาด้วย
4. ขั้นตอนต่อไปหลังจากขึ้นเมนูแล้ว ก็จะรอให้ผู้ใช้ทำการกดคีย์บอร์ด เพื่อรับค่าที่ผู้ใช้เลือกจากเมนู
5. โปรแกรมรoutines จะทำการส่งค่าที่ผู้ใช้เลือก ผ่านกลับไปยังโปรแกรมที่เรียกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 FULLSCREEN MENU แบบ AUTO

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ ห้ามเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสาร
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทำงานของ PROGRAM PULL-DOWN MENU

START

READ PARAMETER FROM PROGRAM CALL ; อ่านค่า parameter

DO WHILE KEY <> RETURN ; รับค่าการกดคีย์และแสดงเมนูจนกว่า
จะมีการกด RETURN

DISPLAY MAIN CHOICE ; แสดงตัวเลือกหลัก

IF HAS SUB CHOICE ; ถ้าตัวเลือกหลักนั้นมีตัวเลือกย่อย

DISPLAY SUB CHOICE ; แสดงตัวเลือกย่อยนั้น

ENDIF

READ KEY PRESS ; อ่านการกดคีย์ของผู้ใช้

ENDDO

RETURN VALUE TO PROGRAM CALL ; ส่งค่าที่ผู้ใช้เลือกกลับไป

STOP

รูปที่ 4.5 แสดง Pseudo-code ของโปรแกรมรูกิน PULL-DOWN MENU

จากรูปที่ 4.5 และรูปที่ 4.6 เป็นการแสดง PSEUDO CODE และ FLOWCHART ของรูกินการจัดการการขึ้นจอภาพแบบ FULL SCREEN MENU แบบ AUTO

หลักการทำงานของรูกินแบบนี้ก็คือ

1. เมื่อมีการเรียกใช้รูกินของการขึ้นเมนูแบบเต็มจอ แบบ AUTO ซึ่งโปรแกรม จะทำการคำนวณตำแหน่งของการขึ้นเมนูให้ ก็จะมีการอ่านค่าพารามิเตอร์ต่างๆ จากโปรแกรมที่เรียกใช้มา

