

## แนวคิดและทฤษฎี

### แนวคิด

โปรแกรมจอภาพภาษาอาร์พีจีที่ใช้ในการแสดงผลบนเครื่องไอบีเอ็ม S/36 มีส่วนประกอบ 2 ส่วนคือ

1) โปรแกรมภาษาอาร์พีจี ใช้ในการควบคุมการทำงานต่างๆ รวมถึงการแสดงผลของจอภาพ

2) ข้อกำหนดของจอภาพ ใช้ในกำหนดข้อความที่ใช้ในการแสดงผล โดยเริ่มทำงานเมื่อได้รับคำสั่งจากโปรแกรม และส่งผลกลับไปยังโปรแกรม ในการเขียนโปรแกรมและข้อกำหนดของจอภาพกระทำแยกจากกัน มีความสัมพันธ์กันเฉพาะการรับส่งข้อมูลเท่านั้น การปรับปรุงแก้ไขจอภาพมีผลกระทบต่อารรับส่งข้อมูลระหว่างโปรแกรมและจอภาพ และในการเปลี่ยนแปลงแก้ไขโปรแกรมการหาจุดที่ต้องแก้ไขทำได้ลำบาก

### ทฤษฎี

1. **ข้อมูล** คือ คำที่ใช้อธิบาย หรือข้อความที่กล่าวถึงสิ่งใดสิ่งหนึ่ง ข้อความนั้นอาจเป็นตัวเลข ตัวอักษร หรือสัญลักษณ์อื่นๆ มีรายละเอียดดังนี้

1.1 **ชนิดของข้อมูล (Data type)** โดยทั่วไปมีการแบ่งชนิดของข้อมูลออกเป็น 3 ประเภท

- 1) ชนิดตัวอักษร (Alphabetic) เป็นข้อมูลที่มีเฉพาะตัวอักษร
- 2) ชนิดตัวเลข (Numeric) เป็นข้อมูลที่มีเฉพาะตัวเลข 0-9
- 3) ชนิดตัวอักษรเลข (Alphanumeric) เป็นข้อมูลที่มีทั้งตัวเลขและตัวอักษร รวมถึงสัญลักษณ์พิเศษด้วย

1.2 **เขตข้อมูล (field)** คือ กลุ่มของอักขระ ตั้งแต่ 1 อักขระขึ้นไปที่น่ามาต่อเนื่องกัน แล้วหมายถึงสิ่งหนึ่งสิ่งใด

2. ภาษาอาร์พีจี ลักษณะการทำงานของภาษาอาร์พีจี มีการทำงานเป็นวงจรซึ่งเรียกว่า “วงจรอาร์พีจี (RPG Cycle)” มีรายละเอียดดังนี้

2.1 ส่วนประกอบของภาษาอาร์พีจี มีรายละเอียดดังนี้

1) การทำงานก่อนเข้าวงจร ทำโดยโปรแกรมย่อยที่เรียกว่า House Keeping routine การทำงานเป็นไปโดยอัตโนมัติ งานที่ทำได้แก่ การเปิดแฟ้มข้อมูล สร้างตัวแปร

2) ส่วนรับข้อมูลเข้า (Input area) เป็นส่วนที่อ่านข้อมูลเข้า ใช้เก็บข้อมูลที่ได้จากการอ่าน เปรียบเทียบข้อมูล รวมถึงการตรวจสอบการสิ้นสุดของแฟ้มข้อมูล

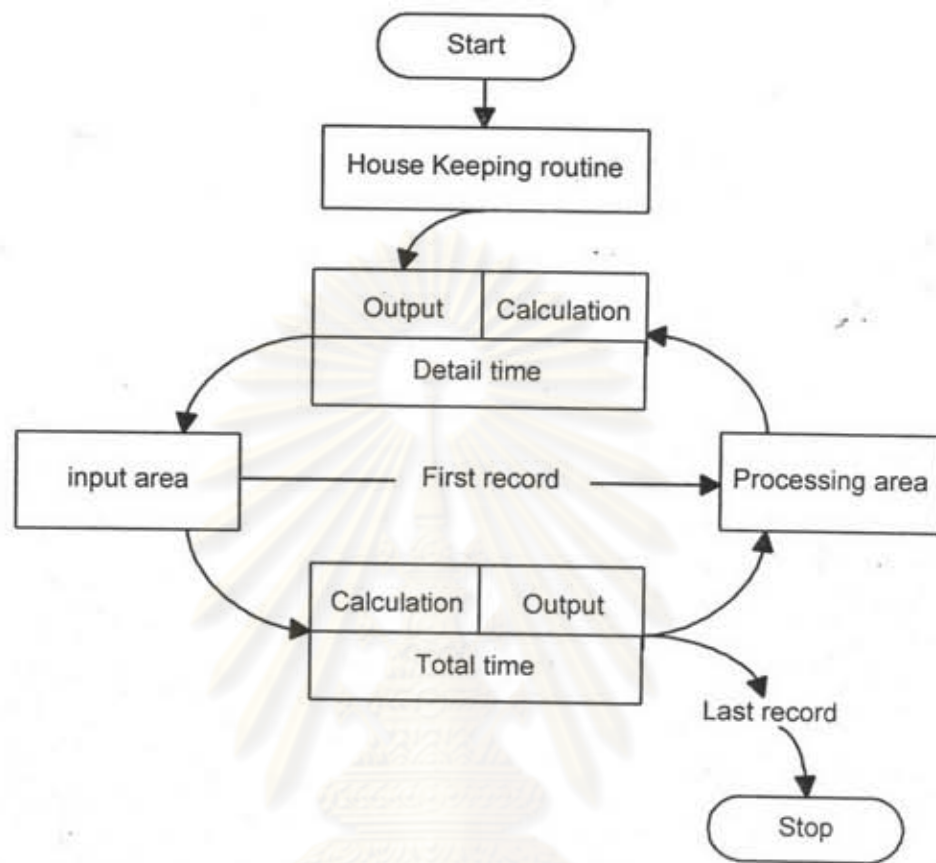
3) ส่วนเก็บข้อมูลสำหรับการประมวลผล (Processing area) ใช้เก็บข้อมูลที่ได้จากส่วนพักข้อมูลนำเข้า เพื่อใช้ในโปรแกรม

4) ส่วนการทำงานจังหวะรายละเอียด (Detail time) ใช้ในการประมวลผลข้อมูล ผู้ใช้สามารถโปรแกรมให้ทำงานได้ มีส่วนประกอบ 2 ส่วน คือ ส่วนที่ใช้ในการประมวลผลและส่วนการแสดงผล การทำงานเป็นไปตามลำดับ คือ ทำการประมวลผลทั้งหมดก่อนแล้วจึงทำการแสดงผล

5) ส่วนการทำงานจังหวะรวม (Total time) ใช้ในการประมวลผลข้อมูลหลังจากที่ได้อ่านข้อมูลตัวใหม่เข้ามา แต่การประมวลผลจะใช้ข้อมูลเดิมที่อยู่ในส่วนเก็บข้อมูลสำหรับการประมวลผล ผู้ใช้สามารถโปรแกรมให้ทำงานได้ มีส่วนประกอบเช่นเดียวกับส่วนการทำงานจังหวะรายละเอียด

6) ส่วนจบการทำงาน (Last record) เป็นการตรวจสอบการสิ้นสุดของวงจรอาร์พีจี

ส่วนประกอบต่างๆของภาษาอาร์พีจีแสดงได้ด้วยรูปดังนี้



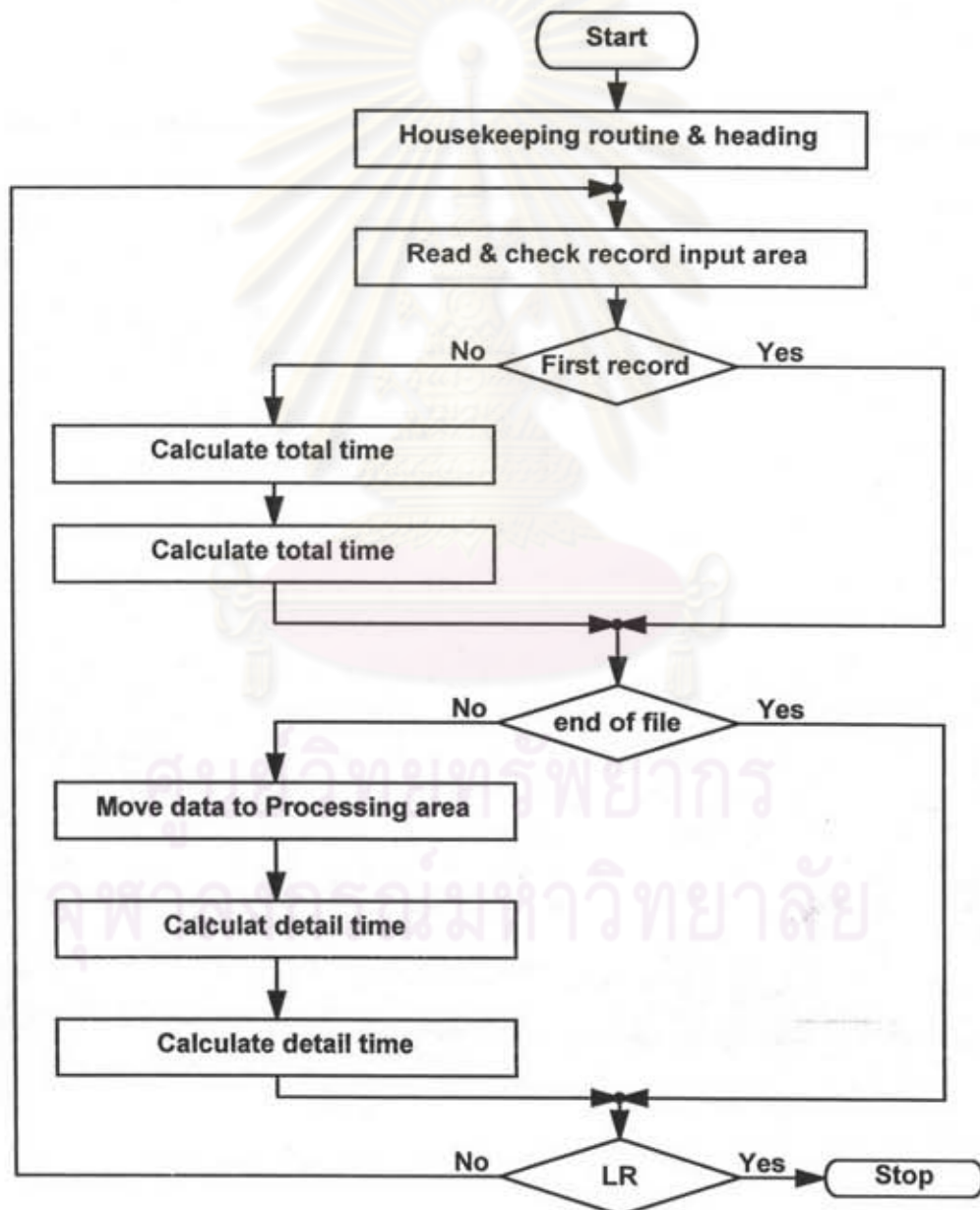
รูปที่ 2.1 แสดงส่วนประกอบของวงจรอาร์พีจี

2.2 การทำงานของวงจรอาร์พีจี การทำงานแบ่งออกเป็น 2 จังหวะ คือ จังหวะรวมและจังหวะรายละเอียด ในแต่ละจังหวะการทำงานจะมีการทำงาน 2 ชนิด คือ ส่วนการคำนวณ และส่วนการนำข้อมูลออก การทำงานของวงจรอาร์พีจีมีดังนี้

- 1) ส่วนการทำงานก่อนเข้าวงจร ทำการเปิดแฟ้มข้อมูลกำหนดตัวแปรและตัวชี้บอก (Indicator) พร้อมกับแสดงผลส่วนของการนำข้อมูลออก
- 2) อ่านแฟ้มข้อมูลและนำผลที่ได้เก็บในส่วนรับข้อมูลเข้า
- 3) ตรวจสอบว่าเป็นข้อมูลตัวแรกหรือไม่ ถ้าใช่จะข้ามการทำงานในส่วนจังหวะรวมไป
- 4) ทำการประมวลผลในส่วนของจังหวะรวม
- 5) ทำการแสดงผลในส่วนของจังหวะรวม
- 6) ตรวจสอบว่าเป็นจบแฟ้มข้อมูลหรือไม่ ถ้าใช่จะข้ามการทำงานในส่วนจังหวะรายละเอียดไป

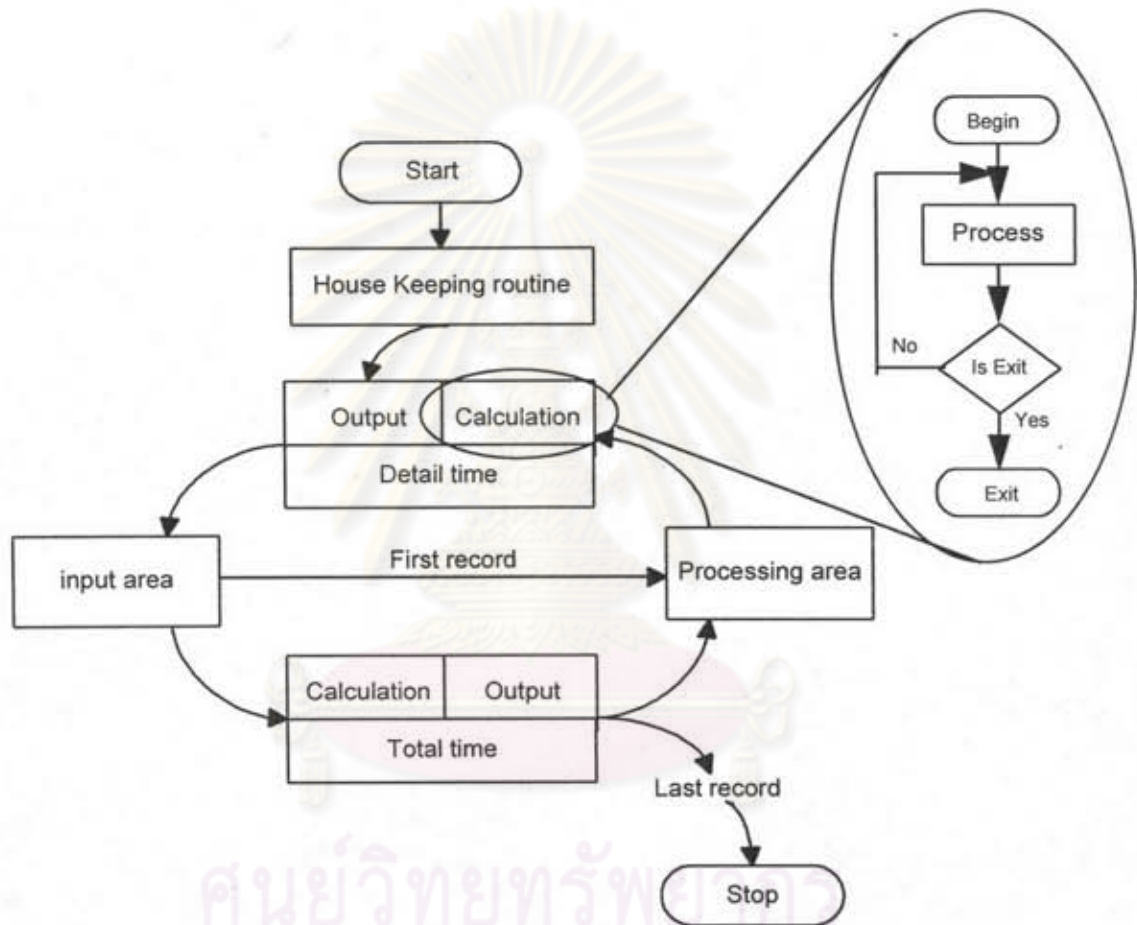
- 7) ทำการย้ายข้อมูลในส่วนรับข้อมูลเข้า ไปเก็บไว้ในส่วนเก็บข้อมูล  
สำหรับการประมวลผล
- 8) ทำการประมวลผลในส่วนของจังหวัดรายละเอียด
- 9) ทำการแสดงผลในส่วนของจังหวัดรายละเอียด
- 10) ทำการตรวจสอบว่าตัวชี้บอกแสดงการจบเพิ่มข้อมูลเป็นจริงหรือไม่ ถ้าไม่ใช่จะข้ามไปเริ่มอ่านเพิ่มข้อมูลตัวถัดไป

การทำงานสามารถแสดงได้ด้วยผังงานดังนี้



รูปที่ 2.2 ผังงานแสดงการทำงานของวงจรรารีพีซี

2.3 การควบคุมวงจรรหัสอาร์พีซีโดยผู้เขียนโปรแกรม จากการทำงานของวงจรรหัสอาร์พีซีที่ผู้เขียนโปรแกรมสามารถเข้าควบคุมการทำงานได้ในส่วนของจังหวะรายละเอียด โดยที่ผู้เขียนโปรแกรมเป็นผู้ควบคุมการทำงานได้ตลอดเวลาจนกว่าจะจบการทำงาน การนำข้อมูลเข้าออกจะทำได้โดยการใช้คำสั่งทำให้อ่านหรือเขียนเพิ่มข้อมูล การทำงานสามารถแสดงได้ดังรูป



รูปที่ 2.3 แสดงการทำงานของภาษาอาร์พีซีที่ควบคุมโดยผู้เขียนโปรแกรม

2.4 การตั้งชื่อในภาษาอาร์พีซี ในภาษาอาร์พีซีมีกฎในการตั้งชื่อตัวแปร, เซตข้อมูล, เพิ่มข้อมูลดังนี้

- 1) เป็นตัวอักษรตัวใหญ่หรือตัวเลข
- 2) ขึ้นต้นเป็นตัวอักษร
- 3) ตัวแปรหรือเซตข้อมูลมีความยาวไม่เกิน 6 ตัวอักษร และชื่อเพิ่มข้อมูลมีความยาวไม่เกิน 8 ตัวอักษร

2.5 **แบบฟอร์มของภาษาอาร์พีจี** แบบฟอร์มของภาษาอาร์พีจีที่ใช้กันอยู่มี 7 แบบคือ

2.5.1 **แบบฟอร์ม H (Control Specification)** ใช้ระบุชื่อของโปรแกรม และใช้กำหนดการควบคุมอื่นๆ เช่น กำหนดรูปแบบวันที่ กำหนดการตรวจสอบข้อผิดพลาด แบบฟอร์มนี้จะเป็นแบบฟอร์มแรกสำหรับโปรแกรมภาษาอาร์พีจี มีรายละเอียดดังนี้

- 1) ชนิด ใช้แสดงชนิดของแบบฟอร์ม
- 2) การตรวจสอบข้อผิดพลาด ใช้ในการกำหนดให้มีการตรวจสอบข้อผิดพลาดด้วยคำสั่ง DEBUG หรือไม่
- 3) ชนิดของวันที่ที่ใช้ ถ้าเป็น "M" จะเป็นรูปแบบเดือนวันปี, "D" จะเป็นรูปแบบ วันเดือนปี และ "Y" สำหรับรูปแบบ ปีเดือนวัน
- 4) ชื่อของโปรแกรม ใช้ระบุให้ตัวแปลภาษา เก็บผลจากการแปลไว้ที่ชื่อที่ระบุ