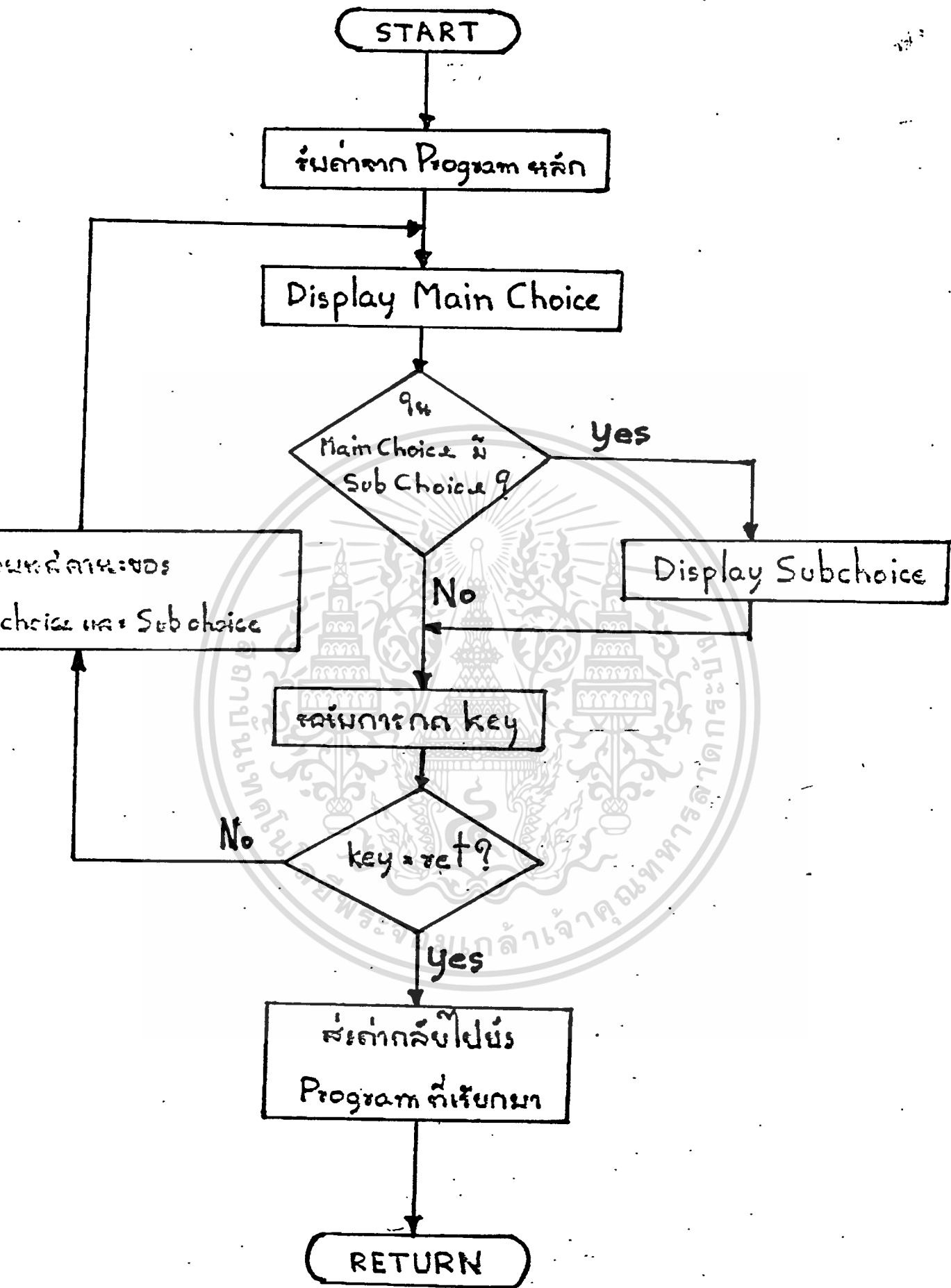
2. โปรแกรมจะทำการคำนวณตำแหน่งของการขึ้นเมนู โดยจะดูจาก parameter ที่ผู้ใช้กำหนดให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จากนั้นก็จะทำการขึ้นเมนูแบบเต็มจอ ตามตำแหน่งที่คำนวณได้ โดยค่าที่จะขึ้นในเมนู ก็จะเป็นค่าที่ถูกส่งมาจากโปรแกรมที่เรียกมาด้วย
4. ขั้นตอนต่อไปหลังจากขึ้นเมนูแล้ว ก็จะรอให้ผู้ใช้ทำการกดคีย์บอร์ด เพื่อรับค่าที่ผู้ใช้เลือกจากเมนู
5. โปรแกรมมรฐินจะทำการส่งค่าที่ผู้ใช้เลือก ผ่านกลับไปยังโปรแกรมที่เรียกมา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.6 PULLDOWN MENU