2.5.2 **แบบฟอร์ม F (File Description Specification)** ใช้ในการกำหนดรายละเอียดเกี่ยวกับแฟ้มข้อมูลแต่ละแฟ้มที่ใช้ในโปรแกรม เช่น ชื่อ ประเภทของแฟ้มข้อมูล ความยาวของข้อมูล ตลอดจนวิธีการในการได้มาซึ่งข้อมูล มีรายละเอียดดังนี้

- 1) ชนิด ใช้แสดงชนิดของแบบฟอร์ม
- 2) ชื่อแฟ้มข้อมูล ใช้บอกชื่อของแฟ้มข้อมูลที่ใช้
- 3) ชนิดของแฟ้มข้อมูล ใช้ระบุการใช้งานของแฟ้มข้อมูลว่าเป็นแบบ นำเข้า(Input) ส่งออก(Output) ปรับปรุง(Update) หรือ ผสม(Combined)
- 4) การทำงานของแฟ้มข้อมูล ใช้ระบุการทำงานของแฟ้มข้อมูล เช่น การทำงานแบบนำเข้าขั้นต้น(Primary) แบบผสม(Chain) แบบตามความต้องการ(Demand) และแบบสามารถทำได้หลายรูปแบบ(Full procedural )

- 5) ความยาวของระเบียบของแฟ้มข้อมูล
- 6) วิธีการทำงานของแฟ้มข้อมูล ใช้ระบุการทำงานเป็นแบบสุ่ม หรือเรียงลำดับ

- 7) ความยาวของดัชนี ในกรณีที่มีการใช้ดัชนี
- 8) ชนิดของตัวชี้ตำแหน่งของระเบียบ
- 9) ระบุว่าแฟ้มข้อมูลมีการใช้ดัชนีหรือไม่ สำหรับแฟ้ม

แบบมีดัชนี

- 10) ตัวชี้บอก ใช้สำหรับควบคุมจุดสิ้นสุดของหน้าสำหรับ

งานพิมพ์

- 11) ตำแหน่งเริ่มต้นของดัชนี ใช้สำหรับแฟ้มที่มีดัชนี

12) มีการใช้แบบฟอร์มเพิ่มเติม สำหรับการนับบรรทัดของเครื่องพิมพ์หรือไม่หรือส่วนเพิ่มเติมส่วนของแฟ้มข้อมูลชนิดของตาราง

13) ชนิดของแฟ้มข้อมูล เช่น เครื่องพิมพ์ (PRINTER), จานแม่เหล็ก (DISK), จอภาพ (WORKSTN)

14) การต่อบรรทัด ในกรณีที่มีการกำหนดรายละเอียดเพิ่มเติม

15) กำหนดชื่อของอุปกรณ์ เช่น ชื่อของภาษาจอภาพ

16) กำหนดให้สามารถเพิ่มข้อมูลได้หรือไม่

2.5.3 **แบบฟอร์ม L (Line counter Specification)** ใช้กำหนดบรรทัดในการพิมพ์ มีรายละเอียดดังนี้

- 1) ชนิด ใช้แสดงชนิดของแบบฟอร์ม
- 2) ชื่อของเครื่องพิมพ์ที่ระบุในแบบฟอร์ม F
- 3) จำนวนบรรทัดที่ใช้ใน 1 หน้า
- 4) บรรทัดที่กำหนดให้ขึ้นหน้าใหม่

2.5.4 **แบบฟอร์ม E (Extension Specification)** ใช้กำหนดรายละเอียดเกี่ยวกับตัวแปรชุด มีรายละเอียดดังนี้

- 1) ชนิด ใช้แสดงชนิดของแบบฟอร์ม
- 2) ชื่อของตัวแปรชุด
- 3) จำนวนข้อมูลต่อ 1 ระเบียบ
- 4) จำนวนสูงสุดของระเบียบ
- 5) ความยาวของระเบียบ
- 6) จำนวนทศนิยม ใช้สำหรับกรณีที่เป็นตัวเลข

ในการกำหนดตัวแปรชุด ภายในจะระบุไว้ท้ายของโปรแกรมโดยมี \*\* บอกจุดเริ่มต้น ส่วนข้อมูลในแต่ละระเบียบจะอยู่ในบรรทัดถัดมา

2.5.5 **แบบฟอร์ม I (Input Specification)** ใช้กำหนดเกี่ยวกับข้อมูลที่จะนำเข้าของแฟ้มข้อมูลชนิดต่างๆ แบ่งเป็น 2 ส่วนคือ

1) **ส่วนแฟ้มข้อมูล** ใช้ในการระบุว่าเป็นข้อมูลของแฟ้มข้อมูลใด หรือเป็นโครงสร้างข้อมูล มีส่วนประกอบที่สำคัญดังนี้

- ก) ชนิด ใช้แสดงชนิดของแบบฟอร์ม
- ข) ชื่อของแฟ้มข้อมูลที่อ้างถึง

หรือตัวแปร มีรายละเอียดดังนี้

2) ส่วนเขตข้อมูล ใช้ในการระบุลักษณะของเขตข้อมูล

- ก) ชนิด ใช้แสดงชนิดของแบบฟอร์ม
- ข) ตำแหน่งเริ่มต้นของข้อมูล
- ค) ตำแหน่งสิ้นสุดของข้อมูล
- ง) จำนวนทศนิยม ในกรณีที่ข้อมูลเป็นตัวเลข
- จ) ชื่อของข้อมูล

2.5.6 แบบฟอร์ม c (Calculation Specification) ใช้ในการกำหนดหน้าที่การประมวลผล โดยใช้คำสั่งต่าง เช่น การคำนวณ การนำข้อมูลเข้าออก มีรายละเอียดดังนี้

- 1) ชนิด ใช้แสดงชนิดของแบบฟอร์ม
- 2) ตัวชี้บอก ใช้แสดงเงื่อนไขในการทำงาน
- 3) ตัวถูกกระทำ(Factor1) ใช้เป็นตัวถูกกระทำ
- 4) คำสั่ง ใช้ระบุงานที่จะกระทำ
- 5) ตัวกระทำ(Factor2) ใช้เป็นตัวกระทำ
- 6) ที่เก็บผลลัพธ์ ใช้เป็นในการเก็บผล
- 7) ตัวชี้บอกสำหรับผลลัพธ์

ในแต่ละคำสั่งอาจประกอบด้วยส่วนประกอบที่กล่าวมาแล้วทั้งหมดหรือบางส่วนขึ้นอยู่กับความต้องการของคำสั่งนั้นๆ ซึ่งแสดงความต้องการการใช้ส่วนประกอบดังนี้

ตารางที่ 2.1 แสดงคำสั่งในภาษาอาร์พีจีทีนิยมใช้

Factor1	คำสั่ง	Factor2	Result	ตัวชี้บอก	หน้าที่
Option	ADD	Require	Require		บวกเลข
Require	BEGSR				จุดเริ่มต้นของโปรแกรมย่อย
Option	ENDSR				จุดสิ้นสุดของโปรแกรมย่อย
Require	CASxx	Require	Require		ไปทำโปรแกรมย่อยถ้าเงื่อนไขถูกต้อง
Require	CHAIN	Require		Require	ค้นหาข้อมูล
Require	DIV	Require	Require		หารเลข
	MVR		Require		เก็บเศษจากการหาร
	DO				สร้างโครงสร้างแบบวงรอบ



ตารางที่ 2.1 (ต่อ)

Factor1	คำสั่ง	Factor2	Result	ตัวชี้บอก	หน้าที่
Require	DOU	Require			ให้ทำงานจนกว่าเงื่อนไขจะถูก
Require	DOW	Require			ให้ทำงานขณะที่เงื่อนไขถูก
	ELSE				ทำเงื่อนไขที่ไม่จริงของ IF
Option	END				บอกจุดสิ้นสุดของโครงสร้าง
	EXSR	Require			ไปทำงานที่โปรแกรมย่อย
Require	IF	Require			ทำตามเงื่อนไขที่จริง
	MOVE	Require	Require		เคลื่อนย้ายข้อมูล
	MOVEL	Require	Require		เคลื่อนย้ายข้อมูลชิดซ้าย
Option	MULT	Require	Require		คูณเลข
	READ	Require		Require	อ่านแฟ้มข้อมูล
Option	SUB	Require	Require		ลบเลข
	Z-ADD	Require	Require		ตั้งค่าตัวเลขค่าบวก
	Z-SUB	Require	Require		ตั้งค่าตัวเลขค่าลบ

2.5.7 **แบบฟอร์ม O (Output Specification)** ใช้ในการนำข้อมูลออก แบ่งออกเป็น 2 ส่วนคือ

1) **ส่วนแฟ้มข้อมูล** ใช้ในการกำหนดแฟ้มข้อมูล วิธีการประมวลผล และเงื่อนไขในการทำงาน มีรายละเอียดดังนี้

- ก) ชนิด ใช้แสดงชนิดของแบบฟอร์ม
- ข) ชื่อของแฟ้มข้อมูลที่อ้างถึง
- ค) ชนิดของการทำงานตามวงจรรารีพีซี
- ง) ระบุประเภทของการประมวลผล เป็นแบบเพิ่มลดหรือแก้ไขระเบียบข้อมูล

จ) ส่วนควบคุมบรรทัดการพิมพ์  
 ฉ) ชื่อของส่วนนำออก ใช้ในกรณีที่ควบคุมการนำออกจากโปรแกรม

2) **ส่วนเขตข้อมูล** ใช้กำหนดตำแหน่งที่จะทำการบันทึกข้อมูล มีส่วนประกอบที่สำคัญดังนี้