4.4 การทำงานของ PROGRAM HELP SCREEN

START

READ PARAMETER FROM PROGRAM CALL ; อ่านค่า parameter

DISPLAY WINDOW ; แสดงกรอบตามตำแหน่งที่รับมา

READ FILE ; อ่าน file ที่ต้องการแสดง

DO WILE KEY <> ESCAPE ; เมื่อยังไม่มีการกด ESCAPE

DISPLAY DATA IN WINDOW ; แสดงข้อมูลที่อยู่ใน file

READ KEY ; อ่านคีย์

CHANGE PAGE ; เปลี่ยนหน้าจอของข้อมูลใน file

ENDDO

RETURN

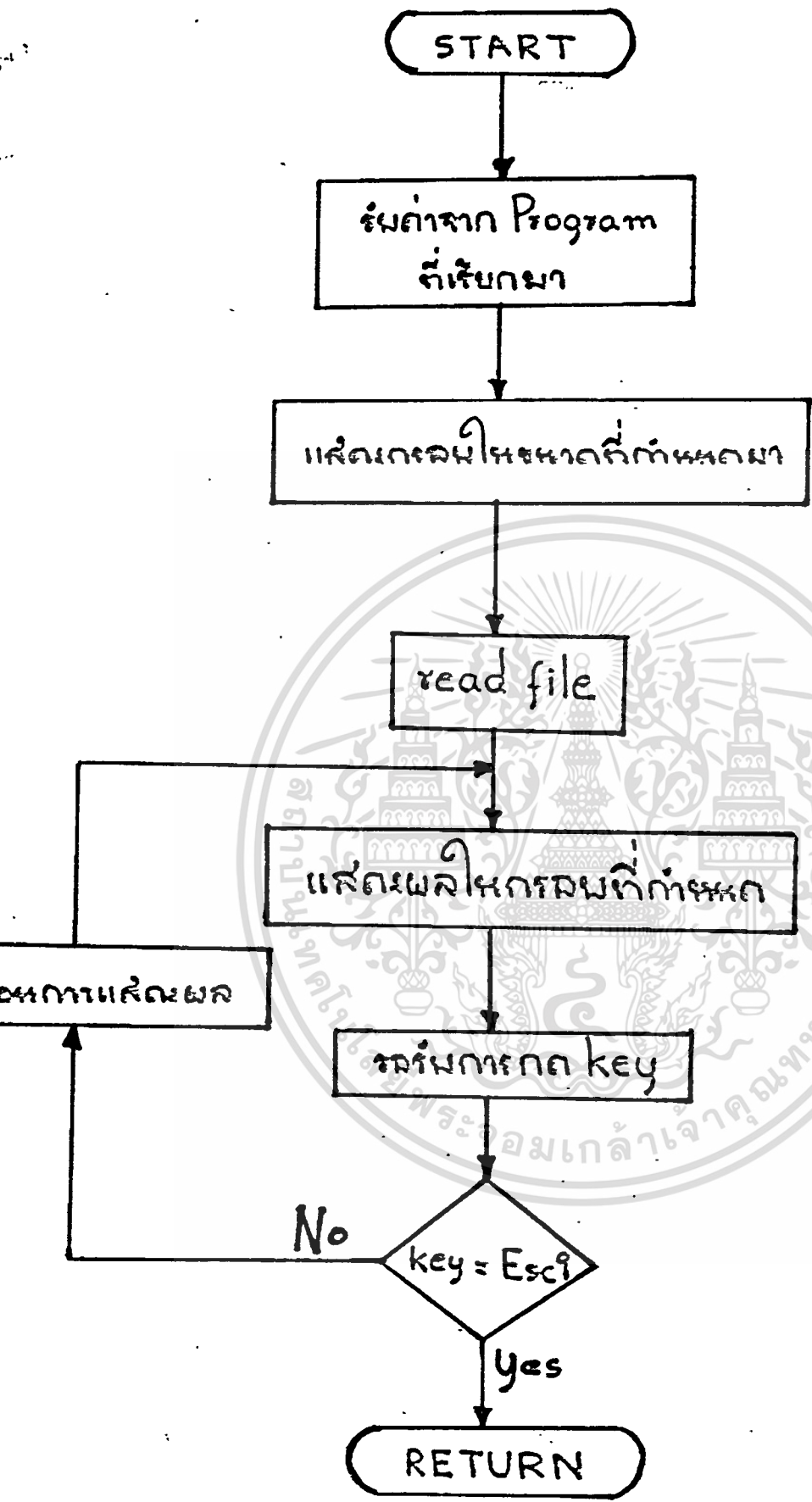
รูปที่ 4.7 แสดง Pseudo-code ของโปรแกรมรoutines HELP SCREEN

จากรูปที่ 4.7 และรูปที่ 4.8 เป็นการแสดง PSEUDO CODE และ FLOWCHART ของรoutinesการจัดการการขึ้นจอภาพแบบ HELP SCREEN

หลักการทำงานของรoutinesแบบนี้ก็คือ

1. เมื่อมีการเรียกใช้รoutinesของ help screen ซึ่งโปรแกรม ก็จะมีการอ่านค่าพารามิเตอร์ต่างๆ จากโปรแกรมที่เรียกใช้มา
2. โปรแกรมจะทำการขึ้นกรอบ ตามตำแหน่ง parameter ที่ผู้ใช้กำหนดให้
3. จากนั้นก็จะทำการอ่าน file ที่จะทำการแสดงขึ้นมา
4. แสดงข้อมูลที่อยู่ใน file ที่ต้องการให้อยู่ในกรอบที่กำหนด
5. รอการกดคีย์ของผู้ใช้ในการเปลี่ยนหน้าจอ
6. เมื่อมีการกด ESCAPE ก็จะออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามเผยแพร่ข้อมูลนี้ไปยังบุคคลอื่นโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.๘ HELP SCREEN

บทที่ 5

ประสิทธิภาพในการนำไปใช้งาน

5.1 ประสิทธิภาพของโปรแกรม

โปรแกรมสำหรับการจัดการจอภาพที่จัดทำขึ้นนี้ มีประสิทธิภาพในการนำไปใช้งาน คือ

1. สามารถนำไปใช้งานร่วมกับจอภาพได้ทุกประเภท โดยโปรแกรมนี้จะทำการตรวจสอบลักษณะของจอภาพที่ใช้งานอยู่ และทำการปรับการทำงานของโปรแกรมให้เข้ากับลักษณะของจอภาพนั้น เพื่อให้สามารถทำงานร่วมกันได้
2. สามารถลดรูปแบบการเขียนโปรแกรมส่วนแสดงผลไปได้มาก
3. มีความสามารถในการแสดงผล โดยแยกเป็นแต่ละหัวข้อ ดังนี้

3.1 FULL SCREEN MENU

ทั้งแบบ AUTO และ แบบ MANUAL มีความสามารถในการแสดงผล ดังนี้

- ขนาดของการแสดงผลทางจอภาพ สามารถกำหนดได้สูงสุด 25 * 80 ตัวอักษร
- จำนวนตัวเลือก สามารถกำหนดได้สูงสุด 18 ตัวเลือก

3.2 PULL DOWN MENU

มีความสามารถในการแสดงผล ดังนี้

- จำนวนตัวเลือกหลัก (MAIN CHOICE) สามารถกำหนดได้สูงสุด 7 ตัวเลือก โดยแต่ละตัวเลือกหลักมีจำนวนตัวอักษรสูงสุด 10 ตัวอักษร
- จำนวนตัวเลือกย่อย (SUB CHOICE) สามารถกำหนดได้สูงสุด 20 ตัวเลือก โดยแต่ละตัวเลือกย่อยมีจำนวนตัวอักษรสูงสุดไม่เกินขนาดของกรอบที่แสดงผล

3.3 HELP SCREEN

มีความสามารถในการแสดงผล ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขนาดของไฟล์ที่นำมาแสดงผลบนหน้าจอ จะขึ้นอยู่กับขนาดของไฟล์ที่เก็บอยู่ในอุปกรณ์เก็บข้อมูล (STORAGE) เช่น ขนาดของไฟล์ที่เก็บอยู่มีขนาด 1024 ไบต์ โปรแกรมในส่วน HELP SCREEN นี้ก็สามารถแสดงผลได้ทั้งหมด 1024 ไบต์เช่นกัน

3.4 หน้าที่พิเศษอื่น ๆ

มีความสามารถในการแสดงผล ดังนี้

- สามารถทำการแสดงผลแบบข้อความ และแบบตัวอักษร โดยแบบข้อความมีทั้งกำหนดความยาวของข้อความได้ และโปรแกรมกำหนดความยาวให้
- สามารถทำการลบหน้าจอ โดยกำหนดขอบเขตของหน้าจอที่จะทำการลบได้
- สามารถทำการเก็บบันทึก โดยเลือกหน้าจอที่จะทำการบันทึกได้
- สามารถทำการดึงข้อความกลับมาแสดงผลบนหน้าจออีก โดยกำหนดขอบเขตของจอภาพที่จะแสดงผลได้
- สามารถสร้างหน้าต่างการแสดงผล โดยกำหนดขอบเขต , ชนิดของหน้าต่างได้

5.2 ตัวอย่างการใช้งาน

- FULL SCREEN MENU MENU AUTO

ลักษณะรูปแบบการเรียกใช้ คือ

`FULL_MENU_AUTO (CHOICE, CHOICE_SELECT, RET)`

ดังตัวอย่าง

```
MENU = "MENU,5,1 INSERT,2 DELETE,3 EDIT,4 DISPLAY,5 QUIT"  
CHOICE = "SELECT ---> "  
FULL_MENU_AUTO (MENU,CHOICE,RET)
```