- ก) ชนิด ใช้แสดงชนิดของแบบฟอร์ม
- ข) ชื่อของเขตข้อมูล
- ค) ตำแหน่งสุดท้ายของเขตข้อมูล

ดูแบบฟอร์มของภาษาอาร์พีจีได้ในภาคผนวก ก

2.6 **ตัวชี้บอก** เป็นเครื่องมือในการบอกสถานะต่างๆในภาษาอาร์พีจี เพื่อใช้ในการตรวจสอบเงื่อนไขต่างๆ เช่น ผลของการอ่านข้อมูลว่าจบเพิ่มข้อมูลหรือไม่ การตรวจสอบการสิ้นสุดการทำงานของโปรแกรม ตัวชี้บอกมีหน้าที่ต่างๆกันดังนี้

ตารางที่ 2.2 แสดงตัวชี้บอก

ตัวชี้บอก	หน้าที่
O1 - 99	ใช้ทั่วไป
L1 - L9	ใช้ในการตรวจสอบการเปลี่ยนแปลงของข้อมูลนำเข้า
KA-KY	ใช้สำหรับการรับข้อมูลจากจอภาพ
LR	ใช้กำหนดใช้จบโปรแกรมเมื่อหมดวงรอบ
1P	เป็นตัวบ่งชี้ที่ทำงานครั้งแรกของการเข้าสู่วงรอบการทำงาน

2.7 **โปรแกรมโครงสร้างภาษาอาร์พีจี** ภาษาอาร์พีจีมีลักษณะที่สามารถเขียนโปรแกรมแบบโครงสร้างได้ ลักษณะการเขียนโปรแกรมแบบโครงสร้างมี 3 แบบ

2.7.1 **แบบเรียงลำดับ (Sequence)** เป็นการทำงานแบบเรียงกัน ต่อๆไปลักษณะของโปรแกรมภาษาอาร์พีจีเป็นแบบเขียนลงฟอร์มโดยมีคำสั่งๆละ 1 บรรทัดซึ่งตรงกับโปรแกรมโครงสร้างตามแบบเรียงลำดับ



รูปที่ 2.4 แสดงผังงานโปรแกรมโครงสร้างแบบเรียงลำดับ

ตัวอย่าง

C  
C

MULT SALARY  
MOVEL'EDIT'

SAL#YR  
MODE

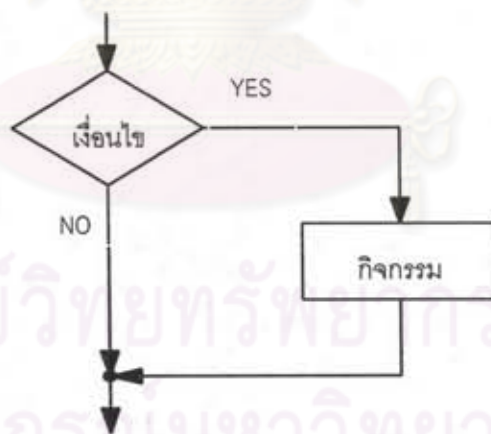
2.7.2 แบบเงื่อนไข (Condition) ใช้ในการกำหนดทางเลือกในการทำกิจกรรม ในการระบุเงื่อนไขเพื่อใช้ตรวจสอบสามารถกำหนดได้ดังนี้

ตารางที่ 2.3 แสดงเครื่องหมายที่ใช้ในการตรวจสอบเงื่อนไข

เงื่อนไข	เครื่องหมาย
เท่ากับ	EQ
น้อยกว่า	LT
มากกว่า	GT
น้อยกว่าหรือเท่ากับ	LE
มากกว่าหรือเท่ากับ	GE
ไม่เท่ากับ	NE

การเลือกทำแบบมีเงื่อนไขมี 3 แบบคือ

1) IF...END ใช้ในการเลือก ทำกิจกรรมเพียงอย่างเดียว



รูปที่ 2.5 แสดงผังงานโปรแกรมโครงสร้างแบบ IF...END

รูปแบบของคำสั่ง IF...END แสดงได้ตามตารางดังนี้

ตารางที่ 2.4 แสดงคำสั่ง IF...END

ตัวถูกกระทำ	คำสั่ง	ตัวกระทำ	ผลลัพธ์
field1	IFxx คำสั่งอื่นๆ	field2	
	END		

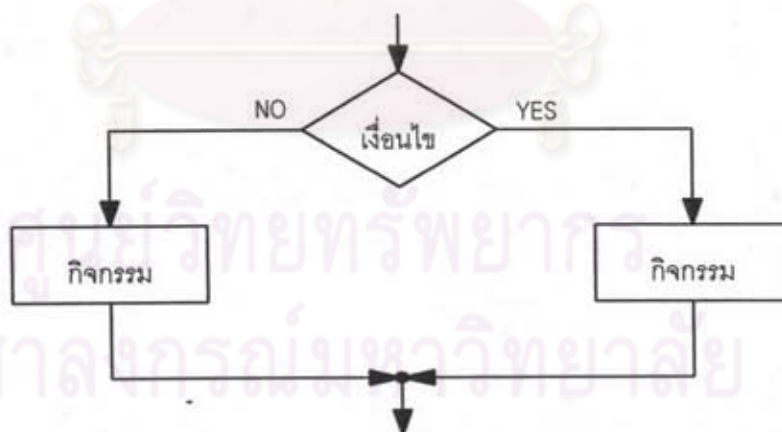
ตัวอย่าง

```

C          #WVERR  IFEQ 'N'
C          SETOF
C          CODE    CHAINMAEMP          99
C          99     MOVE 'N'           #FOUND
C          N99    MOVE 'Y'           #FOUND
C          END
    
```

2) IF...ELSE...END ใช้ในการเลือกทำกิจกรรมอย่างใด

อย่างหนึ่ง



รูปที่ 2.6 แสดงผังงานโปรแกรมโครงสร้างแบบ IF...ELSE...END

รูปแบบของคำสั่ง IF...ELSE...END แสดงได้ตามตารางดังนี้

ตารางที่ 2.5 แสดงคำสั่ง IF...ELSE...END

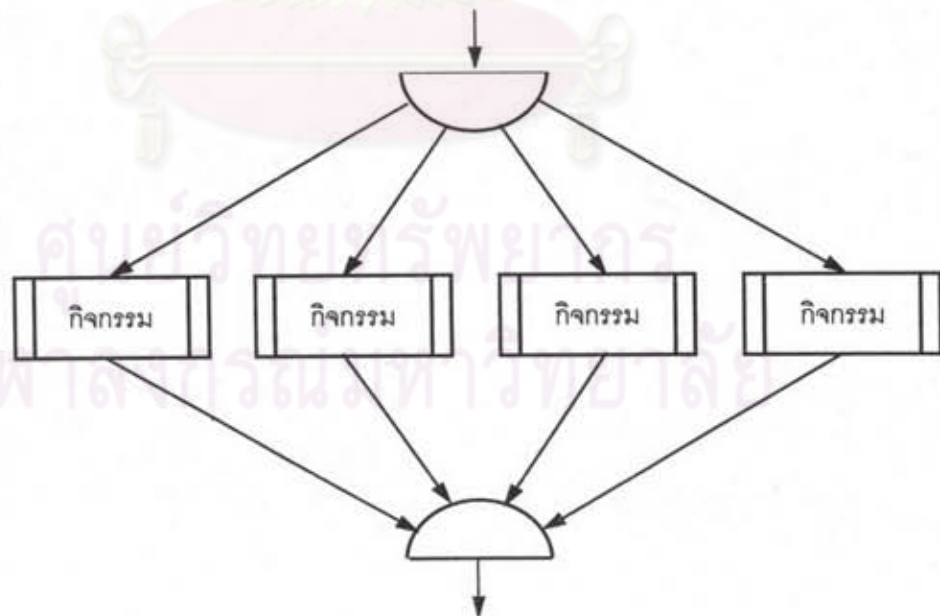
ตัวถูกกระทำ	คำสั่ง	ตัวกระทำ	ผลลัพธ์
field1	IFxx คำสั่งอื่นๆ ELSE คำสั่งอื่นๆ END	field2	

ตัวอย่าง

```

C      #FOUND      IFEQ 'Y'
C      MOVE 'N'      #ERROR
C      ELSE
C      MOVE 'Y'      #ERROR
C      END
  
```

3) **CASE...END** ใช้ในการเลือกทำกิจกรรมที่มีทางเลือกมากกว่า 2 ทางเลือก ในการเลือกกระทำกิจกรรมทำโดยการเรียกใช้โปรแกรมย่อย



รูปที่ 2.7 แสดงผังงานโปรแกรมโครงสร้างแบบ CASE...END

รูปแบบของคำสั่ง CASE แสดงได้ตามตารางดังนี้

ตารางที่ 2.6 แสดงคำสั่ง CASE

ตัวถูกกระทำ	คำสั่ง	ตัวกระทำ	ผลลัพธ์
field1	CASxx	field2	subroutine
field1	CASxx	field2	subroutine
	CAS		subroutine
	END		

ตัวอย่าง

```

C      CODE      CASEQ 'A'      ADDRTN
C      CODE      CASEQ 'D'      DELRTN
C      CODE      CASEQ 'E'      UPDRTN
C      CASE      ERRRTN
C      END
  
```

2.7.3 **แบบวน (LOOP)** ใช้ในการทำงานแบบซ้ำๆ ในการระบุเงื่อนไขเพื่อใช้ตรวจสอบกำหนดเช่นเดียวกับโครงสร้างแบบเงื่อนไข การทำกิจกรรมแบบแบบวนมี 2 แบบ

1) **DO...UNTIL** ใช้ในการกระทำกิจกรรมจนกระทั่งเงื่อนไขที่กำหนดไว้เป็นจริง



รูปที่ 2.8 แสดงผังงานโปรแกรมโครงสร้าง DO...UNTIL

รูปแบบของคำสั่ง DO...UNTIL แสดงได้ตามตารางดังนี้

ตารางที่ 2.7 แสดงคำสั่ง DO...UNTIL

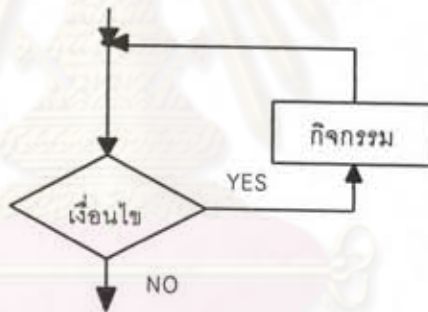
ตัวถูกกระทำ	คำสั่ง	ตัวกระทำ	ผลลัพธ์
field1	DOUxx คำสั่งอื่นๆ END	field2	

ตัวอย่าง

```

C      #LOOP      DOUGT100
C      ADD 1      COUNT
C      END
    
```

2) **DO...WHILE** ใช้ในการกระทำกิจกรรมในขณะที่เงื่อนไขที่กำหนดไว้เป็นจริง



รูปที่ 2.9 แสดงผังงานโปรแกรมโครงสร้าง DO...WHILE

รูปแบบของคำสั่ง DO...WHILE แสดงได้ตามตารางดังนี้

ตารางที่ 2.8 แสดงคำสั่ง DO...WHILE

ตัวถูกกระทำ	คำสั่ง	ตัวกระทำ	ผลลัพธ์
field1	DOWxx คำสั่งอื่นๆ END	field2	

ตัวอย่าง

```

C      #LOOP      DOWLT100
C      ADD 1      COUNT
C      END
    
```

2.8 คำสั่งของภาษาอาร์พีซี คำสั่งภาษาอาร์พีซีสามารถจัดแบ่งออกเป็นกลุ่มตามการใช้งานได้ดังนี้

2.8.1 กลุ่มการคำนวณ ได้แก่คำสั่งที่ใช้ในการคำนวณต่างๆ โดยใช้กับข้อมูลที่เป็นตัวเลขเท่านั้น โดยผลลัพธ์ที่ได้จะถูกนำไปเก็บที่ตัวแปรที่ใช้เก็บผลลัพธ์ เช่น

ตารางที่ 2.9 แสดงคำสั่งกลุ่มการคำนวณ

คำสั่ง	หน้าที่
ADD	ใช้ในบวกเลข
SUB	ใช้ในการลบเลข
Z-ADD	ใช้ตั้งค่าเลขบวก
Z-SUB	ใช้ตั้งค่าเลขลบ
MULT	ใช้ในการคูณ
DIV	ใช้ในการหาร
MVR	ใช้เก็บเศษจากการหาร

2.8.2 กลุ่มเคลื่อนย้ายข้อมูล เป็นคำสั่งย้ายข้อมูลจากตัวแปรหนึ่งไปยังตัวแปรอื่นๆ เช่น

ตารางที่ 2.10 แสดงคำสั่งกลุ่มเคลื่อนย้ายข้อมูล

คำสั่ง	หน้าที่
MOVE	ใช้ในการเคลื่อนย้ายข้อมูล
MOVEL	ใช้ในการย้ายข้อมูลโดยการย้ายขีดซ้าย

2.8.3 กลุ่มโปรแกรมย่อย เป็นคำสั่งเกี่ยวกับโปรแกรมย่อย เช่น

ตารางที่ 2.11 แสดงกลุ่มโปรแกรมย่อย

คำสั่ง	หน้าที่
BEGSR	ใช้บอกจุดเริ่มต้นของโปรแกรมย่อย
ENDSR	ใช้บอกจุดสิ้นสุดของโปรแกรมย่อย
EXSR	ใช้ในการสั่งให้โปรแกรมย่อยทำงาน



2.8.4 กลุ่มคำสั่งควบคุมขั้นตอนการทำงาน เป็นคำสั่งในการเขียนโปรแกรมแบบมีเงื่อนไข และการกระโดดข้ามขั้นตอนการทำงาน เช่น

ตารางที่ 2.12 แสดงกลุ่มคำสั่งควบคุมขั้นตอนการทำงาน

คำสั่ง	หน้าที่
DO	ใช้กำหนดวงรอบ
DOUxx	ทำเงื่อนไข ทำจนกระทั่งเงื่อนไขเป็นจริง
DOWxx	ทำเงื่อนไข ทำในขณะที่เงื่อนไขเป็นจริง
CASxx	ไปทำโปรแกรมน้อย ตามเงื่อนไข
IF	ตรวจสอบเงื่อนไข ถ้าถูกให้ทำคำสั่งต่อไป
ELSE	ให้ทำคำสั่งต่อไปถ้าเงื่อนไข IF ไม่เป็นจริง
END	ใช้บอกจุดสิ้นสุดของโครงสร้างทุกประเภท
GOTO	ใช้ในการกระโดดข้ามขั้นตอนการทำงาน
TAG	ใช้บอกตำแหน่งปลายทางของการโดด

2.8.5 กลุ่มการจัดการเพิ่มข้อมูล เป็นคำสั่งที่ใช้จัดการเกี่ยวกับเพิ่มข้อมูล เช่น

ตารางที่ 2.13 แสดงกลุ่มการจัดการเพิ่มข้อมูล

คำสั่ง	หน้าที่
READ	ใช้ในการอ่านข้อมูล
CHAIN	ใช้ในการค้นหาข้อมูล
SETLL	ใช้ในการกำหนดตำแหน่งของระเบียบข้อมูล

2.8.6 กลุ่มทั่วไป เป็นกลุ่มคำสั่งที่ใช้งานทั่วไป เช่น

ตารางที่ 2.14 แสดงกลุ่มทั่วไป

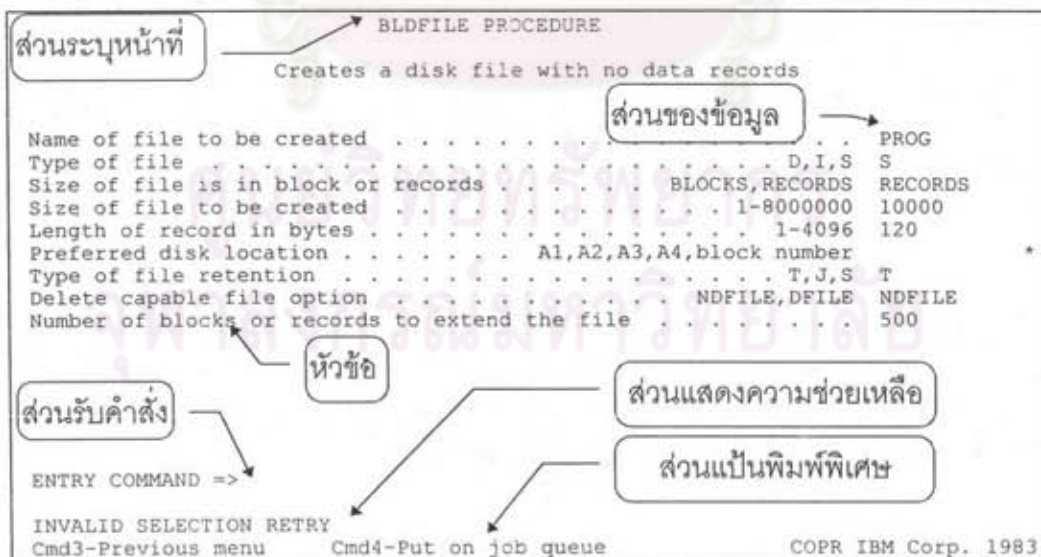
คำสั่ง	หน้าที่
SETON	ใช้ตั้งสถานะเปิด
SETOFF	ใช้ยกเลิกสถานะปิด

3. จอภาพ จอภาพเป็นส่วนที่ใช้ในการแสดงข้อมูลต่างๆ มีรายละเอียดดังนี้

3.1 ส่วนประกอบของจอภาพ จอภาพที่ใช้แสดงผลสามารถแบ่งส่วนต่างๆตามหน้าที่ได้ดังนี้

- 1) ส่วนระบุหน้าที่ (Format Identification) เป็นส่วนที่ใช้แสดงชื่อหน้าที่ของจอภาพ เพื่อบอกหน้าที่ของการแสดงผล
- 2) หัวข้อ (Captions) เป็นข้อความที่ใช้อธิบายความหมายของข้อมูลที่แสดงว่าข้อมูลที่แสดงนั้นคืออะไร
- 3) ส่วนของข้อมูล (Data Area) ใช้ในการแสดงข้อมูล (Display Area)หรือรับข้อมูล(Entry Area)
- 4) ส่วนแสดงความช่วยเหลือ (Message area) ใช้แสดงข้อความเพื่อให้ผู้ใช้ทราบถึงสิ่งที่ต้องทำในการกรอกข้อมูลหรือเกิดการผิดพลาดขึ้น
- 5) ส่วนรับคำสั่ง (Command area) เป็นส่วนที่ใช้ในการรับคำสั่งการทำงานจากผู้ใช้เพื่อนำไปทำงานต่อไป
- 6) ส่วนแป้นพิมพ์พิเศษ (Function /Command key list) ใช้แสดงข้อความบอกหน้าที่ของแป้นพิมพ์พิเศษว่าทำหน้าที่อะไร

ตัวอย่างของส่วนประกอบของจอภาพ



รูปที่ 2.10 แสดงส่วนประกอบของจอภาพ

### 3.2 ชนิดข้อมูล ข้อมูลที่จอภาพสามารถแสดงผลได้มีรายละเอียดดังนี้

3.2.1 **ข้อความ** ใช้ในการแสดงข้อความอธิบายการทำงานบนจอภาพ หรือเป็นคำอธิบายข้อมูลที่ใช้ในการแสดงผล