การเรียกใช้ให้แสดงผล FULL SCREEN MENU แบบ AUTO นี้ จะต้องมีการกำหนดรูปแบบการขึ้นจอภาพของตัวเลือกต่าง ๆ ในตัวแปรก่อน เช่น จากตัวอย่าง มีการกำหนดลักษณะการขึ้นจอภาพในตัวแปรชื่อ MENU ที่ประกอบด้วยตัวเลือก 5 ตัวดังกล่าว , ตัวแปรชื่อ CHOICE ที่มีข้อความให้ทำการเลือก หลังจากกำหนดลักษณะการขึ้นจอภาพในตัวแปรแล้ว จึงทำการเรียกใช้คำสั่งการแสดงผล FULL SCREEN MENU แบบ AUTO ได้ ตัวแปรชื่อ RET เป็นค่าที่ส่งกลับมายังโปรแกรมหลักของผู้เขียนโปรแกรม เพื่อให้ทราบว่ามีการเลือกตัวเลือกใด

- FULL SCREEN MENU แบบ MANUAL

ลักษณะรูปแบบการเรียกใช้คือ

```
FULL_MENU_MAN (CHOICE,CHOICE_SELECT,RET)
```

ดังตัวอย่าง

```
MENU = "1,35,MENU,70,5,3,10,1 INSERT,2 DELETE,3 EDIT,4  
DISPLAY,5 QUIT"  
CHOICE = "20,10,SELECT--> "  
FULL_MENU_MAN (MENU,CHOICE,RET)
```

การเรียกใช้ให้แสดงผล FULL SCREEN MENU แบบ MANUAL นี้ ก็ต้องมีการกำหนดรูปแบบการขึ้นจอภาพของตัวเลือกต่าง ๆ ในตัวแปรก่อนเช่นเดียวกัน แต่แบบ MANUAL นี้จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องมีการกำหนดจุดเริ่มต้นของข้อความที่จะแสดงผล , Attribute ของการแสดงผล เช่น จากตัวอย่าง ตัวแปรชื่อ MENU มีการกำหนดจุดเริ่มต้นของ MENU คือ 1,35 Attribute คือ 7๑ , จุดเริ่มต้นของตัวเลือกคือ 3,1๐ , ตัวแปรชื่อ CHOICE มีการกำหนดจุดเริ่มต้นของข้อความคือ 2๐,1๐ หลังจากนั้นจึงมีการเรียกใช้ให้ทำการแสดงผล FULL SCREEN MENU แบบ MANUAL โดยตัวแปร RET ก็เช่นเดียวกับแบบ AUTO

- PULL DOWN MENU

ลักษณะรูปแบบการเรียกใช้ คือ

PULL_DOWN (MAIN_CHOICE,RET,SUB1,...,SUBn)

ดังตัวอย่าง

```
MAIN_C = "MENU,3,FILE,EDIT,DISPLAY"
```

```
FILE = "5,LOAD,SAVE,NEW,WRITE TO,QUIT"
```

```
EDIT = "๑"
```

```
DISPLAY = "3,LIST ALL,LIST,BROWSE"
```

```
PULL_DOWN (MAIN_C,RET,FILE,EDIT,DISPLAY)
```

การเรียกใช้การแสดงผล PULL DOWN MENU นี้ จะต้องมีการกำหนดรูปแบบการขึ้นจอภาพในตัวแปรก่อน ซึ่งจะประกอบด้วยตัวเลือกหลัก ตัวเลือกย่อย เช่น จากตัวอย่าง ตัวแปร MAIN_C เป็นตัวแปรที่แสดงให้ทราบว่า MENU นี้ประกอบด้วยตัวเลือกหลักคือ FILE,EDIT, DISPLAY ซึ่งแต่ละตัวเลือกหลักนี้จะประกอบด้วยตัวเลือกย่อยอีก ดังรายละเอียดในตัวอย่าง สำหรับตัวแปร RET ก็เป็นตัวแปรที่ใช้ในลักษณะเช่นเดียวกับ FULL SCREEN MENU