3.2.2 **เขตข้อมูล** ใช้ในการแสดงข้อมูลและหรือรับข้อมูล แบ่งตามประเภทของการรับส่งข้อมูลได้ 3 ประเภทคือ

- 1) **แสดงผล (OUTPUT)** ใช้ในการแสดงข้อความเพียงอย่างเดียว
- 2) **รับข้อมูลเข้า (INPUT)** ใช้ในการรับข้อมูลเข้า โดยไม่มีการแสดงผลมาก่อน
- 3) **แสดงผลและรับข้อมูลเข้า (INPUT/OUTPUT)** ใช้แสดงผลและรับข้อมูล

3.3 **ลักษณะการแสดงผลบนจอภาพ** จอภาพมีลักษณะการแสดงผลต่าง ๆ ดังนี้

- 1) **แสดงความสว่างสูง (High intensity)** เป็นการแสดงข้อความที่มีการเพิ่มความสว่างขึ้น ใช้ในการเน้นข้อความที่แสดง
- 2) **แสดงการกระพริบ (Blink)** เป็นการแสดงข้อความที่กำหนดให้มีการแสดงการกระพริบ ใช้ในการเตือน
- 3) **แสดงการผันกลับ (Reverse)** เป็นการแสดงข้อความที่กำหนดให้มีการแสดงผลกลับกันกับแบบปกติ ใช้ในการบอกความผิดพลาด
- 4) **แสดงการขีดเส้นใต้ (Underline)** เป็นการแสดงข้อความที่มีการขีดเส้นใต้ ใช้ในการเน้นข้อความที่แสดง
- 5) **ไม่ต้องการแสดงผล (Nondisplay)** เป็นการแสดงข้อความที่มีการปกปิดเพื่อไม่ให้มองเห็น ใช้ในการควบคุมหรือเป็นจุดทดสอบ
- 6) **แสดงการแบ่งคอลัมน์ (ColumnSep)** เป็นการแสดงขอบเขตของคอลัมน์สำหรับข้อความที่แสดงผล

3.4 **ข้อกำหนดของจอภาพ (Display format specification)** ใช้กำหนดรายละเอียดของการแสดงผลของจอภาพ มีรายละเอียดดังนี้

3.4.1 **ลักษณะของจอภาพ** ตามข้อกำหนดของจอภาพมีดังนี้

- 1) ขนาดของจอภาพมีขนาด 24 x 80 ตัวอักษร
- 2) ข้อความที่แสดงบนจอภาพต้องไม่อยู่ติดกัน โดยการเว้นด้านหน้าข้อความไว้ให้ห่างกัน 1 ตัวอักษร ทำให้ตำแหน่งที่เริ่มมีข้อความอยู่ที่แถวที่ 1 และคอลัมน์ที่ 2

3) ข้อกำหนดของจอภาพ สามารถประกอบด้วยจอภาพได้มากกว่า 1 จอภาพ

3.4.2 ส่วนประกอบของข้อกำหนดของจอภาพ โครงสร้างของข้อกำหนดของจอภาพแต่ละจอมีส่วนประกอบ 2 ส่วนคือ

1) ส่วนควบคุม (Display control specification) ใช้ในการควบคุมการทำงานของจอภาพ มีส่วนประกอบคือ

1) ชื่อจอภาพ เป็นชื่อที่ใช้ในโปรแกรมเพื่อเรียกให้จอภาพทำการแสดงผล ในแต่ละจอภาพจะมีชื่อที่ไม่ซ้ำกันการตั้งชื่อขึ้นต้นด้วยตัวอักษรตัวใหญ่ตัวต่อมาจะเป็นตัวอักษรตัวใหญ่หรือตัวเลขก็ได้ ความยาวไม่เกิน 8 ตัวอักษร

2) บรรทัดเริ่มต้น ใช้บอกบรรทัดแรกที่เริ่มทำการแสดงผล ซึ่งอาจจะไม่ใช่บรรทัดแรกของจอภาพก็ได้

3) จำนวนบรรทัดที่ต้องการลบ ก่อนการแสดงผลจะมีการลบข้อความที่มีอยู่บนจอภาพออกก่อน การกำหนดจำนวนบรรทัดที่จะลบ นับจากบรรทัดแรกที่แสดงผล แต่ต้องไม่เกินจำนวนบรรทัดที่สามารถแสดงได้บนจอภาพ

4) สถานะที่จะส่งเสียงเตือน กำหนดสถานะที่จะส่งเสียงเมื่อทำการแสดงผล โดยการกำหนดตัวชี้บอกที่ควบคุม ในการใช้งานสถานะของตัวชี้บอกจะเป็นตัวกำหนดว่าจะส่งเสียงหรือไม่

5) กำหนดการใช้แป้นพิมพ์พิเศษ ใช้ในการกำหนดให้มีการใช้แป้นพิมพ์พิเศษได้หรือไม่

6) แป้นพิมพ์พิเศษ ใช้ในการกำหนดแป้นพิมพ์พิเศษที่ใช้กับจอภาพนั้นๆ เริ่มจากแป้นพิมพ์ F1 ถึง F24 โดยการแทนด้วยอักษร A-Y ยกเว้น O แสดงได้ตามตารางดังนี้

ตารางที่ 2.15 แสดงความสัมพันธ์ของตัวอักษรและแป้นพิมพ์ที่ใช้

ตัวอักษร	แป้นพิมพ์	ตัวอักษร	แป้นพิมพ์	ตัวอักษร	แป้นพิมพ์
A	F1	I	F9	R	F17
B	F2	J	F10	S	F18
C	F3	K	F11	T	F19
D	F4	L	F12	U	F20
E	F5	M	F13	V	F21
F	F6	N	F14	W	F22
G	F7	P	F15	X	F23
H	F8	Q	F16	Y	F24

2) ส่วนข้อมูล (Field definition specification ) ใช้ระบุส่วนประกอบของจอภาพ ส่วนประกอบแต่ละส่วนจะต้องอยู่ห่างกัน 1 คอลัมน์ โดยแบ่งเป็น 2 ชนิด คือ

ก) รายละเอียดเกี่ยวกับข้อมูล ใช้ระบุส่วนรายละเอียดที่แสดงผลดังนี้

(1) ชื่อ ใช้ในการระบุชื่อของข้อความที่แสดงผล ปกติใช้กับข้อความที่ต้องการจะป้อนข้อมูล ส่วนข้อความคงที่จะปล่อยว่างไว้

(2) ความยาว กำหนดความยาวของข้อความที่ใช้แสดงผล

(3) บรรทัดที่แสดง ระบุบรรทัดที่ใช้ในการแสดงผล โดยนับจากบรรทัดที่กำหนดเป็นบรรทัดเริ่มต้นเป็นบรรทัดที่ 1

(4) คอลัมน์ที่แสดง ใช้ในการระบุคอลัมน์ที่ใช้ในการแสดงผล เริ่มจากคอลัมน์ที่ 1 ถึง 80

(5) สถานะการนำข้อความมาแสดงผล เป็นการระบุสถานะในการแสดงทางจอภาพ

(6) สถานะการรับข้อความ ใช้ระบุว่าข้อความสามารถป้อนข้อมูลลงไปได้หรือไม่

(7) ชนิดของข้อความ ที่รับเข้า ระบุชนิดของข้อมูลที่รับเข้า ถ้าเป็นตัวเลขให้ใส่ 'N' ตัวอักษรให้ว่างไว้

(8) ลักษณะการแสดงผล การแสดงผลข้อความมีลักษณะดังนี้คือ ความสว่างสูง กระพริบ ไม่แสดงผล ผันกลับ และขีดเส้นใต้ โดยการระบุเป็น 'Y' เมื่อต้องการให้แสดงลักษณะดังกล่าว

(9) ส่วนข้อความคงที่ ใช้ระบุข้อความคงที่ที่ใช้แสดงบนหน้าจอภาพ

(10) ส่วนต่อบรรทัด ใช้บอกสถานะการต่อบรรทัดของข้อความที่คงที่ที่ยาวเกิน 23 ตัวอักษร

ข) ข้อความเพิ่มเติม ใช้ในการต่อข้อความที่ยาวเกิน 23 ตัวอักษร โดยการกำหนดการต่อบรรทัดให้กับส่วนรายละเอียด และทำการใส่ข้อความส่วนที่เหลือ

4. การประมวลผลของเขตข้อมูล ใช้ในการตรวจสอบความถูกต้องของข้อมูลที่ป้อนเข้าไปในเขตข้อมูล หรือใช้ในการทำงานอื่นๆ เช่น การคำนวณวันที่ การค้นหาข้อมูล ซึ่งจะถูกกำหนด โดยการเขียนโปรแกรม ให้มีลักษณะการทำงาน 2 ส่วนคือ

1) ก่อนแสดงผล (Pre Display) เป็นการทำงานก่อนการแสดงผล ใช้ในการกำหนดค่าของเขตข้อมูลที่จะแสดงผล

2) หลังแสดงผล (Post Display) หรือการตรวจสอบความถูกต้อง (Validate Process) ใช้ในการตรวจสอบความถูกต้องของเขตข้อมูลหรือ การประมวลผล เช่น การคำนวณต่างๆ การเตรียมการเก็บแฟ้มข้อมูล

ตัวอย่าง

การเปลี่ยนวันที่เพื่อใช้ในการแสดงผล ข้อมูลของวันที่เก็บในรูปแบบ ปีเดือนวัน ส่วนในการแสดงผลต้องแสดงในรูปแบบ วันเดือนปี จึงต้องปรับข้อมูลของวันที่ใหม่ ก่อนการแสดงผล และเมื่อรับข้อมูลเข้าแล้ว ทำการเปลี่ยนข้อมูลกลับเป็น ปีเดือนวันดั้งเดิม ดังรูป



รูปที่ 2.11 แสดงการประมวลผลของเขตข้อมูลวันที่

5. การติดต่อกับแฟ้มข้อมูลของภาษาอาร์พีจี ในการติดต่อกับแฟ้มข้อมูลมีการกระทำ 2 ประเภท

5.1 การนำเข้าข้อมูล กระทำโดยการใช้คำสั่งอ่านหรือค้นหาข้อมูลจากแฟ้มข้อมูล จะถูกนำไปใส่ไว้ในที่เก็บของแฟ้มข้อมูลตามเขตข้อมูลในตำแหน่งต่างๆ

ตัวอย่าง

การอ่านข้อมูล โดยใช้คำสั่ง READ

```
C                READ EDIT                10
```

ตัวอย่าง

การค้นหาข้อมูล โดยใช้คำสั่ง CHAIN

```
C                LINE    CHAINEDIT        10
```

5.2 การส่งออกข้อมูล ภาษาอาร์พีจีกำหนดวิธีการส่งออกข้อมูลไว้ในแบบฟอร์ม O ทำให้การส่งข้อมูลออกต้องมีการกำหนดชื่อให้กับส่วนนำข้อมูลออก พร้อมกับระบุถึงวิธีการประมวลผลของแฟ้มข้อมูลไว้ในแบบฟอร์ม O แล้วใช้คำสั่งเรียกใช้งานชื่อส่วนนำข้อมูลออกนั้น เพื่อทำการส่งข้อมูลออกไปสู่แฟ้มข้อมูล

ตัวอย่าง

การเพิ่มข้อมูล โดยใช้คำสั่ง EXCPT

```
C                EXCPTADD
OMAINFO  EADD    ADD
O                MACODE    10
O                MANAME    50
O                MAZIP     55
O                MAAMT     60
```

การแก้ไขข้อมูล โดยใช้คำสั่ง EXCPT

```
C                EXCPTUPDATE
OMAINFO  E      UPDATE
O                MANAME    40
O                MAZIP     45
O                MAAMT     50
```

การลบข้อมูล โดยใช้คำสั่ง EXCPT

```
C                EXCPTDELETE
OMAINFO  EDEL   DELETE
```

6. การติดต่อระหว่างโปรแกรมกับจอภาพ ในการแสดงผลผ่านจอภาพของโปรแกรมภาษาอาร์พีจี จะมองจอภาพเสมือนแฟ้มข้อมูลแฟ้มหนึ่ง กระทำโดยใช้คำสั่งเขียนลงแฟ้มข้อมูลเป็นการส่งข้อมูลไปยังจอภาพ และอ่านแฟ้มข้อมูลเป็นการรับข้อมูลจากจอภาพเข้าสู่โปรแกรม เช่นเดียวกับการอ่านเขียนแฟ้มข้อมูลปกติ การส่งข้อมูลจะเกิดขึ้นเมื่อมีการกดปุ่ม ENTER หรือปุ่มพิเศษที่กำหนด การมองจอภาพเป็นเสมือนแฟ้มข้อมูล จำเป็นต้องเพิ่มเติมข้อมูลบางอย่างให้กับวิธีการใช้แฟ้มข้อมูลแบบปกติ ซึ่งมีรายละเอียดดังนี้

- 1) กำหนดชนิดของแฟ้มข้อมูลเป็นแบบทั้งอ่านและเขียนได้
- 2) กำหนดวิธีการประมวลผลเป็นชนิดควบคุมการทำงานด้วยโปรแกรม
- 3) ขนาดระเบียบ กำหนดให้เท่ากับผลรวมของขนาดข้อมูลที่มากที่สุด
- 4) การแฟ้มข้อมูลเป็นแบบเรียงลำดับ
- 5) อุปกรณ์ที่ใช้ให้กำหนดเป็นจอภาพ
- 6) กำหนดการต่อบรรทัด
- 7) ใส่ชื่อแฟ้มจอภาพที่ใช้ โดยนำหน้าด้วย 'FMST'

ในการเขียนให้ทำการเขียนลงแฟ้มข้อมูลจอภาพโดยระบุชื่อจอภาพที่จะใช้ก่อน การอ่านจะทำการอ่านแฟ้มข้อมูลจอภาพ ซึ่งข้อมูลที่ได้จะถูกนำเข้าสู่ระเบียบของแฟ้มข้อมูลเหมือนแฟ้มข้อมูลปกติ

ตัวอย่าง

การติดต่อกับจอภาพ SCR10

```

FWS      CD  F      123          WORKSTN
F
IWS      NS
I*  FORMAT-SCR10
I
C          EXCPTSCR10          1  50  COMMAN
C          READ  WS
C          SETON                LR
OWS      E          SCR10
O
O          COMMAN              K8  'SCR10  '
O

```

7. การโอนโปรแกรมระหว่างเครื่องคอมพิวเตอร์ การโอนโปรแกรมระหว่างเครื่องไมโครคอมพิวเตอร์กับเครื่องคอมพิวเตอร์ไอบีเอ็ม S/36 มีการทำงาน 3 ขั้นตอนดังนี้

7.1 การโอนข้อมูล ในการโอนข้อมูลจากเครื่องไมโครคอมพิวเตอร์มายังเครื่องไอบีเอ็ม S/36 กระทำโดยผ่านโปรแกรมโอนย้ายข้อมูล เช่น PC SUPPORT และ IDEA LINK ซึ่งจะทำการโอนแฟ้มข้อมูลจากเครื่องไมโครคอมพิวเตอร์มาเป็นแฟ้มข้อมูลบนเครื่อง



ไอบีเอ็ม S/36 ในการโอนจะต้องทำการสร้างแฟ้มข้อมูลบนเครื่องไอบีเอ็ม S/36 ก่อน โดยการ  
ใช้คำสั่ง BLDFILE ทำการสร้างแฟ้มข้อมูลชนิดเรียงลำดับมีความยาวของระเบียบ 120 ตัวอักษร

รูปแบบของคำสั่งมีดังนี้

BLDFILE <ชื่อแฟ้มข้อมูลที่ใช้>,S,RECORDS,<จำนวนบรรทัดของแฟ้มข้อมูล>,120,,  
T,,NDFILE,,<จำนวนระเบียบที่จะเพิ่มในกรณีทีแฟ้มข้อมูลเต็ม>

ตัวอย่าง

BLDFILE PROG,S,RECORDS,10000,120,,T,,NDFILE,,500

สามารถอธิบายได้ดังนี้

ทำการสร้างแฟ้มข้อมูล ชื่อ "PROG" ชนิดเรียงลำดับ มีขนาด 10000 ระเบียบ มีความ  
ยาวของระเบียบ 120 ตัวอักษร เป็นแฟ้มข้อมูลถาวร ถ้าแฟ้มข้อมูลเต็มสามารถขยายได้ครั้งละ  
500 ระเบียบ

หรือใช้วิธีตอบคำถามจากหน้าจอภาพ กระทำโดยการป้อนคำสั่ง BLDFILE แล้วกดแป้นขอความ  
ช่วยเหลือ ( HELP ) จะปรากฏจอภาพขึ้น ทำการตอบคำถาม โดยปกติจะป้อนชื่อ, ชนิด, ขนาด  
และความยาวของแฟ้มข้อมูล ขนาดที่ต้องการขยายในกรณีแฟ้มข้อมูลเต็ม กดแป้นให้โปรแกรม  
รับคำสั่ง โปรแกรมจะทำการเรียกโปรแกรม BLDFILE ให้ทำการสร้างแฟ้มข้อมูล จากนั้นทำการ  
โอนโดยใช้ PC SUPPORT หรือ IDEA LINK ต่อไป รูปแบบของจอภาพแสดงได้ดังรูป

BLDFILE PROCEDURE		Optional-*
Creates a disk file with no data records		
Name of file to be created	PROG	
Type of file	D, I, S	S
Size of file is in block or records	BLOCKS, RECORDS	RECORDS
Size of file to be created	1-8000000	10000
Length of record in bytes	1-4096	120
Preferred disk location	A1, A2, A3, A4, block number	
Type of file retention	T, J, S	T
Delete capable file option	NDFILE, DFILE	NDFILE
Number of blocks or records to extend the file		500
Cmd3-Previous menu      Cmd4-Put on job queue      COPR IBM Corp. 1983		

รูปที่ 2.12 แสดงจอภาพที่ใช้ในการสร้างแฟ้มข้อมูล

7.2 แปลงแฟ้มข้อมูลให้เป็นโปรแกรม เครื่องคอมพิวเตอร์ S/36 เก็บโปรแกรมแยกออกจากแฟ้มข้อมูล โปรแกรมจะรวมกันอยู่ในแฟ้มห้องสมุด (Library) ในการแปลงข้อมูลทำได้โดยนำแฟ้มข้อมูลลงไปอยู่ในแฟ้มห้องสมุด โดยการแยกข้อมูลตามชนิดของข้อมูลที่กำหนดเพิ่มเติมไว้ในโปรแกรมที่จะทำการโอนโปรแกรม ในแฟ้มข้อมูลที่จะส่งมีได้หลายโปรแกรม ซึ่งอาจเป็นโปรแกรมภาษาและภาษาจอภาพรวมกันอยู่ ในการแบ่งแยกโปรแกรมแต่ละโปรแกรมออกจากกัน จะมีส่วนควบคุมเพื่อระบุชื่อ, ชนิด ประเภทของโปรแกรม ตลอดจนจบจุดจบของโปรแกรม โครงสร้างของโปรแกรมแต่ละโปรแกรมมีรายละเอียดดังนี้