- HELP SCREEN

ลักษณะรูปแบบการเรียกใช้คือ

HELP (ROW1, COL1, ROW2, COL2, FILENAME)

ดังตัวอย่าง

```
FILENAME_1 = "HELP.TXT"
```

```
HELP (10, 10, 20, 60, FILENAME_1)
```

การเรียกใช้การแสดงผล HELP SCREEN นี้ จะต้องมีการกำหนดขนาดของจอภาพ ที่ต้องการให้ข้อความปรากฏ และมีการกำหนดชื่อไฟล์ที่ต้องการนำมาแสดงผล เช่น จากตัวอย่าง ไฟล์ที่นำมาแสดงบนหน้าจอชื่อ HELP.TXT โดยมีขนาดของการแสดงผลตามที่กำหนด

- หน้าทีพิเศษอื่น ๆ เกี่ยวกับจอภาพ

ลักษณะรูปแบบการเรียกใช้คือ

```
DISPLAY (ROW, COL, STRING, ATTRIBUTE, PAGE)
```

```
DISP (ROW, COL, STRING, ATTRIBUTE, PAGE, LENGTH)
```

```
DISP_CHAR (ROW, COL, STRING, ATTRIBUTE, PAGE)
```

```
CLEAR (ROW1, COL1, ROW2, COL2, ATTRIBUTE, PAGE)
```

```
SAVE (PAGE)
```

```
PICK (ROW1, COL1, ROW2, COL2, PAGE)
```

```
WINDOW (ROW1, COL1, LEN_ROW, LEN_COL, KIND_OF WINDOW,
```

```
ATTRIBUTE, PAGE)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตั้งตัวอย่าง

DISPLAY (10,10,"HELLO",70,0)

เป็นการแสดงผลข้อความ "HELLO" บนจอภาพโดยมีจุดเริ่มต้นที่ 10,10 มี
ATTRIBUTE = 70 (REVERSE) ที่ PAGE 0



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปการทำงานและวิจารณ์

จากการจัดทำโครงงานนี้ ในส่วนที่ทางผู้จัดทำได้ทำเสร็จสมบูรณ์แล้ว คือ การแสดงผลทางจอภาพ โดยเป็นการขึ้น MENU ในรูปแบบที่เป็น FULLSCREEN MENU ทั้งแบบ AUTO และแบบ MANUAL , PULLDOWN และ HELPSCREEN ซึ่งรูปแบบต่าง ๆ นี้เป็นส่วนที่ใช้งานร่วมกับภาษา C นอกจากนี้ยังได้จัดทำหน้าที่พิเศษอื่น ๆ ที่เกี่ยวกับการแสดงผลทางจอภาพ ในการทดลองนำไปใช้งานพบว่า โปรแกรมที่จัดทำนี้สามารถใช้งานได้เป็นอย่างดี ตรงตามวัตถุประสงค์ที่ได้กำหนดไว้ ซึ่งถ้าได้มีการพัฒนาต่อให้สามารถใช้งานร่วมกับภาษาระดับสูงอื่น ๆ อีก และเพิ่มเติมรายละเอียดการแสดงผลทางจอภาพให้มีความสามารถมากขึ้น ก็จะเป็นโปรแกรมหุ่นการจัดการจอภาพที่สามารถนำไปใช้งานได้อย่างมีประสิทธิภาพ

กิติกรรมประกาศ

ขอขอบคุณ อาจารย์ที่ปรึกษา อ.ศนิวัฒน์ กิตตินันท์ เพื่อน ๆ ตลอดจนผู้ที่ให้ความช่วยเหลือ คำแนะนำ และกำลังใจ ตลอดระยะเวลาที่ได้จัดทำวิทยานิพนธ์ฉบับนี้ ซึ่งทำให้การจัดทำวิทยานิพนธ์ฉบับนี้ เสร็จลุล่วงไปได้ด้วยดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. Microsoft , "Microsoft Macro Assembler Programmer's Guide" ,
Microsoft Corporation , 401 p. , 1987
2. Borland , "Turbo C User guide and Reference guide" , Borland
International Inc , 685 p. , 1987



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้