7.2.1 ส่วนควบคุม ใช้ระบุชื่อ, ชนิด และประเภทของข้อมูลโปรแกรม เป็นข้อความบรรทัดแรกของโปรแกรมที่จะส่ง มีส่วนประกอบที่สำคัญดังนี้

- 1) ชนิดของข้อมูล (LIBRARY) ระบุเป็น S
- 2) ชื่อ (NAME) ใช้ระบุชื่อโปรแกรม
- 3) ประเภท (SUB) ใช้ระบุประเภทของข้อมูลดังนี้
  - ก) โปรแกรมภาษาอาร์พีจีใช้ RPG
  - ข) ข้อกำหนดของจอภาพใช้ FMT

7.2.2 ส่วนของข้อมูล เป็นส่วนของโปรแกรมหรือภาษาจอภาพ

7.2.3 ส่วนจุดสิ้นสุดของข้อมูล ทำโดยการใส่คำสั่ง "// CEND" ที่จุดสิ้นสุดของโปรแกรม

ตัวอย่าง

ข้อมูลชื่อ EDIT เป็นชนิดโปรแกรมภาษาอาร์พีจี

// COPY LIBRARY-S,NAME-EDIT,SUB-RPG

โปรแกรม

// CEND

การเปลี่ยนแฟ้มข้อมูลที่ได้จากเครื่องไมโครคอมพิวเตอร์มาเป็นโปรแกรม ทำได้โดยใช้คำสั่ง TOLIBR ในการเปลี่ยนแฟ้มข้อมูล คำสั่งดังกล่าวจะทำหน้าที่แยกโปรแกรมแต่ละโปรแกรมออกจากกันโดยดูจากโครงสร้างของโปรแกรม แล้วเก็บลงในห้องสมุดโปรแกรม

รูปแบบของคำสั่งมีดังนี้

TOLIBR <ชื่อแฟ้มข้อมูล>, F1,, REPLACE, <ชื่อห้องสมุดโปรแกรมที่ใช้เก็บ>,,, ALL, SOURCE

ตัวอย่าง

TOLIBR PROG,F1,,REPLACE,MYLIB,,,,ALL,SOURCE

สามารถอธิบายได้ดังนี้

ทำการเปลี่ยนแฟ้มข้อมูลชื่อ PROG ไปไว้ในห้องสมุดโปรแกรมชื่อ MYLIB

หรือใช้วิธีตอบคำถามจากหน้าจอภาพ กระทำโดยการป้อนคำสั่ง TOLIBR แล้วกดแป้นขอความช่วยเหลือ ( HELP ) จะปรากฏจอภาพขึ้น ทำการป้อนชื่อแฟ้มข้อมูล ชื่อห้องสมุดโปรแกรม แล้วกดแป้นให้โปรแกรมรับคำสั่ง โปรแกรมจะทำการเรียกโปรแกรม TOLIBR ให้ทำการแปลงแฟ้มข้อมูล รูปแบบของจอภาพแสดงได้ดังรูป

TOLIBR PROCEDURE		Optional**
Copies disk, diskette, or tape files into a library		
Name of file containing library members . . . . .	PROG	
Location of file . . . . . I1 ,F1	F1	
Creation date of file . . . . .		*
To replace existing library member, enter REPLACE . . . . . REPLACE	REPLACE	*
Name of library to contain members . . . . .	TEMP	
Name of member, partial name of members, or ALL . . . . .	ALL	
If partial name is entered, enter ALL . . . . .	ALL	*
Type of library member . . . . . LIBRARY, LOAD, PROC, SOURCE, SUBR	SOURCE	
Subtype of member . . . . .		*

Cmd2-Page back      Cmd4-Put on job queue      COPR IBM Corp. 1986

### รูปที่ 2.13 แสดงจอภาพที่ใช้ในการแปลงข้อมูลเป็นโปรแกรมต้นฉบับ

**7.3 แปลโปรแกรม** การแปลงจะทำบนเครื่องไอบีเอ็ม S/36 โปรแกรมจอภาพประกอบด้วยส่วนของภาษาอาร์พีจีที่ใช้ควบคุม และข้อกำหนดของภาษาจอภาพที่ใช้ในการแสดงผล ในการแปลงจะต้องทำการแปลทั้ง 2 ส่วน

**7.3.1 ส่วนของภาษาอาร์พีจี** การแปลภาษาอาร์พีจีจะใช้โปรแกรม RPGC ในการแปล โดยสามารถทำงานได้ 2 ลักษณะ คือ

1) การใช้คำสั่ง เป็นการป้อนคำสั่งโดยตรงเพื่อทำการแปล โดยการป้อนคำสั่ง RPGC แล้วกดแป้นให้เครื่องทำงานตามคำสั่ง

รูปแบบคำสั่งมีดังนี้

RPGC <ชื่อของโปรแกรมภาษาอาร์พีจี>, <ที่อยู่ของโปรแกรม>

ตัวอย่าง

RPGC PROG,MYLIB

สามารถอธิบายได้ดังนี้

สั่งให้เครื่องทำการแปลโปรแกรมภาษาอาร์พีจีชื่อ PROG ที่อยู่คือ MYLIB

2) ใช้วิธีตอบคำถาม จากหน้าจอภาพ กระทำโดยการป้อนคำสั่ง RPGC แล้วกดแป้นขอความช่วยเหลือ ( HELP ) จะปรากฏจอภาพขึ้น ทำการตอบคำถาม โดยปกติจะป้อนชื่อ และที่อยู่ของโปรแกรม จากนั้นกดแป้นให้โปรแกรมรับคำสั่ง โปรแกรมจะทำการเรียกโปรแกรม RPGC ให้ทำงานให้ตามรายการที่ใส่ไว้ รูปแบบของจอภาพแสดงได้ดังรูป

RPGC PROCEDURE		Optional-*
Compiles an RPG II source program		
Name of source program to be compiled . . . . .		
Name of library containing source program . . . . .		MYLIB
Create diagnosed source member . . . . .	NODSM,DSM	NODSM
Output option for compiler listings . . . . .	PRINT,NOPRINT,CRT	PRINT
Create cross-reference listing . . . . .	NOXREF,XREF	NOXREF
Maximum number of requesting display stations . . . . .	0-99	0
Never-ending program . . . . .	NONEP,NEP	NONEP
Name of library to contain compiled program . . . . .		*
Cmd3-Previous menu	Cmd4-Put on job queue	Cmd14-More options
COPR IBM Corp. 1986		

รูปที่ 2.14 แสดงจอภาพที่ใช้ในการแปลโปรแกรมภาษาอาร์พีจี

### 7.3.2 ข้อกำหนดของจอภาพ ในการแปลข้อกำหนดของจอภาพ

จะใช้โปรแกรม FORMAT ในการแปล โดยสามารถทำงานได้ 2 ลักษณะเช่นกัน คือ

1) การใช้คำสั่ง เป็นการป้อนคำสั่งโดยตรงเพื่อทำการแปล โดยการป้อนคำสั่ง FORMAT คำสั่ง FORMAT วิธีการแปลมีหลายวิธีคือ การสร้างใหม่ (Create) การเพิ่ม (Add) การลบ (Delete) และการแก้ไข (Update) โดยปกติจะทำการสร้างใหม่ทั้งหมด

รูปแบบคำสั่งมีดังนี้

FORMAT <วิธีการแปล>,<ชื่อของผลของการแปล>,<ที่อยู่ของผลของการแปล>,<ชื่อภาษาจอตภาพ>,<ที่อยู่ของภาษาจอตภาพ>,<จำนวนจอตภาพที่ใช้>

ตัวอย่าง

FORMAT CREATE,PROGFM,MYLIB,PROGFM,MYLIB,50

สามารถอธิบายได้ดังนี้

ทำการแปลภาษาจอตภาพโดยวิธีสร้างใหม่ ผลจากการแปลให้มีชื่อว่า PROGFM นำไปไว้ที่ MYLIB ใช้ภาษาจอตภาพชื่อ PROGFM อยู่ที่ MYLIB จำนวนจอตภาพที่ทำการแปลมีจำนวน 50 จอตภาพ

2) ใช้วิธีตอบคำถาม จากหน้าจอตภาพ กระทำโดยการป้อนคำสั่ง FOTMAT แล้วกดแป้นขอความช่วยเหลือ( HELP ) จะปรากฏจอตภาพขึ้น ทำการตอบคำถาม โดยปกติจะป้อนวิธีการแปล, ชื่อและที่อยู่ของผลการแปล, ชื่อและที่อยู่ของชื่อกำหนดของจอตภาพ, และจำนวนจอตภาพที่จะทำการแปล จากนั้นกดแป้นให้โปรแกรมรับคำสั่ง โปรแกรมจะทำการเรียกโปรแกรม FORMAT ให้ทำการแปลให้ รูปแบบของจอตภาพแสดงได้ดังรูป

FORMAT PROCEDURE		Optional-*
Processes source statements for \$SFGR display formats		
Format generation option . . . . .	CREATE,ADD,DELETE,UPDATE	CREATE
Name of display format load member . . . . .		PROGFM
Name of library to contain load member . . . . .		MYLIB
Name of display format source member . . . . .		PROGFM
Name of library containing source member . . . . .		MYLIB
Number of display formats in source member . . . . .	1-255	1
To replace an existing load member, enter REPLACE . . . . .	REPLACE	
Halt on error messages . . . . .	HALT,NOHALT	HALT
Print option . . . . .	PRINT,NOPRINT,PARTIAL	PRINT

จุฬาลงกรณ์มหาวิทยาลัย

Cmd2-Page back      Cmd4-Put on job queue      COPR IBM Corp. 1983

รูปที่ 2.15 แสดงจอตภาพที่ใช้ในการแปลข้อกำหนดของจอตภาพ